# Using Ipsative Assessment to Improve Feedback Quality and the Student Assessment Experience in University Computer Science.

Thesis By

Ryan Crosby

In partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science

School of Computing

Newcastle University

December 2020

# Declaration

I declare that this thesis is my own work unless otherwise stated. No part of this thesis has previously been submitted for a degree or any other qualification at Newcastle University or any other institution.

Ryan Crosby

# List of Publications

Computing: Ipsative Assessment. Improving the student experience in higher education using personalised assessment and feedback. ITiCSE '19, July 15–17, 2019, Aberdeen, Scotland Uk © 2019 Copyright held by the owner/author(s). ACM ISBN 978-1-4503-6301-3/19/07. https://doi.org/10.1145/3304221.3325565

CS EDU Doctoral Consortium, "A personalised feedback system for computer programming" 02-05-2019

Horizons in STEM Presentation "Ipsative Assessment and Feedback for improving student learning"

Ryan Crosby. 2021. Student Perceptions of Assessment and Feedback - are they valid?. In Computing Education Practice 2021 (CEP '21), January 7, 2021, Durham, United Kingdom. ACM, New York, NY, USA, Article 4, 4 pages.

# Abstract

Assessment and feedback are important aspects of Higher Education used to provide certification to students and to facilitate learning. Therefore, both practices are important to get right. Despite their significance, the results of the National Student Survey show that assessment and feedback are the biggest sources of dissatisfaction for UK undergraduate students (Ferrell, 2012).

This thesis outlines a study that set out to understand the issues faced by undergraduate students' in their experience of assessment and feedback, and whether the adoption of an Ipsative assessment approach could help to ameliorate these issues. Ipsative assessment focuses on the students' improvement from the previous assessment and their personal best as opposed to the meeting of external criteria. The focus of Ipsative assessment is placed on the students learning gain and the distance travelled, not on the ability to meet certain assessment criteria (Hughes, 2011; Hughes et al, 2014).

This research tracked three cohorts of undergraduate students taking their first Computer Science module in programming at Newcastle University. Case study 1 set out to understand students' perceptions of issues within their assessment and feedback experience. Data were collected from student focus groups, interviews and questionnaires. Case study two analysed historical online feedback given to previous cohorts on the module CSC1021, programming 1, during the academic years 2012/2013 to 2016/2017. In total, 942 items of feedback were analysed to determine if the perceived issues were consistently present in the feedback given to students. Finally, to investigate the potential impacts of Ipsative assessment, a system called 'Computing: Ipsative Assessment' was trialled with first-year students.

Three main issues were identified to contribute to student dissatisfaction, (i) a lack of assessment literacy, (ii) the mark-driven nature of students and (iii) a mismatch of expectations between staff and students regarding feedback. By encouraging the use of self-reflection and self-guided learning through Ipsative Assessment, these issues could be mitigated.

## Acknowledgements

I would like to thank my supervisor, Dr Marie Devlin, for her continued support and fuelling of my passion for all things pedagogy. I would also like to thank Dr Jennifer Warrender who throughout the PhD process went from advisor to friend, thank you for all the food and rambles you have supplied throughout the years. To my Mam and Dad, who let me stay in their house long after I should have left and supported me every step of the way. To Amy, thank you for listening to my rambles, and guiding me through the anxiety that comes from dealing with a PhD. To all the Computing support staff and academic staff, who there are too many to name, thank you, your support has been immeasurable. To you, for reading this thesis. A thesis is a work of passion; I hope to share that passion with you.

# Table of Contents

# Table of Figures

# Table of Tables

# Glossary of terms

| Number | Term | Definition |
|---|---|---|
| 1 | Ipsative Assessment | A form of assessment that measures a student's learning gain throughout a course rather than focus on an external criterion. |
| 2 | Assessment Literacy | Assessment literacy is a student's ability to understand the purpose of assessment and how it is important to the development of their learning |
| 3 | Summative Assessment | A form of assessment that indicates a learner's success in meeting the assessment criteria and learning outcomes of a course |
| 4 | Formative Assessment | A form of assessment that focuses on the developmental purpose by giving feedback on their performance and how it can be improved |
| 5 | Learning gain | "The distance travelled by students during their studies" (HEFCE, 2016) |
| 6 | Surface Learning | A student intends to complete the work with as little effort as possible. Surface learning is an approach which students apply with the aims of passing the assessment without gaining a mastery of the material |
| 7 | Deep Learning | A student gains a true mastery of the subject and can begin to make connections between the skills used on different assignments. |

# Chapter 1. Introduction

## *1.1 Problem Overview*

Higher education has two main purposes, to facilitate learning and to provide certification (Washer, 2007). A core component to achieving these goals is the use of assessment and feedback. Assessment and feedback have two main categories, the first is summative assessment, which provides a quantifiable, often high stakes value, to a student's knowledge and is the main tool used to provide certification. Summative assessment is often termed as 'assessment of learning,' however, all assessment should be 'for learning' in that it benefits the student and helps with knowledge gain. The second, formative assessment, is often termed as assessment for learning. Formative assessment has low stakes value and is commonly used to help inform a student of their knowledge state but has minimal risk to the final grade attached to it, hence facilitating and encouraging learning (QAA, 2018; Petrovic et al., 2019). When used to their full potential summative and formative assessment are tools that help guide students through their degree (Wilson, 2006).

The National Student Survey (NSS) is a way of measuring student opinions on their experiences of higher education. The NSS is a "key component of the regulatory landscape in UK higher education" (Office for Students, 2020), it is an important measurement that details the quality of teaching resources at a Higher Education level. The NSS supplies publicly viewable information of student opinions on their time at a given Higher Education institute. It is important to keep a good NSS score, as not only do these results give universities an understanding of their current student perceptions, but it also shows prospective students information on how the courses they are applying to look.

The NSS results in the last decade show that assessment and feedback is the "single biggest source of student dissatisfaction within the Higher Education experience in the UK" (Ferrell, 2012). The results are calculated based on the number of respondents who selected "4: Agree" and "5: Strongly agree" on the survey. In the academic year 2015/2016, the NSS had one of the lowest results in the assessment and feedback category in the past decade. Newcastle University obtained a 49% positive response to the statement "feedback on my work has helped me clarify things that I did not understand" (HEFCE, 2017).

Based on a review on the NSS in 2017, there were changes made to the survey statements. Due to these changes, comparing the years following 2017 and the impact they had on student satisfaction is difficult and makes an exact comparison impossible (Office for Students, 2019). However, when looking at the new statement “I have received helpful comments on my work” the latest result from 2019 for the UK is 73%. While this is an improvement compared to previous years, these results still show an issue with students’ perceptions of feedback. Table 1 shows a breakdown of the NSS scores across England from 2010 to 2016, focusing on the assessment and feedback questions. The results within the NSS are expected to be above 80%. Office for Students explain that since 2008 Overall Satisfaction on the NSS “has never fallen below 80 per cent” (Office for Students, 2019), clearly this result is used as a positive benchmark. Therefore, with results ranging from 57% to 77% across the assessment and feedback questions, it is clear that there is a national dissatisfaction. It is particularly important to pay attention to the results of the statement “feedback on my work has helped me clarify things I did not understand.” While clarifying misunderstandings is a crucial aspect of feedback (Black and William, 1998; Ferrell 2012) students are identifying that this does not happen, with the highest rate of agreement in the NSS being 68% at the time of the data collection.

| NSS Question | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 |
|---|---|---|---|---|---|---|---|
| The criteria used in marking have been clear in advance. | 70 | 72 | 74 | 75 | 76 | 77 | 77 |
| Assessment arrangements and marking have been fair. | 72 | 73 | 75 | 76 | 76 | 77 | 77 |
| Feedback on my work has been prompt. | 59 | 62 | 65 | 68 | 69 | 70 | 71 |
| I have received detailed comments on my work. | 63 | 66 | 68 | 70 | 71 | 72 | 72 |
| Feedback on my work has helped me clarify things I did not understand. | 57 | 60 | 63 | 65 | 66 | 68 | 68 |

Table 1 Assessment and Feedback NSS results from 2010 – 2016 of full-time students within England

The assessment and feedback NSS results with the highest agree rate were the questions regarding assessment. The questions regarding clarity of the criteria and fairness of the questions have remained closer to 80% than the feedback questions. These results suggest that while students may not be satisfied with the practices, they are considered to be fair and valid.

While these results are illuminating, the validity of the NSS is often brought into question when discussing student perceptions. In their report, Callender (2014) argues that the NSS explores student satisfaction rather than the actual quality of the teaching and assessment provided. Therefore, the results from the NSS cannot be used to conclusively state that assessment and feedback are not at a good standard, they are however a good identifier of the perceptions of students.

### *1.2 Research Motivation*

As assessment and feedback are a crucial aspect of higher education, it is important to understand why students are dissatisfied and where practices could be potentially improved. This research investigates the potential causes of student dissatisfaction, how this impacts academic practice and how these issues could be ameliorated.

There are common problems with assessment and feedback that have been identified throughout pedagogical literature. There are three issues that are important for this research. First it can often be seen that students lack assessment literacy. Assessment literacy (Definition 2, page xi) is a student's ability to understand the purpose of assessment and how it is important to the development of their learning (Smith, 2013). A lack of assessment literacy can lead to additional issues such as the mark driven nature of students, which can cause a lack of understanding of the assessment purpose. This mark-driven nature is perhaps further exacerbated by the criteria driven nature of assessment (Smith et al, 2013). Third, the modularisation of modern day courses in combination with a lack of assessment literacy in students make the connections and skill transference difficult to see between modules (Ferrell, 2012).

Ipsative assessment (Definition 1, Page xi) and feedback is a form of assessment that measures a student's learning gain throughout a course rather than focus on an external criteria. It focuses on the improvement that a student has made throughout their time as a learner (Hughes, 2014). Ipsative assessment and feedback may help reduce the impact the issues stated above by helping students to develop their assessment literacy. When following an Ipsative form, students practice the ability to identify what skills they need to improve and how these skills are linked to each other. This can help to reduce the impact of modularisation while also assisting the students in the development of their assessment literacy skills. The highly mark driven nature of students, often caused by the external criteria referenced standards may also be mitigated with the use of Ipsative assessment. This could be achieved by placing focus on the improvements made by the students, therefore encouraging a growth mindset (Hughes, 2011; Dweck et al., 1998).

This research investigates the factors which may impact a student's perception of their Higher Education experience. To understand where these issues stem from and what the student experience is within higher education, two studies conducted with Newcastle University Computer Science students will be used as the primary data sources for this research. Within these studies, this research investigates what good practice is, how Ipsative practice compares, and if Ipsative assessment could help solve some of the perceived student issues. These two case studies were conducted during an 'introduction to programming' module, investigating both student perceptions and the feedback given to students. The module in question CSC1021 (a full module description can be seen in Appendix A) aims to:

*"give students an appreciation of the way problems can be solved using a programming language. It provides an intuitive introduction to the concept of programming and covers basic programming constructs including comments, variables, assignment statements, control structures (selection, repetition), expressions, modularisation, input and output."* (Nesbitt, 2020)

As programming is a core concept in a Computer Science degree, this module was chosen to investigate the assessment and feedback problems and potential solutions.

### *1.3 Research Questions*

To understand the causes of student dissatisfaction and how the impact of these issues may be reduced, this research aims to answer the following questions:

**RQ1** What is good practice in assessment and feedback in undergraduate computer science in the UK and how does current practice at Newcastle University compare?

> In addition to a literature review to understand what good practice is in higher education, staff will be interviewed to identify what current practices are being used at Newcastle University.

**RQ2** What are the current perceived assessment and feedback issues experienced by students?

> To answer RQ2, students will be interviewed to identify their current perceptions of assessment and feedback as well as their perceived issues. Additionally, these interviews will identify what experiences students have with assessment and feedback and how they use it within their learning.

**RQ3** How does Ipsative assessment compare to current practice and can it be used to ameliorate issues identified by students?

> As this research focuses on students on the Computer Science course, current tools for programming assessment are compared to identify what they do well and what they do not do well, in order to identify what is missing from these approaches. From this Ipsative assessment is explored as a potential way to reduce the impact of these issues identified by students.

### *1.4 Research Risks*

The main risks foreseen with this research is there may be a reluctance from students and staff to explain how they use and provide feedback. They may end up stating what they expect the researchers want to hear rather than outlining what they do in practice. A resolution for this is to use anonymous questionnaires during lectures where students will already attend.

### *1.5 Research Contributions*

The primary contribution of this research is the trial of an Ipsative assessment inspired system with undergraduate novice programmers. The system is justified via the evaluation of student perceptions and the exploration of historical data. The system is evaluated via an undergraduate trial to investigate whether Ipsative assessment can help ameliorate the issues faced by students regarding assessment and feedback. Additional contributions include a developed feedback criteria and two additional data sets regarding students assessment and feedback experiences at a UK higher education institute with a focus on the Computer Science discipline.

### *1.6 Thesis Structure*

- Chapter Two of this thesis explores the relevant literature and theories relating to assessment and feedback discussed throughout this research. Chapter two concludes with an exploration of a developed feedback criteria to explore what consists of best feedback practice.
- Chapter Three outlines the methods used within this research, along with the statistical tests that were used to investigate potential causes and links in the data.
- Chapter Four investigates the qualitative and quantitative results to understand the perceived issues students have with assessment and feedback and the impact on academic practice.
- Chapter Five explores the quality of the feedback given to students using the developed feedback criteria from Chapter Two.

- Chapter Six details a trial using the developed system to investigate if Ipsative assessment could be a potential solution to student dissatisfaction.
- Chapter Seven discusses the work carried out throughout this research and how it relates to the identified literature.

# Chapter 2. Literature Review

### *2.1 Introduction*

As explored in chapter one, two main purposes of higher education are to provide certification and to facilitate learning (Washer, 2007), to fulfil both of these purposes higher education must have a focus on lifelong learning with an active outlook on learning aims and key skills (Boud et al., 2006). Shadbolt identified that in addition to the discipline specific knowledge a graduate should also have ‘soft’ skills, these soft skills include communication, problem solving, project management and the ability to work in a team (Shadbolt, 2016)

For the average Computer Science graduate the importance of these skills can be seen in their daily work, as code will often be developed to solve a problem, whether that is in the form of bug fixes, or creating brand new software. A Computer Science graduate will also have to communicate with their client (their boss or a paying customer) or a wider team. Graduates should also be able to effectively apply interpersonal skills to further work with others in the organisations that they are in (Shadbolt, 2016).

In his review of business-university collaborations, Professor Sir Tim Wilson (2012) emphasises the importance of universities providing students with learning opportunities to gain and develop these skills. Each of these aspects of higher education is supported by assessment and feedback. Certification can only be provided once the summative assessment has taken place, showing the student’s competencies. To facilitate learning, assessment helps guide a student's understanding of a subject and to identify what their current progress is along with gaps that they may wish to fill.

To answer the research questions identified in Chapter 1 Section 1.3, this chapter explores the purpose of assessment and feedback within Higher Education and how it can be used to develop the desired skills within Higher Education. As the case studies of this research take place in a Computer Science school, this chapter also explores the importance of assessment and feedback within computer science specifically. Section 2.3 explores the current issues with assessment and feedback practices within higher education and how this may impact student perceptions are also explored. Following Section 2.3, Section 2.4 explores good practice of assessment literacy within assessment and affect at higher education level, followed by an exploration of the assessment of programming in Section 2.5. Section 2.6 outlines Ipsative assessment and feedback and investigates them as a potential form of assessment to help ameliorate the issues faced by students with regards to assessment and feedback. Finally, based on the literature explored, a set of feedback criteria are developed to help explore and identify best feedback practice.

### *2.2 Current assessment and feedback in higher education.*

#### *2.2.1 Assessment*

Current assessment and feedback practices in higher education have two generalised forms, summative assessment and formative assessment. The QAA defines summative assessment (Definition 3, page xi) as a form of assessment that indicates a learner's success in meeting the assessment criteria and learning outcomes of a course (QAA, 2018). This meeting of criteria allows a higher education institute to provide quantifiable value to a student's knowledge, abilities and competencies that they have gained throughout their time of studying. It provides both the learner and assessor with information on a student's progress. As a measure of performance, summative assessment often takes place at key learning points, usually within the middle and end of a module (Ibabe et al, 2009).

Summative assessment must be criteria-referenced. To be beneficial to the students, each requirement for the assessment must be clear and accessible to the students with no hidden features or tasks. It is not necessary that the students are able to reach these criteria, as that is the distinguishing factor of certification, but clarification for validity and fairness is important (Hughes, 2011).

Brown states that giving a student a classification is important to give them an idea of the stage they are currently in with their learning, however, for this to be achieved, the assessment needs to be valid (Brown, 2001).

Validity can be broken down into three main components (Wakeford 2003, pp60):

1. Face validity is concerned with the appropriateness of the assessment for the leaners and the audience. An assessment which is suitable for a master's degree student will not be suitable for a foundation degree student.
2. Construct validity is concerned with the authenticity of the assessment. Does it assess what it is claiming to assess?
3. Impact validity: is concerned with what impact the assessment will have on the students' knowledge.

A way to ensure that assessment is valid and useful to the student is to ensure that assessors use evidence of achievement that is clearly matched against criteria and learning objectives. This allows the students to understand what they need to do to gain the corresponding grade. For assessment to be fair, valid and accessible, it should have a clear purpose and be easy to read. There also must be a variety of the types of assessment given to students. For assessment to be applicable to all and inclusive, the teacher needs to have a variety of assessment methods within their course. Using just one form of assessment, such as exams or essay writing results in these forms reducing their accessibility (Brown 2005). When combined, clarity and validity give students a very specific understanding of what is required and leaves passing the assessment down to the knowledge and skills of the students. While assessment should be a central part of the learning experience it should also be noted that what is important is not the end product, but the process taken to get there (Black et al. 1998; Gibbs and Simpson 2003; Boud 2009; Newcastle University, 2019 a,b,c,d,e,f ).

The issue with summative assessment, however, is its mark driven, criteria referenced nature. Due to this clear understanding of what needs to be achieved to reach a certain mark, modern students tend to focus on "beating the mean" as their measure of success, rather than mastering the material (Wilson, 2006). Beating the mean becomes a problem when all assessment given to students is summative and high stakes. High stakes assessment occurs when the grade of the assessment is worth a large contribution of the total module mark and will have a significant impact in the resulting classification(Knight, 2002). The high stakes nature of summative assessment and the desire to 'beat the mean' provides a partial answer to RQ2 (What are the current perceived assessment and feedback issues experienced by students?). The focus on marks can lead to student dissatisfaction and disengagement if a student is unable to reach these standards.

This mark driven nature of students is often negated via the use of formative assessment. Formative assessment (Definition 4, Page xi) as defined by the QAA is assessment that focuses on the developmental purpose by giving feedback on their performance and how it can be improved (QAA 2018*)*. Formative assessment has five main methods of activity that centre around sharing success criteria with learners:

1. **Classroom questioning**. When students ask questions within the classroom discussing the material with their peers and teachers, they begin to develop the ability to identify what topics they have difficulty with and how to identify gaps in their knowledge.
2. **Comments only marking.** By providing comments only, students must focus on their skills and abilities and how to improve rather than the numerical value given to their learning. This provides students a way to experiment and make mistakes without having the pressures of a high stakes assessment.
3. **Peer assessment.** When students engage in peer assessment and giving feedback to others, it enables the student to begin to identify mistakes and errors which they may

not have spotted in their own work. It gives the student opportunities to constructively comment on work without a personal attachment.

4. **Self-assessment**. When a student is able to safety assess and criticise their own work it helps the students to identify frequently made errors. This however is only possible after a student has experienced this outside their own work. Self-assessing gives students the opportunity to identify their own strengths and weaknesses and target their learning accordingly.
5. **Formative use of summative tests.** Using summative tests as a way to benefit learners can help remove the pressure and belief that a summative test is to provide marks only. A summative test should also develop the students understanding of their own knowledge, not just to provide a certification.

(Black, 1998; Black and William, 1998).

An important aspect of formative assessment is the removal of high stakes marks; therefore, assessment becomes feedback-driven and developmental, with less focus on "beating the mean". It provides direction for both knowledge state and future learning (Petrovic et al. 2019). This removal of the marks focuses the students on their skills and helps the students identify their weaker areas and to regulate their learning in accordance with their skill set. Within computer science, formative assessment can be most frequently found in lab exercises. Sarkar (2006) explores a typical lab set up for computer science students using a weekly session with dedicated lab exercises. The students within Sarkar's study were Level 6 Networking students (second-year students) who had pre-existing programming knowledge. The aim of the lab exercises was for students to practice the theory and skills introduced in lectures. Sarkar reports that these lab exercises were well received by the students and that for the years that students completed these practicals the average cohort grade increased, compared to those that that did not complete them. For novice programmers, these practicals need to emphasise the role of both theory and practice (Thunė and Eckerdal, 2018). This helps the students to form the mental models needed to program, while also increasing the ability to self-regulate by having the opportunity to identify which parts of the theory the student feels weakest on. It is often the case that the skills learned through the formative assessment are the skills that will be used within the high stakes summative assessment.

Traditionally, formative and summative assessment have been identified as two critical pillars of assessment, separate and in some cases mutually exclusive, with prioritisation of formative assessment to help foster a 'for learning' environment (Taras, 2008). They are, however, intrinsically linked, both forms of assessment should support learning. In essence, all summative assessment should also be formative in that it provides direction to the learner (Hughes, 2011). While formative assessment should not have associated high-stake grades, it should give the students direction, and action plans towards the summative assessment. Daly gives a practical example of this, where programming lab exercises are given to students but are unmarked. Daly reported that as students are aware that the final assessment is a programming test, they use the formative lab tests to tailor what they learn and improve their programming skills (Daly, 2004). In their 2008 study, Taras sampled 50 lecturers from an education setting within a university located in England. This study aimed to investigate the link between formative and summative assessment. In their study, 38% of the valid responses reported that formative assessment and summative assessment are separate entities, with 40% of responses stating it was not and 10% of responses reporting "sometimes". The results of this study show some support to the fact that formative and summative assessment have been considered two separate tasks as opposed to one whole. These issues need to be resolved, if instructors view these two factors as different, so will the students, and therefore will tend to focus on the high stakes assessments.

### *2.2.2 Feedback*

The NSS results in chapter 1 (page 2) show that the feedback statements have a lower result than the assessment statements. To understand where potential issues could be (RQ2) current feedback practice needs to be explored.

When feedback is given on a summative piece of work, it often focuses on the assignment itself, as opposed to the skills that the student can develop. The feedback given to the student is in the context of the current assessment piece i.e. Where did the student go wrong? What criteria did the student not meet? (Hattie et al., 2006). This places the focus on the criteria of the assessment and how the student met it, with minimal reference to the skills that can be developed. The focus on the assessment criteria within the feedback may encourage the mark driven nature of students. Feedback on formative assessment often focuses on the skills of the student and the work that they need to do to close a gap in their knowledge (Hughes, 2011). The emphasis with formative feedback is on the work and the task at hand, rather than the marks and criteria missed. It is within this comparison of summative and formative feedback that we begin to understand why both need to be considered at the same time, as opposed to two separate pillars of the assessment structure.

Ott et al. (2016) break 'feedback' down into four major levels;

1. Task. The task-level shows the student which criteria they have or have not completed.
2. Process. The process level explores the students understanding of the subject, have they understood the material, or has the student used fact recall?
3. Self-regulation. At the self-regulation level, the student needs to be able to take control of their own learning, directing what they know and where they need to go.
4. Self. At the self-level, students must be able to identify their own patterns of work and behaviour.

Figure 1 shows how each task level aligns with summative and formative feedback.

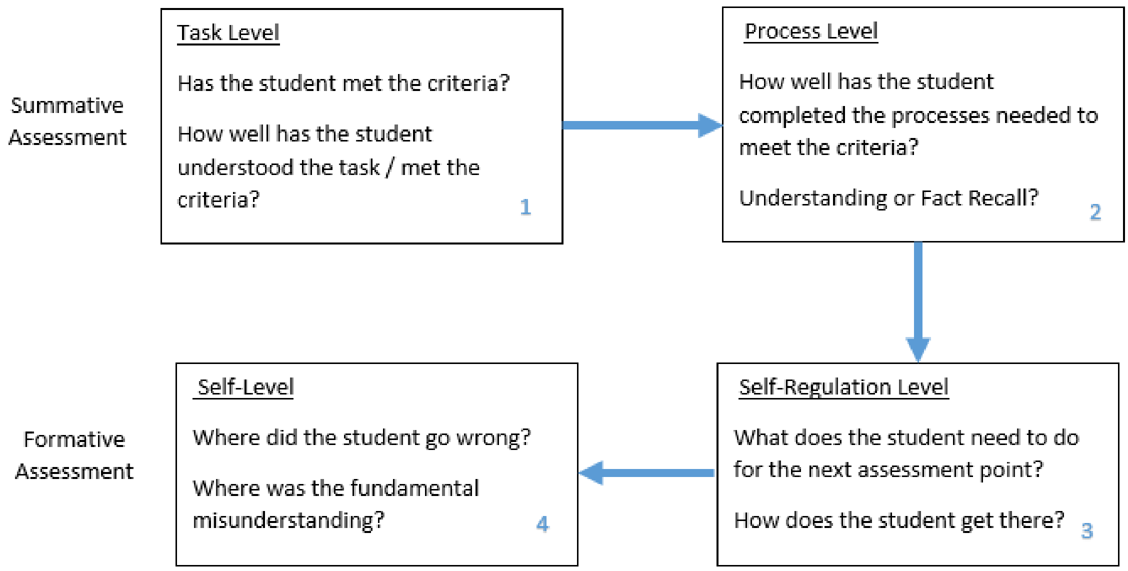


*Figure 1 Summative and formative assessment in the feedback structure*

Students should be able to use these four levels of feedback to answer three main questions: Where am I going? How am I doing? Where to next? If a student can answer these three questions the assessment then becomes 'for learning' as opposed to 'of learning' and helps to facilitate learning (Ott et al., 2016). This break down of feedback can also be seen in Black and Williams (1998) exploration of the functions of feedback. Black and William explore that feedback has specific functions that it needs to achieve.

First, feedback should identify where the students have made mistakes within their work. This takes place in level one the 'Task level' of Otts feedback levels. In level one feedback informs students if they have a misunderstanding of the base elements, such as is there a mistake in the syntax, or a misunderstanding of the purpose of the task. For Computer Science students, level one is where the focus is on the assessment criteria, did the student develop function x and did they use the correct constructs to do so, such as control structures. It shows the student how well they achieved what the initial specification set out to do.

The second purpose as defined by Black and William is to give feedback on the students' ability to use the knowledge that they have gained. This is seen in level 2, the 'Process level', where the feedback focusses on why the student may have made this mistake. Feedback at this level identifies any misconceptions the students have with the material. For a computer science student, the process level explores if a student used these constructs in an appropriate way. The feedback identifies the students use of control structures, methods and scope of the project. Topics which require more than a basic knowledge of the material presented.

It is within level one and two of Ott's feedback levels that that we see the summative aspects of feedback take place. The feedback often focuses on how the students can improve their work in relation to the assessment criteria. Levels three and four are where the formative aspects of feedback take place.

A third component to good feedback is that of a student's ability to reflect and improve from their feedback, identifying where they went wrong and what steps are needed to get to the next stage of their learning. This takes place in level 3, the self-reflection level. There is a possibility that students did not produce a high-quality solution due to a misunderstanding of what the assignment asked, the student may have focused on a superficial aspect instead. It is here that the student must begin to use their own assessment literacy skills to learn how to interpret specifications (Black and William, 1998).

In level 4, the 'self-level' students are required to understand what mistakes they made within their studies and how they can structure their own learning. It requires a student to understand how they work and how they could improve their work in the future.

This section identifies what current practice should be within assessment and feedback, however as identified in Chapter 1, students are currently dissatisfied with these practices. Section 2.3 identifies some of the common problems within assessment and feedback and how they can impact this student dissatisfaction.

### *2.3 Common problems in assessment and feedback*

To answer RQ2 (What are the current perceived assessment and feedback issues experienced by students?) it is important to first understand what the theoretical identified issues regarding assessment and feedback within the literature are.

A common issue regarding assessment and feedback is the modularisation of current university programmes (Ferrell, 2012). Modularisation can often cause student to miss the connections between the skills learned in each module, resulting from a disconnection from the learning process as a whole. This disconnection often occurs towards the end of the module when high stakes summative assessment is typically completed. This has a very little positive impact on either function of higher education (facilitation of learning and providing certification), as there is minimal facilitation of learning and the numerical value of the knowledge given to the student is very circumstantial (Ferrell, 2012). Due to this, students can feel overworked and see little point to the final assignments and how they connect, which may cause a disengagement. Modularisation may have caused students to view assignments as entities on their own, rather than components of a wider degree

Assessment literacy or the lack of assessment literacy is an issue commonly seen in undergraduate students. Smith draws attention to the issue raised by Francis (Francis, 2008) that states there is often a difference between what the student is being assessed on and what the student believes they are being assessed on. The impact of this issue, Smith explores is reduced by the use of well-defined assessment criteria. This, however, has introduced a different problem, the mark driven nature of students (Smith et al, 2013). The mark driven nature of students is perhaps exacerbated by the criteria referenced assessment pieces most assessors focus on. Alongside these criteria, the common use of feedback as mark justification, from the perspective of the student if not the intention of the assessor, has the student focus on what will help them pass the assessment as opposed to master the material. Commonly known as performance goals, this type of behaviour has the student focusing on what the assessor desires as opposed to mastery of the knowledge. Yorke defines this as "learned dependence" (Yorke, 2003).

An example includes the work by Matshedisho who investigated the usage of marking rubrics as opposed to criteria referenced work. 367 students took part in Matshedisho' study completing compulsory tutorials in classes of 15-20 students across 3 bachelor's degrees in the university of Johannesburg (Matshedisho, 2020). Rubrics as seen in Figure 2, are criteria-based marking schemes which are often presented in grid structure. Rows define and break criteria into sub parts and columns identify the grading criteria (Jones et al, 2017).

| | 1st Class Outstanding<br>8+ | 1st Class Mark<br>7 | 2:1 Class Mark<br>6 | 2:2 Class Mark<br>5 | 3rd Class Mark<br>4 | Failing Mark<br>1 - 3 | Missing<br>0 |
|---|---|---|---|---|---|---|---|
| Understanding of Disaster<br><br>60% | Will have five different case studies which show an excellent understanding of the computing disaster where discussion:<br><br>Is detailed enough to show the individuals understanding of what happened but still concise enough that there is no 'waffle'. | Will have five difference case studies which show a good understanding of each computing disaster where discussion:<br><br>Is detailed and concise, but may read as if they are paraphrasing what was said on the reference. There may be one or two mistakes. | Will have five case studies which show an understanding of each computing disaster where discussion:<br><br>Is detailed, but may read as if they have limited understanding in parts. There will be a few mistakes throughout. | May have four case studies which show an understanding of each computing disaster where discussion:<br><br>May not have relevance and lack clarity of what they are discussing.<br>It may be difficult to read and have a large number of mistakes throughout. | May have two – three case studies which show little understanding of each computing disaster where discussion:<br><br>Discussion may not have relevance or even be strictly computing related, there may be confusion throughout with many mistakes. | One or Two case studies rather than the required five. Each case study shows the student can describe what happened by paraphrasing what they have read. | Missing any description of ethical disaster. |
| Analysis of BCS Code of Conduct<br><br>30% | Analysis of relevant BCS code of conduct, which makes sense throughout, without error. | Analysis of relevant BCS code of conduct, which makes sense throughout, with occasional error.<br><br>May have missed a code of conduct item out of the analysis. | Analysis of relevant BCS code of conduct, which does not always make sense, with errors. Made analysis for all case studies.<br><br>May have missed a code of conduct item out of the analysis or not always shown an understanding. | Analysis of BCS code of conduct, which does not make sense, lots of errors. Made analysis for all case studies.<br><br>May have missed a code of conduct item out of the analysis or not always shown an understanding. | Analysis of BCS code of conduct, which does not make sense, lots of errors.<br>May have not made an analysis for some of the case studies.<br><br>May have missed a code of conduct item out of the analysis or not always shown an understanding. | Made an analysis once or twice, has made mistakes throughout analysis. | Missing any description of BCS code of conduct. |

*Figure 2 Rubric Example (Heels L, 2016)*

Matshedisho found that Rubrics can be confusing to a student and can influence their responses to a task, they are, however, reliable, and consistent methods to use. In their study the aspects of the rubric that students all found best were the specific, unambiguous parts. However, even when using rubrics, students were still very mark driven, focusing on crossing grade boundaries and the clarity needed for each one. Further to this there can often be a reluctance on the part of the student to engage with material that counts little towards the final grade, which due to the increasing use of formative assessment and feedback can prove problematic (Matshedisho, 2020).

Another issue experienced within higher education is the lack of assessment diversity. With the increasing student numbers and resource-demanding nature of current-day programmes, assessors will often fall back on methods within which they are most comfortable, such as essay writing and examinations. It is important that students are provided with a range of assessment techniques that are fit for the topic that they are assessing (Ferrell 2012; Brown and Knight, 1994).

Wilson suggests that there is a lack of communication between teachers and students. Teachers often assume students know things, and students are prone to avoid stating that they do not, due to a lack of confidence. This lack of communication can cause student dissatisfaction, the lack progress and misconceptions regarding basic material can develop a fixed mindset within students, and potentially lead to an incorrect understanding (Wilson, 2006).

One of the more common misconceptions regarding feedback is what consists of feedback. When answering the NSS, students often think of feedback as written feedback and there is often less emphasis placed on verbal feedback given to the student (Nicol, 2010). An example of verbal feedback is discussions within practical classes where the assessor has an opportunity to give rapid feedback on a students' work (Yorke, 2003). This reliance on written feedback can often be detrimental to the student as written feedback should be used as part of a larger feedback plan, and should be used to supplement the learning with contact with teachers in the form of tutorials (Nicol, 2010).

An issue arises when students do not understand the use or meaning of feedback. Tutor comments are only useful to the student when they are understood. The students may also be unfamiliar with the "disciplinary discourse" i.e. phrases and vocabulary only used within the specific discipline (Winstanley, 2017, pg70). This also depends, however, on the importance of the purpose of the feedback. If the desire is to point out the student's mistakes, this is less important. However, if the desire for feedback is to assist with longitudinal development, it is crucial that the student not only understand why they are getting feedback but also to understand the feedback itself. Furthermore, as always, greater quantity does not always mean greater quality. Feedback is only useful if its aim is to help close a gap in a student's knowledge (Winstanley, 2017, pg71). It is important that the student obtain useful and clear feedback as opposed to pages of unclear feedback (Price et al., 2010). Therefore, encouraging students to focus on feedback as an important part of learning becomes a challenge, more so when students are often unaware of what feedback is, what it contains and how to use it (Nicol, 2010).

Due to these factors it is important to understand how assessment can be developed to ameliorate these identified issues and improve the student experience.

### *2.4 Assessment and Feedback in Practice*

Section 2.4 explores the current theories of good practice for assessment and feedback within higher education, such as Blooms Taxonomy and Kolb's learning theory, to answer the first part RQ1, (What is good practice in assessment and feedback in undergraduate computer science in the UK?).

#### *2.4.1 Bloom's Taxonomy*

To understand how best to assess a student and give feedback to help a student improve, it is important to understand how students learn best, an important starting point to do this is Bloom's Taxonomy (Bloom et al., 1956). Blooms Taxonomy created in 1948 was developed as a method to classify educational goals for student performance evaluation. Bloom's Taxonomy was later revised by Lorin Anderson to place a focus on actionable tasks to encourage engagement, however the principles remained the same (Coffey, 2008). To date Bloom's Taxonomy is consistently used in classrooms at all levels of the educational process and is therefore an important aspect to explore when considering the development of assessment and feedback.

Bloom categorised six main levels that make up the Taxonomy of Educational Objectives. These were as follows:

- **Knowledge**, in the taxonomy is defined as the student's ability to recall facts. There is a limited application of the knowledge at this stage. This includes the ability to recall a specific item of information but also to be able to identify where this knowledge fits in the broader scheme. Bloom notes that knowledge is involved in all the other levels of the taxonomy, but as a component as opposed to the whole level. Knowledge, however, also involves the student being able to organise and judge the material they are presented with. They should be able to identify the techniques and methods used within their field of study.

When working at the knowledge level computer science students are required to identify the syntax and constructs of the code. The student should be able to answer questions that require basic recall, e.g. to list, define or describe constructs within the programming language that they are learning.

- **Comprehension** is a skill in which the student is expected to be able to communicate their knowledge with others. They are also expected to be able to understand what is being communicated with them and achieve this with all forms of communication. The first aspect of comprehension is the ability to take the information given and give meaning to the individual components. The second aspect is the ability of the student to interpret the information they have been given, including linking the ideas and their importance together. Finally, the student should be able to extrapolate from the data given, making predictions and estimations; this will also involve students identifying their own beliefs regarding the subject.

Once a computer science student has progressed onto the Comprehension level, they should be able to effectively identify the constructs and theory identified within the knowledge level and to explain the functions of a piece of code.

- **Application** requires that a student applies the knowledge gained in the comprehension level to any given situation. There can be two ways a student applies their knowledge, the first being the student recognising the problem to guide their actions, then using the abstraction of the knowledge and comprehension gained to solve the problem. The second form of abstraction occurs when the problem is unfamiliar to the student, but they identify familiar aspects to help solve the problem.

In the application level, when a computer science student is presented with a problem, they should be able to apply the concepts learned or use patterns that they are familiar with to solve it. Examples would be to modify a loop or to solve a problem similar to one they have been given in class. Scott suggests this could take the form of measurement conversions e.g. if in class they have converted meters to kilometres, in a practical they should be able to convert degree Centigrade to Fahrenheit.

- **Analysis**, in this level, the role of the student is to not only use the information to solve problems but to also break the information down into its individual parts and explore the relationships between them. This skill extends beyond the ability of a student to identify facts but to also "draw conclusions, distinguish fact from hypothesis and identify conclusions from formal statements" (Bloom, 1956). The student should also be able to identify relevant information and identify assumptions that are not already mentioned.

Working in the Analysis level, a computer science student should be able to break a program down into its given parts. When given a problem that needs to be coded a student should be able to identify what classes and methods are required to create a functioning solution.

- **Synthesis** involves the combination of each individual elements to form a solution or idea that was not previously clear. This can be combining new information (new parts) with pre-existing parts to gain a fuller understanding. It is here, where the student needs skills beyond subject domains and need to use creativity in their knowledge construction. The creativity explored is not a full creative liberty in the student perspective as it still needs to be rooted in fact. The student does, however, need to think outside the box with how the ideas can be connected.

It is when the transition to the synthesis level begins, that a mastery of the material is shown. When a student reaches the synthesis level, they should be able to create a program using the skills developed to solve a 'new to them problem'. It is at this stage a student should be beginning to develop their knowledge of interfaces and inheritance.

- **Evaluation** is a combination of all the previous levels in the taxonomy. Students working at this level need to make judgements about the value of the knowledge or items in question. These judgements can either be qualitative or quantitative with the criteria being student-driven or externally given.

Finally, within the evaluation level, students should begin to compare the programs with each other. When given two programs that have a similar function a student should be able to identify which one has the better quality, due to efficiency and best coding practices.

Source (Bloom, 1956; Scott, 2003; Thompson et al, 2008; Starr et al 2008).

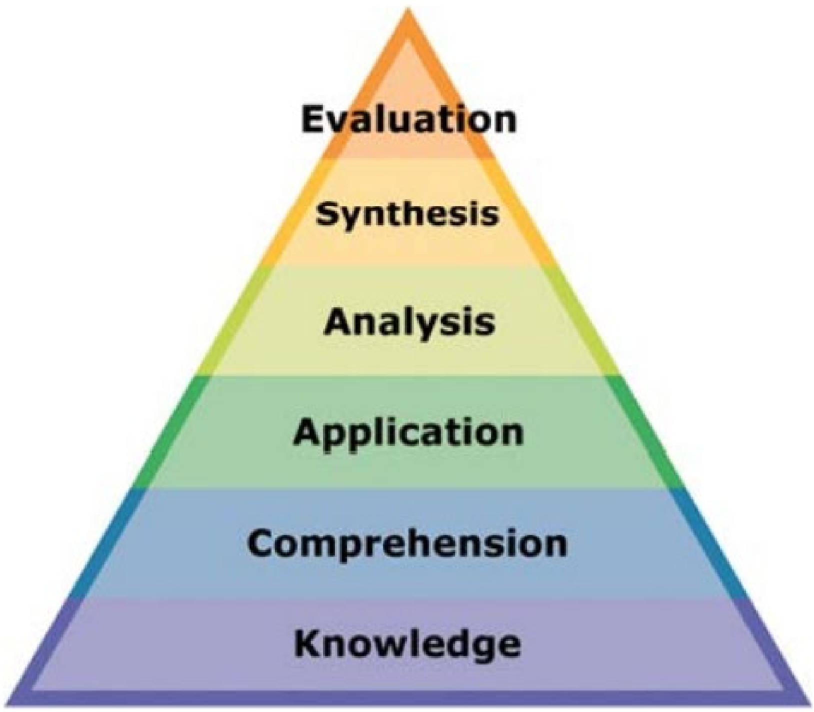


*Figure 3 illustration of Bloom's Taxonomy (Coffey, 2008)*

The type of assessment that a tutor develops depends on which of these levels is the intended target. For example, while a first year computer science student may be expected to create a *FOR* loop to solve a problem (Synthesis level), they would not be expected to evaluate the efficiency of a program (Scott, 2003). Within higher education, specifically the undergraduate components of higher education, a tutor should be expecting a stronger student to be working at the upper levels of the taxonomy (Lister and Leaney, 2003). A problem, however, occurs in the development of assessment, tutors can often assess the lower levels and the upper levels of the taxonomy but not the intermediate levels, therefore creating a 'double bump' in the test scores of students (Scott, 2003). This could be a potential cause of the dissatisfaction reported by students in the NSS. If there is a missing component to the assessment structure e.g. the application and analysis components of the taxonomy, that would suggest that only the strong students would be successful on end of term exams, with those ranging from weaker to average students being dissatisfied. This has introduced another issue as Burgess identifies, a student who can excel at exams, but avoided writing programs, thus not completing the analysis or application levels, may have never written a program but could still pass the programming course (Burgess, 2008).

Nilson (2013) defines strategic, cognitive, and self-knowledge and why these are important to consider when assessing students, Table 2 shows how these knowledge levels map to Bloom's Taxonomy. Strategic knowledge is concerned with learning specific items, tasks and heuristics for the sake of the current assignment. This is also termed as surface learning where a student intends to complete the work with as little effort as possible. Surface learning is an approach which students apply with the aims of passing the assessment without gaining a mastery of the material (Definition 6, Page xi). This is where levels one and two in the Bloom's taxonomy, (as seen in table 2), are focused. Knowledge of cognitive tasks requires a high level of assessment literacy meaning that the students need to think about the task ahead and truly understand what is required. When using cognitive strategies, the student applies the knowledge that they have gained from the base surface level to a given situation, covering levels three, four and five in Bloom's taxonomy. Self-knowledge is used within the evaluation level of the Taxonomy. The ability for a student to evaluate their work requires them to know their own strengths and weaknesses. It is looking at the work that needs to be done and understanding what learning the student is required to do (Nilson, 2013).

| Blooms Taxonomy | Nilson Knowledge levels |
|---|---|
| 6.Evaluation | Self Knowledge |
| 5.Synthesis | Cognitive Knowledge |
| 4.Analysis | |
| 3. Application | |
| 2.Comprehension | Strategic knowledge |
| 1.Knowledge | |

*Table 2 Nilson knowledge levels mapped to Blooms Taxonomy*

The approach that a student takes to learning will have a large influence on where in the taxonomy they are. Struyven explores two different approaches that a student can take during the learning process, surface learning and deep learning (Struyven et al., 2005). The student using a surface learning approach finds a task to be an obstruction and unwelcome addition to their workload. This student will often work within the first two levels of Bloom's taxonomy (knowledge and comprehension). While this allows for an understanding of the basics in a subject, it does not allow the student to grasp what is being taught fully. It is only when a student progresses up the levels of Bloom's taxonomy and applies a deep learning approach that they can begin to connect skills with other modules and assignments, as well as understand what direction their learning needs to take.

This understanding of surface and deep learning can inform what types of assessment will be best used depending on the type of knowledge that is assessed. An example of a question for a surface learning approach is multiple-choice, e.g. 'which is the correct definition of term x?' These types of questions can often be found in the lower level areas of learning, such as the first stage of an undergraduate degree. These are used to identify if a student can successfully use the material in a basic situation, before moving onto the mastery of the material e.g. a student cannot design a program until the student understands what classes and methods are. Once a student understands this, they can learn how methods and classes interact with each other. They can then reflect on what they have done and begin to apply and analyse the knowledge they have gained. A student who has applied a deep learning approach would be able to write a program to solve a given problem without external prompts. Bloom's taxonomy, however, is not the only learning theory that should be considered, Kolb's Experiential learning theory is an additional factor to consider.

### *2.4.2 Kolb's Experiential Learning Theory*

Kolb explored student learning as a process of experience, where the emphasis with the learning is placed on the journey a student takes to get to the end product. Kolb describes learning as a process that remoulds new knowledge to a students' existing knowledge, based upon the experiences that they have. This can be thought of as forming a habit, the more students repeat or experience a form of knowledge, the more they will remember it. Kolb explores the idea that all students have a knowledge state that they enter the learning process with, and due to the experiential nature of the learning process, each student is different (Kolb, 2014). This may lead to student dissatisfaction. The BSc program in Computer Science at Newcastle University does not require a prerequisite of programming knowledge on entry, therefore students enter the course with various levels of programming experience (Newcastle University, 2019b). Student A who has no experience starts their learning at the lowest level of Bloom's taxonomy. When they enter the course, they are beginning to learn the syntax of their chosen language. Student B, who enters the course with experience, may have already mastered this level of knowledge and begin to work on the learning how to use classes and methods, thus progressing to levels 3 and 4, application and analysis. This disparity between the students' confidence and their success at completing the assessment criteria, may cause dissatisfaction and a loss of confidence, which can lead to demotivation.

Figure 4 shows the four stages in the learning cycle as defined by Kolb. The four stages are;

1. **Concrete experience,** in this stage students experience a new knowledge set or gain more information within a pre-existing knowledge set. This must be done without preconceived notions.
2. **Reflective observation**, students must use their experience and the knowledge of other experiences, to reflect on the information.
3. **Abstract conceptualisation**, the student must then integrate this new knowledge into their existing knowledge.
4. **Active experimentation** is where the student must use this knowledge and experience to solve new problems.

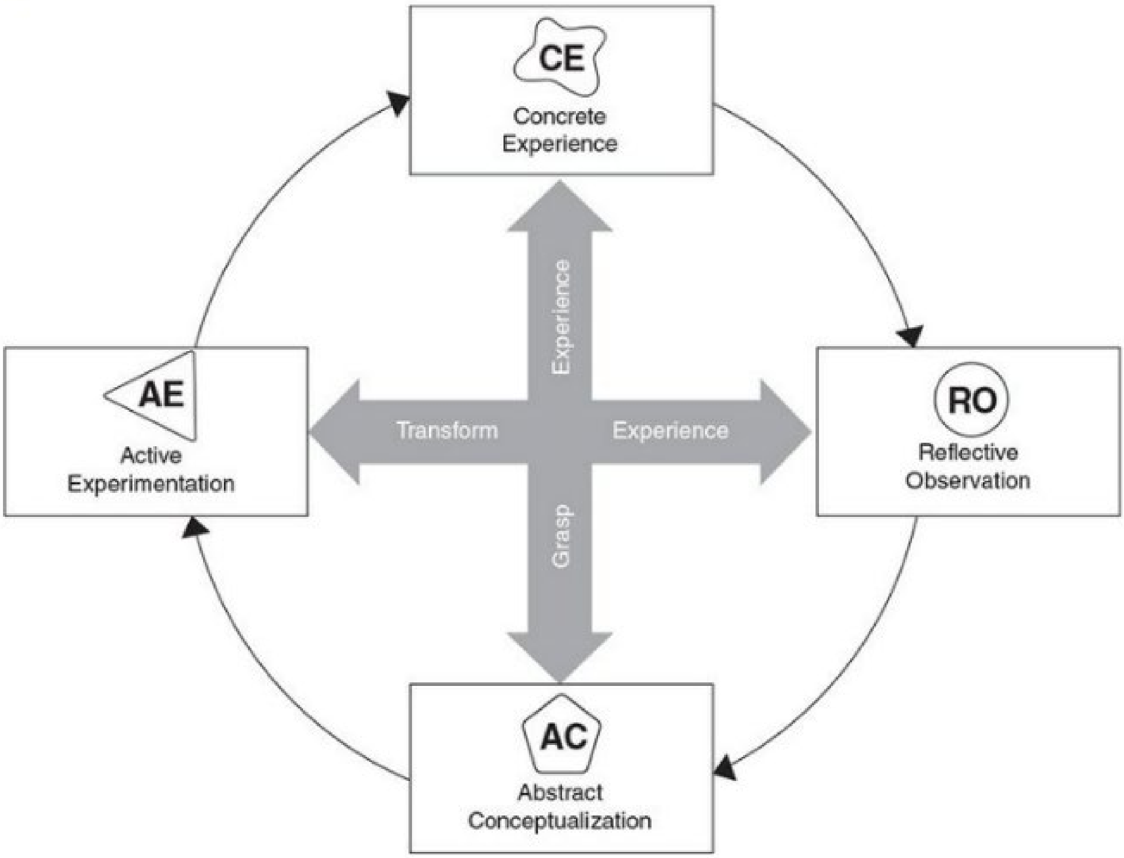


*Figure 4 Kolb's learning cycle (Kolb, 2014)*

Alderman and colleagues (2005), state that learning is a continual process, which is reflected in Kolb's learning cycle. Kolb's cycle draws emphasis to the reflection stages of learning, and how a student is in a constant cycle of learning and improving.

Kolb draws attention to the claims that the learning cycle is too simplistic and too individualist, such as those from Desmond and Jowitt (2012). Kolb however, also identifies that this work was never aimed at replacing all teaching structures but was a framework on which students could build to master their own learning. The framework could also help teachers to understand how each individual student learns and how to best teach them (Kolb, 2014). However, regardless of the types of assessment and learning theories the assessor uses, one of the most influential factors on how a student performs is the student mindset.

Dweck and Legget suggest that there are two types of students in the class, the 'fixed mindset student' and the 'growth mindset student' (Dweck et al., 1998). From the instructors' perspective, these are respectively known as the 'fixed IQ' and 'untapped potential' students. The fixed IQ student believes that they have a certain level of intelligence that cannot be changed or grown. The untapped potential student believes that their intelligence can be improved and developed. The untapped potential student explores the knowledge that they have, the situations that they are in and will associate new information with skills that are learned as opposed to natural ability (Dweck et al., 1998).

It is therefore important that the criteria of the assessment are clear and that a student can identify how to complete and reach it. A combination of clear and understandable assessment criteria, as well as feedback that informs the students where and why their work was incorrect, gives the student the tools to improve their work, from this students can see that they can develop their skills and therefore develop a growth mindset.

A fixed mindset in a student combined with an unclear assessment criteria could lead to student dissatisfaction. If a student believes that they can only achieve certain parts of the assessment and it is not clear what those parts would be, this could cause a student to feel that it is not possible to complete the assessment and will therefore disengage with the material.

It is important to keep these two aspects in mind when developing an assessment, specifically when considering the fixed mindset student. Dweck and Legget describes students in two different ways, the first is those who avoid challenges, are concerned with their behaviour and have a fixed mindset. The second are students who embrace challenge and persist in the presence of an obstacle. Additionally, in their study, Dweck and Legget outlined two different goals types, performance goals and learning goals. Performance goals focus on the grades of the student, whereas learning goals focus on the effort used to learn the material (Dweck et al., 1998). Struyven reported that the type of knowledge a student aims for changes based on the students' perception of the work. For example, in their study, if the students found the workload to be too high, or that the task itself was not a suitable/applicable one, students disengaged and practised a surface knowledge mindset (Struyven et al. 2005). Black also suggests that students who have this "fixed IQ" mindset will stop attempting to learn if they feel that they are unable to (Black,1998). Therefore, when answering RQ1(what is good practice in assessment and feedback in undergraduate computer science in the UK and how does current practice at

Newcastle University compare?) encouraging a student to develop a deep learning mindset and to focus on learning goals can be considered good practice to help students succeed in their studies.

### *2.4.3 Encouraging a deep learning approach*

Encouraging a deep learning approach is important as Struyven found that students who believe intelligence is fixed were more performance orientated (Struyven et al, 2005). When considering the aims of higher education, the fixed mindset student will achieve a certification but will not take action to facilitate their learning. To help develop the facilitating learning aspect of higher education, it is necessary to encourage students to operate at a higher level of Bloom's taxonomy, the 'self' levels as Nilson describes. To do this, students should be encouraged to Self-regulate.

Ott and colleagues (2016) identify that Self-Regulation has multiple stages; the first involves students identifying an area of weakness. This can be in the form of topics not covered, misunderstandings or a basic understanding of a complicated structure. An example of this, for a computer science student could choosing an appropriate control structure and understanding the nuances of choosing between *FOR* statements and *WHILE* statements. The next stage of this cycle would be to identify how to overcome this weakness. The student would use resources such as lecture notes or online materials to review the content surrounding this topic. It is here that the student would apply the learning strategies that best suit their individual needs. Once the student feels that they have a strong grasp on the material, they could then test themselves again, ideally with different material to the initial test. Finally, the student should discuss their progress with another or get feedback on work they have done, this can be feedback on a summative piece of work. Selecting their own strategies to achieve those goals will help focus their attention to self-monitor their progress and adjust the strategies accordingly. Reflection on their failure or success at the end of this cycle will assist with improvements in the future (Ott et al, 2016). Changing a student's mindset from a fixed to growth mindset also requires an understanding of self-regulation. Nilson discusses the benefits of having students become proficient with the self-regulation skill set, that is, the ability to monitor, evaluate and manage their own learning. By using these skills to increase the depth of thinking and develop a student's ability to focus, a mastery of the self-regulation skill set can lead to increased performance and achievement for the student.

It is important to develop these skills as students do not often use feedback in the way that the tutor expects. It is also common that students do not often feel comfortable approaching a tutor to get help with their assignments, therefore being able to assess their own learning is important to a student's productivity (Hughes 2011). Chapter 4, Section 4.3 investigates how feedback is used amongst students at Newcastle University to identify what students perceive current practices are and if the Self-Regulation and reflection skill sets are used by students when exploring their feedback.

There are multiple ways of assisting students with increasing their self-regulation skills. A challenge for self-regulation is identifying the gap between novice skills and expert skills. A way to do this is via the use of sub-goal learning. Sub-goal learning is a common teaching practice within programming. The students break each procedure into sub-goals and solve a piece of the problem at a time. Self-explanation, on the other hand, encourages students to process information via the use of problem-solving. This allows the student to make a connection via each individual component (Margulieux, 2017**).**

Asking students to explain what they have done and the problem itself encourages deeper learning by connecting the principles and skills of what they are doing. It also helps learners to understand new information by connecting old learning processes to new ones. The self-reflection within a learning process is crucial to a student achieving deep learning. Abstracting a problem to a general solution, which can then be adapted to future work and allow the student to include previously learned skills can often be a difficult concept for a student to grasp. These skills, and the ability to synthesise the work with relation to each other, are the requirements to operate at the higher levels of Bloom's taxonomy (Bloom, 1956). Assessment and feedback should help encourage this development. It should be clear to students how each assessment builds upon the previous one. This can be achieved through the use of clear assessment criteria and objectives. Due to this feedback to a student should contain both specific feedback and general feedback. Feedback should tell the student what they did not successfully do and why but should also generalise this for future improvements. Both are needed to allow the student to correct mistakes but also to see how these features can connect with other assignments.

It is important to note that a student with deep learning of a subject can also answer questions based upon a surface knowledge. If a student can understand the meaning of an essay question, they can answer a multiple-choice question, however, the reverse is not true. It is more important for a student to understand the meaning behind the text rather than to memorise the text itself. Evidence shows that those with good learning skills and high confidence in their academic ability seem to prefer the essay-style questions (Struyven et al, 2005). Within computer science, these essay-style questions present themselves in the form of program writing to solve a given problem.

A further development of self-reflection is asking students to explain what they have done and the problem itself, it encourages deeper learning by connecting the principles and skills of what they are doing. It also helps learners to understand new information by connecting old learning processes to new ones. The self-reflection within a learning process is crucial to a student achieving deep learning (Margulieux, 2017).

### *2.4.3 Developing assessment within higher education*

An important aspect to consider when developing an assessment is the perceptions of students, along with the impact that these can have. Struyven explores how a student's wrong perceptions of assessment, along with expectations, affect their learning experience. The expectations of the students towards their assignment can influence the type of learning they do. They found that students who expected a multiple-choice assignment take a surface learning approach, whereas those expecting an essay style question take a deeper approach. However, it is more common that a student will attempt to take a surface learning approach, to learn just for the current assignment over a deeper approach for long term learning (Struyven et al., 2005). Additionally, Ihantola and colleagues (2010) found that when aware of what will be assessed, students will often target their learning towards that detail and practice a surface learning approach, therefore not gaining a true mastery of the material. If students are not aware of exactly what will be assessed they will be unable to tailor their learning and therefore practice a deep learning mindset (Ihantola et al, 2010).

**Question 3 [4 marks]**

Consider the following Java fragment:

```
for (int i = 1; ??? ; i += 4) {
  System.out.print (i + " ");
}
```

What output would this fragment produce when executed if the following Boolean expressions were placed in the position indicated by ???.

```
a) i == 1
b) (i<8) || (i%7 != 0)
```

*Figure 5 Constructed Response Question (Nesbit, 2019)*

A debate in the literature regarding assessment, as well as students taking a surface learning or deep learning approach, is the usage of constructed-response (CR) or multiple choice question (MCQ) assessments. Traditionally, to help develop a deep learning mindset, CR questions are used (Belo et al., 2019). A CR question (Figure 5) is where a student is asked to give a direct answer without external guidance and is believed to require a higher cognitive load than that of a multiple-choice question (MCQ).

Often favoured by assessors CR questions produce a heavier cognitive workload for students, whereas MCQs (Figure 6) are typically viewed to produce a low cognitive load (Nicol et al., 2007). This lower cognitive workload can often be due to students guessing the correct answer as opposed to completing the question themselves. This, however, is not always the case, while it is true that students can sometimes guess at multiple-choice questions, when developed well, multiple-choice questions can require cognitive work (Nicol, 2007). Belo also identifies a study Roberts where MCQs were used to assess Computing Science studies and were found to be an effective way for students to learn and evaluate their progress, however it is also identified that this needs further research (Belo et al, 2019).

7. What will be the output of the following Java code?

```
class average {
    public static void main(String args[])
    {
        double num[] = {5.5, 10.1, 11, 12.8, 56.9, 2.5};
        double result;
        result = 0;
        for (int i = 0; i < 6; ++i)
            result = result + num[i];
        System.out.print(result/6);

    }
}
```

a) 16.34
b) 16.566666644
c) 16.46666666666667
d) 16.46666666666666

*Figure 6 Multiple Choice Question (Sanfoundry, 2011)*

Nicol (2007), investigated a technique by Gardener Medwin, where confidence selectors are ranked alongside the MCQ answer. When answering a question students give a confidence score on a three point scale of 1, 2 or 3. When answering correctly students gain marks based on their level of confidence, Figure 7 shows how the level of confidence a student indicates effects their marks. Should the student indicate a high level of confidence and have an incorrect answer they receive a -6 mark penalty.

| Degree of Certainty | Low | Medium | High | No reply |
|---|---|---|---|---|
| Mark if correct | 1 | 2 | 3 | 0 |
| Penalty if wrong | 0 | −2 | −6 | 0 |

*Figure 7 Gardener Medwin Confidence Marking*

This encourages reflection from the student and removes the likelihood of a student guessing the answer while also indicating that they are confident. There is still a risk that students guess but less so that they guess and mislead their confidence level. This also has the benefit of allowing the teacher to identify the common areas of low confidence within their students. If students are mark driven, this penalty would reduce the number of students guessing and stating high confidence. This technique, however, also reduces the validity and consistency within marking. The risk is also present that students would not identify high confidence due to the fear of losing six marks. This specific method of confidence selection will turn into a risk analysis for students. However, the self-reflection of students concerning confidence in the questions that they are answering is a useful factor to consider when developing questions.

### *2.4.5 Developing good feedback in higher education*

Just as important as good assessment, is developing good feedback that is beneficial to the students.

Laurillard (2002) defines four characteristics that underpin effective feedback:

I. It should be adaptive, that is, contingent on students' needs.
II. It should be discursive, rich in two-way communicative exchanges.
III. It should be interactive, linked to actions related to a task goal and reflective.
IV. It should encourage students and teachers to reflect on the 'goal–action– feedback cycle'.

The goal-action-feedback cycle allows a student to gain an update on how they are performing. A student should set a goal for planned changes and future skills/knowledge to develop, work towards these goals before submitting their work and gain further feedback (Ott et al., 2016; Laurillard, 2002).

The purpose of feedback is to help students understand concepts and ideas as well as to apply their understanding of learning tasks. For feedback to be useful, it must trigger inner reflection, Nicol states "it is assumed that the overall purpose of the feedback process in higher education is to help students develop the ability to monitor, evaluate and regulate their own learning" (Nicol, 2010). It is important that students and teachers are clear as to what the assignment is asking. A lack of clarification between tutor and student can lead to a lack of understanding of the feedback given. Even if said feedback is detailed and clear when the students lack understanding about the purpose of the assignment, it can lead to difficulties with the application of the feedback.

While it is clear what feedback should achieve, it is difficult to know how to achieve it. To help students develop self-evaluation and self-reflection skills, there must be two aspects to consider. The first is the correctness standpoint i.e. does the feedback give good points for the student to develop? The second is the personal standpoint, i.e. is the feedback useful to the student in their learning? Great feedback that students do not understand or do not believe is in an appropriate form, becomes less useful. However, written feedback should be used as part of a larger feedback plan. The written feedback should be used to supplement the learner teacher contact in the form of tutorials (Nicol, 2010).

In 2013 the Quality Assurance Agency (QAA) investigated students "expectations and perceptions of the quality of their learning experience and the academic standards of their chosen programmes" (Kandiko et al, 2013). This study involved 150 UK students spread across 16 institutions. One of the findings from this study was that students were more concerned with feedback quality and usefulness in feeding forward than the timing of feedback. The QAA results, therefore, indicate that students prefer well-written, useful and usable feedback in preference to quick and ineffective feedback. These results suggest that the priority of feedback should be on the quality and how useful to the students it is, not institutional time targets. Feedback should be able to guide students on how to improve, without which, it becomes ineffective. However, it is just as important that the feedback be given in time to be useful for following assignments.

According to Ferrell, valuable feedback "helps learners evaluate their ability, appreciate their current progress, plan development and encourage self-reflection." (Ferrell, 2012). Whilst there are frameworks for summative feedback, i.e. feedback describing mark justification, there are few frameworks for formative assessment and feedback practice (Ferrell, 2012). Despite reported student preferences, universities seem to have a preoccupation with the timing of feedback to students, in that it should be both timely enough, and so that students can learn from it in preparation for their next episode of assessment. Duncan suggests that feedback should always have an element of feeding forward, therefore informing a student that they have poor academic writing, does not help. Identifying what specifically makes the work poor and how the student can fix it is more helpful (Duncan, 2007).

The main component of feedback, in general, is teachers using information gathered about a student via assignments to make a judgement on the current standard of the students learning and how they can progress. This occurs in both formative feedback and summative feedback. Students should always evaluate what is learned from the teacher and therefore, should have a constant loop of student-teacher interaction (Torrance and Pryor, 1998). This student-teacher dialogue allows the student to operate at all four levels of the assessment and feedback structure shown in Figure 1 (page 15). While teacher guidance is a needed input for students to clarify misunderstandings in higher education, it is just as important that the student can experiment on their own, find ways of improving and fill in the gaps of their knowledge. Once a student has achieved this, they start the loop again with the following assessment point. This dialogue allows students to begin to understand what their feedback and assessment are for. It helps reflect on what is important and how the feedback links with other assessment and modules. It can be the case that there is a misunderstanding of what the important focus of the feedback should be. It encourages students to not only reflect on their work but to also decide what is important to clarify with the teacher, and what they need to work on alone. Compared to the common feedback delivery method of giving a student feedback without further follow up, this communication helps reduce the impact of two of the identified problems in Section 2.3 i.e. (i) It removes the issues of teachers assuming that their student have a higher knowledge state than they currently have (ii) it also reduces the issue of students not understanding what feedback consists of, while also clarifying any misunderstandings that the student may have.

### *2.5 Assessing Programming*

#### *2.5.1 Assessing programming skills.*

Section 2.4 discussed good practice in regards to general assessment and feedback to answer RQ1. While the findings from that section still apply, there are some specifics that need to be considered for assessing programming. When investigating how to assess code, it is important to understand how programming courses are structured. Programming courses are often knowledge-driven and apply a focus on learning the syntax and structures of the language. An important aspect that is often lacking, however, is the application of the knowledge.

Furthermore, it often the case that when assessing programming, what is assessed is the product such as an exam script and not on the process, as the required data needed is difficult to collect for large quantities of students (Bilkstein, 2011). In their 2001 study Bailey and Stefaniak had 325 responses to a web survey given to IT professionals in America. The aim of the survey was to collect data on what IT professionals considered important skills for computer programmers. In total 85 skills were identified, with 53 technical skills, 20 soft skills, and 12 business skills. From their survey, the top five skills needed by a computer programmer in the industry are:

1. Code comprehension "The ability to read, understand and modify program written by others."
2. Code specific knowledge "The ability to code programs."
3. To be able to identify errors "Ability to debug programs."
4. The ability to communicate well "Listening skills."
5. Breaking problems down into their smaller components, i.e. "Problem Solving Process."

Source (Bailey and Stefaniak, 2001).

These desired skills are all very heavily practical based and are difficult to assess via exam conditions.

Programming is a difficult concept for a student to learn due to the abstract nature of the subject. It requires a strong understanding of points that are not always concrete. From their literature review, Lahtinen and colleagues state that the majority of coding instructions focus on the surface knowledge needed to learn to code and are context-specific. Students may learn the building blocks of how to code and the semantics and how to write lines of code, however, the difficult part comes through learning how to combine these features into a bigger picture. Therefore, it is important to explicitly teach students these skills (Lahtinen et al., 2005). This is something to consider when writing assessment for programming, the application of these concepts is just as important for a student to master as syntax generation.

Bradley (2020) states that due to the practical nature of programming, to assess programming via exams brings a risk that the assessment is no longer authentic, which can be shown via the practical skills students are required to develop. In their study across two cohorts, with 89 students in cohort one and 113 in cohort two, Bradley trialled a creative assessment structure. To-Do lists through the program Trello (Trello, 2011) were used for students to identify what tasks they needed to complete. The lists had the headings of “To Do, This Lecture, Next Practical, Next Lecture, Done” (Bradley, 2020). The students identified the topics in class and selected what would be covered in the next lectures by in class voting. While done in groups and not individually Bradley encourages students to identify what subjects they feel need more time spent on in class (self-assessment and regulation) and what the students need to work on in the future. This structure allows students, as a collective, to control and direct their learning. The initial findings presented by Bradley show that students received this form of learning well, however, some students still preferred the traditional form of structured lectures.

Similarly, Daly and Waldron (2004) explore the three concepts that exams tend to assess, which they define as "language syntax, software design and problem-solving abilities". Attention is drawn to exams not being suitable for the latter two aspects, where markers often give credit where credit is not due. Daley explores this by suggesting that students who produce weaker responses to exam questions are often credited by the marker as "they can see how a weak attempt could be improved to produce a solution." The alternate option is the use of programming assignments in the form of coursework. Traditionally, however, coursework tasks for a programming module struggle with the issue of plagiarism, both students committing plagiarism and the detection of plagiarism. Daly explored studies which showed that 40% of students admitted that they plagiarised when completing programming assignments. Further, in their study, Daley suggests that coursework is not suitable for assessing program syntax as competent students can use the IDE to identify syntax mistakes (Daley and Waldron 2004).

Skills often explored within programming assessments are ones of problem-solving, the application of programming approaches and the ability to resolve programmer errors. To do this, a programmer needs to be aware of their own abilities (Rum et al., 2016). Students need to be able to identify their weaker areas to be able to effectively analyse and evaluate the work they do and the programs that they write. In this, we can see the application of Bloom's taxonomy, where students are encouraged to work at the higher levels to produce similar results. However, this also requires students to push through difficult problems to understand their own limitations and how to work around them (Rum et al., 2016).

The ability of a student to do this, often separates the weaker submission from the stronger submissions. Robins and colleagues explore Dweck's theory of untapped potential. Robins, however, refers to these students as "movers" and "stoppers". Movers, when faced with a problem they find difficult will look to improve, they will use trial and error along with feedback to investigate further steps to be able to progress to the next stage. Stoppers, on the other hand, will reach a problem and give up - hence we see the fixed IQ students (Robins et al., 2003).

Within programming assignments, novice students often focus on the 'correctness' of the solution, they place the focus on the output rather than the processes. This is a part of a student's limited understanding of how-to code, where code comprehension is secondary to that of the application of syntax. To help students explore a program's full function, Whalley (2011) developed and gave students unit tests which tested a program's secondary functions. It was therefore necessary for students to pass these tests to pass the assignment. Whalley reported that while there was some reluctance from the students, due to the struggle with code function, there was a high level of engagement. There were, however, still some cases of students being mark and correctness driven e.g. as one student stated they should have received full marks as all test cases returned successfully. Whalley found that these test cases could also help with the increasing number of students (and therefore a higher ratio of staff-to-student) by supplying extra clarity before staff are called and supply instant feedback on the work. They do, however, emphasise that these unit tests should not replace all teacher-student interactions. While they can give the student feedback on correctness, it cannot give feedback on code quality (Whalley et al., 2011). Lajis (2018), on the other hand, argues that a student cannot learn the comprehension skills needed to learn to code until they have learned the foundations upon which it is built. Lajis identifies that the cognitive load needed to learn the syntax and the cognitive load needed to problem solve, design and analyse a solution are different.

For novices the concept of diagnosing bugs is difficult, the act of correcting the bugs even more so. When there is little impact to the structure of the code it is easier and doable for the student to correct issues, however it is clear that code comprehension once again is an issue. To further this, when investigating where the most common bugs were, Robins and colleagues found that FOR loops and structures gave students the most issues, as opposed to syntax and initialisations (Robins et al., 2003). When considering the purpose of learning to code, it is important to look at why the code is being created. It is rare that a program is not developed to solve a problem; therefore, it is important to have skills that can help with problem-solving, error handling and the ability to apply mental models to different programs. Therefore, good practice in assessing programming for computer science students requires assessment that goes beyond the use of basic syntax and tests the deeper aspects of being able to code, such as program designing, problem-solving and class usage. How to do this is challenging, Section 2.5.2 explores what is considered to be Good practice when assessing these skills.

### *2.5.2 Good practice in assessing programming*

When considering programming tasks, it is often the case that goals set by teachers are not always met, as teachers often overestimate the skills of the students. Assessors often favour the generation of code, however, the understanding of programming does not always require code generation. Section 2.5.1 explored that some of the more difficult concepts to assess are code comprehension as well as the understanding and application of code. Multiple choice questions (MCQs) can be used to assess these skills as opposed to a CR question (Ala-Mutka, 2005). It is important to note that this understanding of programming concepts does not always translate well to program generation. While CR assessment removes the ability for students to guess at code, MCQs can give the students a starting point. This could lead to the correct application of knowledge to the specific example, which may be sufficient for novice assessment. Considering also the process of elimination that students can take during an MCQ assessment, students are still showing some understanding by selecting what cannot be correct (Ala-Mutka, 2005). Ala-Mutka further suggests that if the aim is to assess a student's practical skills and code generation, this should be done by practical programming tasks. Expanding upon this Lister and colleagues suggest that a poorly explained piece of code under exam conditions could be the result of poor eloquence, not poor understanding. Thus, using multiple-choice questions removes the possibility of this as well as the human judgement often used when marking exams (Lister et al., 2004). MCQs are a valuable tool in the development of novices. While there is a risk of students guessing answers within a MCQ, the analysis of the answers will develop the self-awareness of a student’s knowledge state. The analysis will first show the student what they do not understand and then further hone this knowledge by eliminating the answers the student knows to be false. It is the development of these self-regulatory skills that are important at the novice stage.

Cukieman and colleagues (2014) explore the idea that teaching a student how to learn alongside teaching them how to code is an important experience. They also explore that providing support on learning strategies alongside the technical details on how to code will be beneficial for all students. They explore the changes in assessment structure when a student moves into the undergraduate experience and how it is important to make students aware of the different learning strategies available. The Academic Enhancement Program (AEP) developed by Cukieman and colleagues, requires that all students, not just struggling students, take part. The AEP allows weaker students to get help without feeling as if they are being singled out. The AEP delivers workshops to students that are specifically tailored to their computer science modules. It "encourages students to learn and reflect about a topic of their choice related to success and wellness" (Cukieman et al., 2014). It could be argued that this is not the responsibility of the tutor within higher education, that as adults and independent learners, undergraduate students should be able to seek out these skills themselves. However, the authors also identify that students do not often ask for help when they need it (Cukieman et al., 2014).

Both critical thinking and reflective thinking are desirable outcomes of effective assessment, especially when the focus is to enable a student to become a lifelong learner. Reflection is linked to cognitive behavioural, self-evaluating and self-reinforcing skills (Kuiper et al. 2004). Due to this, it is important to note that the product holds less importance than the learning environment and processes themselves (Gibbs and Simpson, 2003). This is a subject of great importance to be aware of in computer science as often the focus is placed on the functionality of the code, and not the processes that are used. Students often consider the assignment successful if the program outputs a correct solution. To work at the upper levels of Bloom's taxonomy, specifically the evaluation and synthesis levels shown in Figure 3 (page 19), students should be aware of how processes such as program design and good programming practice collate together to make a better end product and to make the student a better computer scientist.

Marin (2009) states that due to the structure of novice programming courses, which rely on computational thinking; first describing the functionality of a program and then creating a coding solution to that problem, any feedback is given needs to be timely and personalised for the novices to grow and expand. To help develop novices into experts, human cognition can be linked to learning to code by focusing on the areas that relate to the "automation of skill with practice, working memory, semantic networks and mental models" (Robins et al., 2003). These are all skills which take practice and recognition beyond that of basic syntax recall. The student needs to be aware of their current knowledge state, along with how to progress from this spot.

It is potentially these attributes that lead to the dissatisfaction from students in assessment and feedback. These skills are difficult to assess and even more difficult for a student to master. They also require a strong grasp of assessment literacy skills to be able to successfully see the links and purpose. It is, therefore, the role of feedback to apply good practices to help guide students in the development of these skills. Feedback must help students understand where the skills are now, how these skills can be developed and how they can get there. To help achieve this, Section 2.7 explores a developed feedback criteria for guidance on what good practice is and how this can be done in a Ipsative form.

### *2.6 Ipsative Assessment*

Ipsative assessment may be a potential way to help improve a students' understanding and skills within a subject and to encourage deep learning of concepts. To answer RQ3 and understand how Ipsative assessment compares to current practice, it is important to understand what Ipsative assessment is. Ipsative assessment involves the monitoring of a student's personal progress throughout a course or a module. It then suggests appropriate tasks and methods that will help them improve their individual learning outcomes based upon an estimation of their current strengths and weaknesses to achieve their personal best (Hughes, 2014). This personal best effect of Ipsative assessment can be seen in sports and gaming. In both areas, the concept is the same, the aim is to beat your personal best on the next iteration. While training for sports, the aim is to keep improving, to beat the earlier score, and then compete with others to win a prize. Similarly, with modern-day gaming, the process is the same, offline gaming allows the user to practice their skills to be better than they were previously then compete with other gamers in an online setting. While education has less emphasis on competition, the fact remains that it is, in a sense, a competition to show your skills. The aim of a degree is for students to improve continuously (facilitation of learning) and to assist a student in identifying themselves as a standout candidate in their field, evidenced by the qualification received at the end of the degree (providing certification) (QAA, 2018; Petrovic et al., 2019).

The traditional structure of current grading systems gives students a perspective of where their skill levels are. Often the student then compares their grades with others leading to a hierarchy of ability. For students who do not perform as well as others, this can be demotivating and can compromise self-esteem (Hughes, 2017a). Hughes discusses that this can lead to students taking a strategic approach to learning and to learn what they perceive is required to pass and gain high marks rather than take a deep learning approach. Ipsative assessment and feedback, however, aims to take a student's assessment and compare it to their previous work as opposed to focusing on external criteria, focusing instead on the *learning gain*. The Higher Education Funding Council for England (HEFCE) defines learning gain as "the distance travelled by students during their studies" (HEFCE, 2016) (Definition 4, Page xi) this focus on the learning gain removes the emphasis on the outputs of the student (Hughes, 2017, pp4). This is important as this could be the difference of a student with experience in computer science designing and editing a program, to a novice student without experience, completing a "hello world" program. It is unfair to compare the output of the two students with vastly different backgrounds in a single session. It is also, however, a limitation of Ipsative assessment, as using practices that emphasise learning gain can be difficult to implement within the strict university regulations, as giving a quantifiable grade to this is difficult. While it is unfair to expect the student without experience to match the output of the student with experience, it is also unfair to expect the student with experience to have the same amount of learning gain as the entry point novice (Hughes, 2017b, pp30). An Ipsative grading system within a traditional assessment form is difficult and unlikely to be used due to the rigid standards that are required when providing qualifications. When structuring Ipsative assessment Hughes describes that similarly to traditional forms assessment, there is often an initial test needed to give a starting point for student growth. Focusing on student growth can also have a positive impact for students as often personal goals set are achievable where reaching the high points of external criteria may not be (Hughes, 2017, pp3). Ipsative assessment does however already happen e.g. in doctoral studies. Students spend time working with a supervisor, getting feedback on work, and finally, a pass/fail result at the end of the process (Hughes, 2017b, pp35). While the assessor is required to give feedback and guide the student, the student also needs to be able to: view their past work, understand where they currently are, where they want to be and the steps needed to get there (Ott et al., 2016). Although Ipsative assessment does not focus on criteria driven assessment schemes, it can be used to help a student complete an assessment by developing their assessment literacy, and therefore their understanding of how to complete assessments. Additionally, Ipsative assessment and traditional forms of assessment are not mutually exclusive. Ipsative assessment does not aim to remove the concepts of grades or a learning structure. It is used as a term to encompass tools and techniques used to enhance student learning, which is why Ipsative assessment is focused on assessment *for* learning as opposed to assessment *of* learning (Hughes, 2017). When assessment is used *for* learning it benefits the students and guides them through their studies, helping with the development of their knowledge. When assessment is *of* learning it has less benefit to the student, its main purpose is to provide certification and has little developmental use for students.

Ipsative feedback can be most seen informally within the classroom setting. Within computer science, Ipsative assessment can be seen in practical labs, where teachers and teaching assistants comment on various iterations of students' work. Within these sessions' teachers give guidance and suggest changes each time the work is viewed. However, due to the informality and verbal aspects of this, this is rarely noted and acted upon by students. Portfolios are a further instance of Ipsative assessment and feedback. Throughout a student's time studying, they can build up evidence of learning gain through instances of their work. However, the marks a student is given usually refer to the quality of the final assessment and do not take into consideration the development of the student (Hughes, 2017b, pg 37). This can be demotivating for the students who do not have experience before the course begins. A student who started with less knowledge could have learned more and still have a lower grade than a student who started the course with experience, this can lead to student dissatisfaction. The lack of acknowledgment of learning gain may provide a partial answer to RQ2, what are the current perceived assessment and feedback issues experienced by students? Table 1 (page 11) shows that on the NSS statement regarding the fairness of assessment within an institution, there is a range of results from 70% to 77%. The lack of acknowledgement of the learning gain a student has made may be a potential reason as to why some students do not consider assessment to be fair.

An important aspect of Ipsative assessment is assessment literacy. Assessment literacy is often a topic discussed from the assessor's perspective. When discussing assessment literacy within assessors, Stiggins identifies that those who are assessment literate can understand the difference between "high and low-quality assessment and are able to apply that knowledge to various measures of student achievement" (Stiggins, 1991). Stiggins identifies two factors when considering assessment literacy: does the assessment clarify the valued achievement outcomes, and does it help identify where there is an issue in a student's learning? Ipsative assessment and feedback help to convey these two components as achievement outcomes are identified on a personal basis to emphasise what is important. Additionally, as Ipsative assessment is personalised to students it becomes clear when each student is not meeting the outcomes outlined. Smith (2013) also identifies three components of assessment literacy: an understanding of the purpose of assessment, an understanding of the procedures and policies surrounding their assessment, and the ability to make a judgement call on what good quality assessment is.

Therefore from the perspective of the student, good assessment literacy is to be able to identify what the purpose of assessment is, how it can be useful to develop their learning and how it is linked with other skills they have learned. In Smiths quasi-experimental study, 369 undergraduate students were involved in a trial to investigate the impact of improved assessment literacy on student grades. They found that while small, there was a positive correlation between the group trained in assessment literacy and their grade on the report. There are two factors which can influence this as Smith addresses. The first factor is the assessment motivation the student has before the intervention takes place. If the student is not motivated to complete and do well on the assessment this will impact the grade a student receives. The second is the ability of that student, the motivation of a student is impacted by that of their ability to complete the work. Regarding assessment literacy Smith states "there is good educational return on the pedagogical investment made." It is interesting that improvement in assessment literacy does show some correlation with grade improvement, (Smith et al., 2013).

An issue with Ipsative assessment, however, is the increase in student numbers in modern-day higher education (Universities UK, 2017). Due to this, the personal one-on-one guidance needed from a teacher for Ipsative assessment to work is a heavy resource process, and difficult to implement. To help counter this, an important aspect of the success of an Ipsative process is that of self-assessment, self-regulation and a student's assessment literacy skills.

Self-assessment is a student's appraisal of their own state of learning, and this is done through the use of reflective questions to try and understand what they know. Self-assessment allows students to "use and make decisions about their own work against a given criteria" (Tai and Adachi, 2019 pp64). This method does require students to have a sufficient amount of knowledge about the assessed topic, along with an understanding of what good and poor assessment looks like. Building on these skills is a student's ability to self-regulate. Self-regulation is a skill that enables a student to manage their own learning. Both self-assessment and self-regulation require students to be motivated and engaged with what they are learning. When a student has mastered these two skills, the student can learn to identify what good performance is based upon the criteria given. A student can achieve this by combining separate bits knowledge they already have to form an understanding of what is expected, more importantly however a student who self regulates can identify how they can get to that level of performance. Ipsative assessment and feedback combines these two practices to enable a student to identify from past work, where they need to improve but also how they can improve in the future. Students who successfully do this have increased academic performance compared to students who do not (Housand & Ries, 2008; Eugenia, 2018; Tai and Adachi, 2019). Using self-reflection and self-assessment develops the skills of students to be self-sustainable without the need for external sources to direct future learning, fulfilling one purpose of higher education, the facilitation of learning (Ibabe, 2009). To answer RQ3 (How does ipsative assessment compare to current practice and can it be used to ameliorate issues identified by students?) it is necessary to understand how Ipsative assessment compares to current assessment and feedback practices within higher education.

When comparing Ipsative assessment (definition 1, Page xi) and summative assessment (definition 3, Page xi) the main differences between the two forms of assessment are within the success criteria. Where summative assessment focuses on the success criteria of a student at any given time, Ipsative assessment considers how the student has improved throughout the time on their course. Therefore, the focus with Ipsative assessment shifts from the end grade to the learning gain that a student has during the course. This removes the emphasis on the final high stakes grade for the student and may encourage a deep learning approach to show mastery of the material.

When considering the levels of Blooms taxonomy Ipsative assessment can help guide and progress a student to the next level by teaching students how to identify where their areas of knowledge fall short, and what they need to learn to improve (Khairuddin and Hashim, 2008). This ability to progress through the levels of the taxonomy require that the student is able to self-assess, self-reflect and self-regulate. As a large component of the Ipsative approach is a students ability to self-assess and self-regulate an Ipsative approach can help a student to work at the upper two levels of Bloom's taxonomy and at the self-knowledge of Nilson's knowledge levels as shown in Figure 3 (page 19) and Figure 4 (page 22) (Bloom 1956, Nilson, 2013).

A concept explored in Section 2.4 was the mindset of students within higher education related to assessment. As a concept, Ipsative assessment rejects the fixed IQ idea and fully embraces the theory of a growth mindset (as defined by Dweck and Legget) within students (Dweck et al, 1998). Summative feedback often informs a student what criteria they did not meet on the assignment and therefore how they did not achieve a mark, a student who has a growth mindset will be able to identify that these are areas for improvement. From this a student should be able to break each task down into smaller components to not feel overwhelmed with their work and that the tasks are achievable. Ipsative assessment can help with this by breaking down the needs of the assessment into individual parts. The breaking down of specification objectives is unique to each student, where one student may need to create smaller sub-goals, another student may need fewer larger sub-goals. This breaking tasks into sub-goals will help students to see where skills like problem-solving and program design can be used in other assessments.

An issue identified in Section 2.3 was that of modularisation causing the links between modules to become unclear. Ipsative assessment may help reduce the impact of this as it requires students to be able to make connections between their previous work and the work they are currently undertaking, which will not be constrained via module boundaries. For example, a student's development and identification of their problem-solving skills will be beneficial to them in all assignments.

Explored in Section 2.5, assessment of programming is an important aspect of this research, it is therefore necessary to consider how Ipsative assessment could help with the assessment of programming. As programming if often knowledge driven with abstract topics, Ipsative assessment may help develop the skills needed by focusing on building the student knowledge over a time period as opposed to having the student take a surface learning approach for one assessment. This can be within the practical sessions that students attend through the use of formative exercises which build into a bigger final assessment. Through practice students can also develop the harder skills which require a programmer to be aware of their own abilities and limitations (Rum et al, 2016) such as error detection. By placing the emphasis on the learning throughout the development of a solution this may direct students away from focusing on the 'correctness' of the solution and instead to focus on the process of creating a program.
Bradley explores the implementation of Ipsative assessment using agile assessment. In their study across two cohorts, with 89 students in cohort one and 113 in cohort two, Bradley trialled a creative assessment structure. To-Do lists through the program Trello (Trello, 2011) were used for students to identify what tasks they needed to complete. The lists had the headings of "To Do, This Lecture, Next Practical, Next Lecture, Done" (Bradley, 2029). The students

identified the topics in class and selected what would be covered in the next lectures by in class voting, this shows a process of Ipsative assessment being used within the classroom. While done in groups and not individually Bradley encourages students to identify what subjects they feel need more time spent on in class (self-assessment and regulation) and what the students need to work on in the future. This structure allows students, as a collective, to control and direct their learning which is a large factor of Ipsative work. The initial findings presented by Bradley show that students received this form of learning well, however, some students still preferred the traditional form of structured lectures. This shows promise that for the understanding of programming, Ipsative forms of learning may help towards reducing student dissatisfaction by allowing control and direction to be decided by students

Section 2.6 identified Ipsative assessment and feedback as a process and how it compares to traditional assessment practices. This work will explore through the use of case studies if Ipsative Assessment and Feedback is a viable way to reduce the impact of the issues identified throughout the literature review to improve student satisfaction.

### *2.7 Feedback Criteria*

Chapter two explores good practice of assessment and feedback along with their associated problems. Based on the literature explored, the following feedback criteria was developed using a set of characteristics of an Ipsative form. This feedback criteria gives guidance on how feedback benefits students and can help reduce the issues faced by students.

**1 Must give ways to improve the work, with recommendable transferable skills.**

An identified issue with assessment and feedback is the modularisation of higher education courses. While an important aspect of giving feedback is to inform the students how they can improve their submitted work, it is also important to inform the student how these skills can be applied in other assignments and modules. This is achieved by also giving feedback on generalised transferable skills. For example, if a student struggles with control structures, feedback should guide students on how to develop these skills and how they are applied to other problems, not just how to improve the control structure used in that instance. This will help students see the connections to different modules and begin to synthesis new theories into their current knowledge (Ott et al. 2016; Nicol 2010; Struyven et al., 2005; Hughes 2011; Ferrell, 2010; Bloom, 1956).

**2 Does not comment on the students themselves but instead comments purely on the work.**

To encourage the facilitation of learning, students must be guided towards a growth mindset. They must be encouraged to see that it is possible to learn the material they set their mind to.

Commenting on the student or the student's abilities enforce the concept of a fixed mindset. For 'at risk' students, this can enforce the idea that certain aspects of the material are beyond their reach. For a well-performing student commenting on their abilities as opposed to the work that they have submitted, could potentially turn confidence into arrogance. By commenting on the work itself, students can identify what aspects of their work need to be improved. This will improve a student's ability to self-assess and self-regulate when working on future assignments (Nicol and Macfarlane-Dick, 2006; Gibbs and Simpson 2003, Black, 1998; Tai and Adachi, 2019, pp64).

***2a The comments should be non-judgemental***

While it is important to not comment on a student's abilities, it is also important that feedback given to students is non-judgmental. The difficulty of a concept will vary depending on the abilities and strengths of a student. It is important that a student is able to identify how they are doing, what is the next target and how do they get there. It is, therefore, crucial that a student feels that their mistakes are acceptable and are allowed to grow (Hughes, 2011; Ott et al., 2016).

***3 Refers to the work itself with given examples.***

While feedback should not comment on the student personally, it should still be personal. General statements that are given to all students provide little use. Feedback that is supplied with context, however, gives room for improvement without turning a student’s confidence to overconfidence. Feedback needs to inform the student where they went wrong within the work, therefore referring to the work itself and giving examples from the text when giving feedback adds the personal element. By referring to work that the student has done, and using feedback to encourage reflection on the work, this encourages an Ipsative practice to help a student develop on their weaknesses (Nicol, 2010; Black and William, 1998).

***4 Must clarify what good performance is and how the student can reach it.***

While a large component of feedback is a reflection on the work completed, it is also important that feedback *feeds forward*. To help with the facilitation of learning, feedback needs to show students how they could have reached the criteria that they did not meet within the specific assignment. To be able to monitor and evaluate their own learning students need to be able to develop mental models regarding what good work is. This can be achieved by providing direction to students within feedback, showing the student not only what a strong solution should be, but the steps needed to get there. When used with feedback criteria 3, it allows students to identify where they are in their learning, where they need to go, and how they can get there (Struyven et al, 2005; Price et al, 2010; Ott et al, 2016; Alderman et al 2005; Laurillard, 2002).

***5 Must recommend a manageable workload.***

It is important when giving feedback to be realistic as to what a student can or cannot do. For example, the focus point of the feedback given to a student would vary depending on the work submitted. An 'at risk' student would not benefit from feedback about concepts beyond their reach. The focus for an 'at risk' student should be gaining an understanding of the basic concepts of that assignment. In comparison, a student who has mastered the basic material could be given further guidance on more advanced topics. If a student feels that they have too much work, this can cause disengagement. To ensure that students stay engaged and give a personal Ipsative aspect, feedback should assist students in regulating their own work, giving suggestions about what the assessor feels is manageable (Struyven et al., 2005; Hughes 2011).

**6 Feedback given to students must be clear.**

A further cause of dissatisfaction and disengagement in students, is a lack of clarity in feedback. This unclarity can take many forms, vagueness, no clear suggestions for future improvement, a misunderstanding of what consists of feedback, and the meaning of the feedback. To minimise a lack of confusion feedback should be clear, in that spelling, grammar and wording should be of a good standard (Price et al., 2010).

**7 When the student gets something wrong, the feedback informs the student why this was wrong.**

An integral part of understanding how to improve is understanding why something was wrong in the first instance. Informing a student that something is wrong is less useful than why it is wrong. Without this understanding of how the attempt was wrong, the student may make the same mistakes in the future in different contexts. Feedback should identify where mistakes have been made. However, since feedback should be *for* learning as much as *of* learning, it also needs to explain *why* mistakes are made. Furthermore, an important aspect of Ipsative assessment is self-assessment and self-regulation. Knowing why something was wrong alongside knowing what good practice is, helps students to build up the appropriate mental models to sufficiently self-assess (Black and William, 1998; Duncan 2007; Ferrell, 2012).

### *2.8 Summary*

This literature review has examined assessment and feedback within higher education and the role it plays in the student experience. Summative and formative assessment play key roles in the two functions of higher education, to provide certification and to facilitate learning. However, while there is a good understanding of the learning theories which constitute good practice within the literature, there are still problems surrounding assessment and feedback which makes best-practice hard to maintain in the classroom and can often cause student disengagement. These problems involve: course modularisation, a lack of assessment literacy among students, learned dependence on grades, a lack of communication between assessor and student as well as the rapidly increasing class sizes within higher education making personalised feedback difficult.

Ipsative assessment which focuses not on the externally referenced criteria but on the student's learning gain from the previous assessment may help ameliorate these issues. Ipsative assessment can develop practical assessment literacy skills within students. Examining past work to identify where there has been learning gain, develops a growth mindset within students. Giving evidence to students to the fact that there can be progression which can include the development of small tasks. This can also develop a student's ability to self-assess, the practice of referring back to previous assessment and identifying areas where they have improved, will enable the student to identify similar areas of improvement in subsequent drafts. From the review of the literature, a set of feedback criteria was developed based on best feedback practice, to be used in the analysis of historical data. The following chapter explores the methods used to collect and analyse the data used within this study.

# Chapter 3. Methods

### *3.1 Introduction*

This chapter outlines the methods used to explore both the qualitative and quantitative data collected during this research. It describes how each data set was collected and how they were used and analysed to help answer the research questions defined in chapter one. This chapter also outlines the statistical tests chosen to analyse the qualitative data collected and why these were the tests used. Finally, the limitations of the data and methods are outlined.

### *3.2 Statistical Tool Kit*

The types of statistical tests to be used depends on the types of data that have been collected, there are three types of data used within this research. The first type of data is Nominal data, nominal data has no rank, there is no value that states one variable is however many times better than the other (examples of Nominal data are sex or age, yes or no). The second type of data is Ordinal data, which has a rank, such as strongly agree or strongly disagree. While these values have a rank, i.e. strongly agree is better than agree, there is not a quantifiable numerical difference (strongly agree is not twice as good as agree). The third type of data used is Ratio scale data, Ratio scale data gives rank and order to the data and has a true zero, therefore is the preferred data type to use when possible (Cohen et al, 2018).

The type of test that can be used on collected data can be split into two broad topics, descriptive tests and inferential test. Descriptive statistical tests are used to describe the data only, these tests cover stats such as the mean and median. No inferences can be made from these results, they are purely informative (Cohen et al, 2018). The second type of statical tests are Inferential statistics. Inferential statistical tests measure the differences between groups and allow the research to make inferences on the data. When choosing Inferential statistical tests, it is important to decide whether the data is suitable for parametric or non-parametric tests. To inform the decision on which statistical tests can be used, normality is the determining factor (Cohen et al, 2018).

Kurtosis and Skewness inform the researcher of the normality of the data distribution. A data set with a perfect distribution would have skewness and kurtosis of 0. The Kurtosis and Skewness of a data set can influence what statistical tests can be used and what inferences can be made. It is suggested that data sets have a skewness or kurtosis value of -1<> +1 for normal distribution, if the skewness is too wide it is unwise to use parametric tests. Outliers to the data can be found via the use of box plots or histograms and the way to handle these outliers is the choice of the researcher. If the distributions are too far out of a normal distribution it is suggested that the researcher does not use parametric tests but instead uses non-parametric (Cohen et al, 2018).

Statistical significance is an important aspect to be considered throughout research, this measurement tells the researcher if there is a difference to be reported in the results given, what is does not do is explore the magnitude of that difference. Due to this, statistical significance on its own is not enough of a result to prove, or disprove, a hypothesis. Cohen explores this by pointing out the aspect that journals have been known to reject research papers that have only supplied statistical significance as evidence of a successful experiment. To measure this difference a measurement termed effect size or 'ETA-Squared' is used. Effect size measures the magnitude of difference between the two variables. This may tell the researcher that while there is a significant difference (the researcher needs to be aware that there has been a change) this may not always be a large enough change to be a factor (Cohen et al, 2018; Richardson, 2011). There are two primary factors in effect size, 'magnitude' the value of the effect size and 'association' the direction of the effect. While small samples may not be able to detect effect sizes this will not be a factor within this research due to the large sample sizes obtained (Cohen et al, 2018).

There are two types of error which can occur when running statistical tests, the first is a 'type-one' error which is a false positive that rejects the Null Hypothesis when in fact the null hypothesis is true. The second error type is a 'type two error' which is a false negative, accepting the null hypothesis when it is false (Cohen et al, 2018). The main concern within this research is receiving a false positive. While there can be indications that aspects of the case studies have positively affected the opinions of the cohort in terms of the current research, there is a large range of factors that cannot be accounted for (discussed in Section 3.6).

### *3.3 Parametric Tests*

Section 3.2 explored how traditionally parametric tests should be used when data is normally distributed and has limited skewness. This research has collected data which typically does not have a normal distribution, such as Likert data (explored in Section 3.4) however, Norman (2010) explores how modern-day statistical tests such as ANOVA, are robust against highly skewed data and non-normally distributed data and are independent of the type of scale. They further explore the literature of Pearson (1931), Dunlap (1931) and Havlicek and Peterson (1976) who confirm that this also holds true with correlation and regression (Norman, 2010). Kaptein also identifies that in CHI (CHI2020, 2018), a predominant human-computer interaction conference, focusing on user experiences, 80% of accepted papers using Likert style data use parametric tests, therefore supporting the use of parametric tests on Likert style data as an accepted peer-reviewed system. Kaptein further emphasises that most educational and health science data use these two practices regularly (Kaptein et al, 2010). Therefore, due to this supporting evidence, parametric tests will be used to analyse the results the quantitative data collected in this research.

### *3.3.1 Analysis of variance (ANOVA)*

ANOVA tests have multiple variations depending on the number of variables the researcher wants to compare: one way ANOVA tests one variable, two way ANOVA takes in account two independent variables and finally multiple ANOVA which is used when there are three or more variables the researcher wants to compare An independent variable is a group, such as teachers/ students; which will require the dependant variable to be a continuous variable (Cohen et al, 2018).

One of the tests used throughout this research was a one-way ANOVA between groups. This analysis of variance looks at the difference between two groups to identify if there is a significant difference between the groups analysed (Pallant, 2010). While an ANOVA will tell the researcher if there is a statistical significance, it will not tell the user where the significance is. To do this the researcher needs to use post hoc tests. These tests depend on the equality of variance. If the variance is equal, the Tukey test is used. The Tukey test takes the results which are not statistically significant and places them with other groups to test for a significant response, this then shows exactly where the similarities and differences are (Cohen 2018 pg783). This will compare each cohort year with the others in the data set and identify which of the cohort years have a statistical difference between them, once the differences have been identified, the effect size will then be calculated. Additionally, the 'Eta-squared' value will help identify the magnitude of the of the difference in addition to the statistical significance.

### *3.3.2 Correlation*

Correlation investigates an association between two variables, a correlation cannot tell you what the cause of the association is, but it can be used to tell the researcher if there is one present. If a relationship exists, Correlation tests can identify if this relationship is asymmetric or symmetric. An asymmetric measure is a one-way relationship between the two variables, so variable 'a' implies variable 'b' but not the reverse. A symmetric measure occurs when 'a' implies 'b' and the reverse is true. Cohen states that when there is a perfect association between the two variables, there is a percentage difference of 100%. Once a relationship has been established the researcher needs to investigate the direction of the relationship, negative or positive, and the magnitude of the relationships. Pearson's product is an established measure of association and ranges from -1 to +1, this coefficient is labelled as r. This is a measure of a linear correlation between two factors (Cohen 2018). If the variables both increase and decrease at the same time this is considered a positive relationship. If one decreases and the other increases, this is known as a negative relationship. It is very rare, within a social research setting, (in this case, education research) to see a perfect correlation. It is, however, important to note that a correlation does not always imply a relationship (Cohen, 2018).

### *3.3.3 Regression Analysis*

Similarly to correlation, regression analysis models the relationship between two variables and allows the researcher to predict the value of a second variable. Regression, however, is different to correlation in the fact that rather than predicting a trend it allows an exact prediction of the values, within a line of best fit. A condition of this test working is a limited number of data outliers, with the best-case scenario being no outliers and that the data points all cluster around the line of best fit. SPSS gives us the variable of $r^2$ which informs the researcher how much the variances of the dependent variable (the number of outliers) affects the results. With multiple regression analysis, the researcher can investigate the impact of two or more independent variables and the effect they have on the dependent variables. (Cohen 2018).

### *3.3.4 Cohen's Kappa*

Cohen's kappa is a measure of inter rater reliability when there are two raters measuring a variable on a categorical scale. It is important to note that when using Cohen's kappa, that just because two raters are in agreement, it does not mean that those raters are correct. An example used by Laerd is just because two doctors refer a patient to see a specialist, it does not mean that this patient does need to be referred. The correctness of the 'judges' is either a judgment the researcher has to make, or a further research point (Laerd, 2018).

## *3.4 Data collection methods*

### *3.4.1 Interviews and focus groups*

To help answer RQ1 "What is good practice in assessment and feedback in undergraduate computer science in the UK and how does current practice at Newcastle University compare?", focus groups and interviews are conducted. Focus groups and interviews are used to collect data via an open dialogue between the researcher and the participants. They allow the participant to be viewed not as data but as a person, with views and perceptions. A benefit to using interviews is that interviews allow for questions to be explored in further depth than a survey or questionnaire. A limitation to this form of data collection is a bias from the interviewer. The interviewer can have a bias in how the questions are asked, or the questions can be worded in a biased way towards the interviewer's goals (Cohen et al., 2018, Pg 506). A focus group is a form of an interview that includes multiple participants. The aim with a focus group is to encourage dialogue between the participants to allow their agenda and not the agenda of the researcher to become prominent (Cohen et al., 2018, Pg 532).

Due to this ability to gather information in greater depth than questionnaires, interviews and focus groups was used to partially answer RQ1 (what is current practice at Newcastle university) and answer RQ2 (what are the current perceived assessment and feedback issues experienced by students). As this was in a small case study at Newcastle University, the results cannot be extrapolated to a nationwide answer. They do however provide some insight into whether the issues identified by students match the issues discussed in Section 2.3.

During Semester two of the 2016/2017 academic year, a focus group was held for undergraduate students in Computer Science at Newcastle University. Initial recruitment was through requests in lectures followed by an email to all students in the cohort, and finally by asking staff members if they could suggest any students. Despite this, student turnout was low, with only four students volunteering in the focus group. The four students were in first year, all male and were registered students on the CSC1021'Programming one' module at Newcastle University. The focus group ran for 30 minutes, the transcript can be seen in Appendix B. Throughout the focus group, students were given a number to identify themselves to ensure anonymity.

Following the low student turn out to the focus groups, interviews were held to attempt to gain more participants. There was a slightly higher response rate for interviews with six students volunteering. Three of these students were first-year students, all male, and all part of the CSC1021 module (for full module information, see Appendix A). An additional three students from the final year of the bachelor's degree volunteered. There were two male students and one female student from the year group. All student participants had completed the module CSC1021 previously. The questions asked in the interviews were the similar to the focus group questions with some additions.

The focus group questions were as follows:

1. What do you think the purpose of assessment is?
2. What do you think the purpose of feedback is?
3. What types of feedback have you had?
4. What is your preferred method of getting feedback?

Do you understand the purpose of each assessment that you get?

After completing the focus group, it was identified that there were some gaps within the questions, therefore within the student interviews additional questions were asked. The interview questions were as follows:

1. What do you think assessment is for?
2. What do you think feedback is for?
3. How do you use feedback?
4. What feedback types work for you? What feedback types do you prefer?
5. What do you think the problems are with the current assessment and feedback practices are? And what are the good things?
6. What has influenced the way you view and use feedback?
7. Is it always clear why you are doing a specific assessment?

To provide an answer to RQ1 and RQ2, question 1, 2, 3, 6 and 7 of the focus group investigate students' views on the purpose of assessment and feedback, how feedback should be used and what is currently done within higher education. As identified in Chapter two when students do not understand the purpose or use of feedback this can cause issues with their longitudinal development and cause students to disengage (Struyven et al 2008, Ott et al, 2016). Understanding what it is that students think feedback is for, along with how they use feedback, is an important component. An important aspect to understanding the issues that students have regarding assessment and feedback is to understand what students consider to be assessment and feedback. Therefore questions 3 and 4, were asked to identify what students thought feedback was and how that feedback was then used. The focus groups and interviews are transcribed, coded and discussed in chapter 4.

Following this, staff were interviewed with the same themes as the students, outlined below. These interviews aimed to investigate the opinions of staff concerning assessment and feedback and to try to identify any differences in their mindsets compared to that of students. There were five staff volunteers for the interviews, all staff interviewed were teaching staff on a computer science undergraduate module. There were a variety of programming languages taught by these lecturers, including C, Java, VDM, SQL, and Bash.

The following questions allow the researcher to gather data on current practice within a Higher Education institute and how this compares with the good practice identified within the literature. These questions were the same as the questions asked to the students within the focus group and interviews. The questions directed towards staff place emphasis on the development of assessment and included an addition of programming assessment specific questions.

The questions on the interview were as follows;

1. As a developer of assessment, what do you think is its purpose?
2. What is the most challenging aspect of assessment creation?
3. What do you think the purpose of feedback is?
4. What is your preferred method of giving feedback?
5. In your experience, do you think students understand the point of every assignment?
6. Do you think students are using feedback in the way you expect?
7. What would you consider to be the best assessment and feedback practice?
8. What do you think are the challenges of assessing programming?
9. What skills are most important to test during a programming assessment?
10. What are the differences between assessing programming assignments and other forms of assignments?

As RQ2 focuses on the perceived issues that students have with assessment and feedback, questions 8, 9 and 10 directed staff to focus on the possible challenges and that students may have with programming assessments specifically. These questions also aim to investigate if creators of assessments are keeping these skills in mind throughout the assessment creation process. Section 2.5 explores how Bailey and Stefaniak identify that the main skills needed by programmers are code comprehension skills and problem-solving skills (Bailey and Stefaniak, 2001). The latter are skills which can be applied in other modules, thereby reducing the impact of modularisation along with the negative impact this can have, and therefore are important to consider when designing assessment (Ferrell, 2012).

### *3.4.2 The assessment experience questionnaire*

As an alternative data collection method to focus groups and interviews the Assessment Experience Questionnaire (AEQ) developed by Gibbs and Simpson (2003) was selected. The questionnaire was a further opportunity to collect student data with less impact on student time and effort. Before giving the questionnaire to the students, it was first edited. The original questionnaire contained questions regarding exam experiences. It was decided that this was not relevant to this project, as at the time of the distribution the students had handed in their final assignments of semester one but had not had their exams. Therefore, the exam questions were removed. On the original unedited questionnaire there were 36 questions total, on removal of the exam questions there were 30 questions remaining.

| Key | Value |
|---|---|
| - | Unanswered/unusable |
| 1 | Strongly Disagree |
| 2 | Disagree |
| 3 | Neutral |
| 4 | Agree |
| 5 | Strongly Agree |

*Table 3 Coding Values of Assessment Experience Questionnaire*

The AEQ was delivered in paper form to students attending the module. The students were informed that this information was to be used in a research project and the answers and data collected would be kept anonymous. The students were asked to withhold any identifying details on the survey to both emulate the anonymity of the NSS results and to try to ensure honesty with the responses. Students were given a chance to ask questions while the forms were filled in and given the contact information of the researcher for follow up requests. The results from these questionnaires were then input into the statistical software SPSS. Table 3 shows how the responses were coded into SPSS.

| Academic year | Number of responses | Total number of students |
|---|---|---|
| 2016/2017 | 161 | 261 |
| 2017/2018 | 147 | 284 |
| 2018/2019 | 225 | 306 |

*Table 4 Student Response Rate and Cohort numbers*

The questionnaires were given to students during the academic years; 2016/2017, 2017/2018 and 2018/2019. The response rates as well as the total number of students in the cohort can be seen in table 4.

The analysis of the AEQ uses both descriptive and inferential statistics to analyse the responses. Descriptive statistics such as calculating the mean of the responses to the AEQ will give a general understanding of the perception of the cohort as a whole, after this initial understanding Inferential statistics tests are used to investigate the data further.

As discussed in Section 3.3. ANOVA tests are used to identify if there is a significant change in the responses of two groups. Therefore, the ANOVA test was chosen to identify if there was a significant change in student perceptions of assessment and feedback based on year group. While using Anova tests to analyse the results of the AEQ, the independent variable will be the cohort year, to separate the students into groups and to allow for a comparison to be made. The continuous variable will be the means of the responses to each question, allowing the researcher to track any shifts in the opinions of each cohort. If there is a significant difference between the years, ETA-squared and tukey tests will be used to identify which year groups have the significant difference but additionally what the magnitude of the difference is. Tuner and Gibbs (2010) also used ANOVA tests to identify differences in responses between distinct groups. The groups in Turner and Gibbs 2010 paper however use Male and Female as their distinct groups as opposed to Cohort year. Additionally they also categorise the Likert scales with following numerical values 1 = 'strongly disagree', 2=' disagree' 3= 'neutral' 4 = 'agree' 5=' strongly agree'.

In addition to ANOVA tests correlation test are used within the analysis of the AEQ to investigate any potential links between questions to help further identify student perceptions and behaviours regarding assessment and feedback. For example, if students find the material to be unacceptable do, they disengage with the feedback?

### *3.4.3 Feedback Criteria*

To help answer RQ1, the feedback criteria (as developed in Section 2.7) was used to investigate the feedback given to students on the module CSC1021 during the academic years 2012-2016. This had two purposes, the first was to see if the problems that students identified from the interviews and AEQ were present within the feedback. Second it was to investigate how close to good practice this feedback was.

The feedback criteria were used to analyse the quality of the feedback given to students taking the module CSC1021 Programming 1. In total, 951 bits of feedback given to student and their final module grades were used to complete this analysis. CSC1021 aims to teach the students the basic understanding of object-oriented programming using Java as the example language. It covers the basic topics that a student is required to know to complete programming tasks.

One of the assessment methods of this module is to complete ten practical lab sessions (an example can be seen in Appendix C) which are signed off by a postgraduate demonstrator within practical sessions. These assignments are pass/fail. There are also two projects that a student must complete, each project is submitted to the Newcastle Electronic Support System (NESS) (Newcastle University, 2020). As the feedback for project one during 2012-2014 was via the use of feedback sheets physically given to students, this data is not available (see Appendix D for feedback sheets). Due to this, only project two was analysed to ensure consistency across the six years of feedback analysis. The aim of project two is to develop a program in Java, which uses classes, inheritance, control structures, collections and good programming practice. The assignment specification can be found in Appendix E. In addition to this, the final module grades of the students were also used to try to investigate any potential correlation between feedback which follows good practice quality and final grade.

The School of Computing supplied all undergraduate data used within this project, the data received were anonymised. Figure 8 shows an example of the coding of the data within SPSS. Each student was given an id number and eight additional datapoints were assigned to each student. The written feedback was coded in a method similar to interview coding, themes throughout the feedback were identified and put into categories. From this it was then decided that either yes, the feedback met the criteria based on this coding, or no it did not. This was then translated into the statistical software SPSS. Each bit of feedback was given a key and associated with each piece of feedback was 8 data points with the values 0 = "No it did not meet the feedback criteria" or 1 = "Yes is did meet the feedback criteria".

| | FC1 | FC2 | FC2A | FC3 | FC4 | FC5 | FC6 | FC7 | YEAR | GRADE |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 2012 | 80 |
| 2 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 2012 | 21 |
| 3 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 2012 | 27 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2012 | 8 |
| 5 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2012 | 35 |
| 6 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 2012 | 75 |

*Figure 8 Feedback Criteria Coding Example in SPSS*

Correlation tests (see Section 3.2.3 for details) are used to investigate if there is a relationship between the amount of feedback criteria met and the final mark that a student obtained. In addition, Multiple Regression was used on the Feedback Criteria results to investigate if meeting a collection of feedback criteria will predict the grade of the students (see Chapter 5). Finally, to ensure consistency, validity and reliability, Cohen's Kappa, a measure of inter-rater agreement between two raters on a categorical data set, was used (Laerd, 2018).

### *3.4.4 Computing: Ipsative Assessment*

To answer RQ3 "How does Ipsative assessment compare to current practice and can it be used to ameliorate issues identified by students? " the tool "Computing: Ipsative Assessment (C:IA)" was developed. C:IA was used in a trial run with first-year students in the module CSC1021. The system was introduced in a lecture to all stage one undergraduate students and was offered as a voluntary addition to their course.

Five first year students responded and took part in this trial of C:IA. The students that took part in the experiment were asked to answer questions before they used the system (see below) on their programming ability. Once the trial was complete the students were asked to answer some questions on the quality and usefulness of the system. One of the five students did not complete the post system interviews. The students were informed about what the system would do and were asked to use it alongside their preparation for lectures and coursework.

Pre-Questions

1. What is your programming experience?
2. What languages have you used?
3. What is the most challenging aspect of computing?
4. With regards to programming, what areas do you feel are a strength?
5. With regards to programming, which areas/topics do you think needs improvements?

After the initial set up meeting the students were not contacted until the end of semester one. The aim of the system was to encourage self-guided learning with an Ipsative approach, therefore input from the instructors was kept to a minimum. Students reported that they did not use the C:IA as a semester-long self-reflective system. Due to this, students were then encouraged to use the system as a revision tool instead, placing heavy emphasis on the goal tracking and quiz aspects of the system. Once the exams were finished and the students had used the tool, the following post questions were asked:

Post Questions

1. How confident do you feel with programming?
2. Did the system help you learn the principles of programming?
3. Do you think you have improved since taking the first questionnaire?
4. Do you feel like there is anything you could still be working on?

5. What features would you suggest to improve the app?

Four out of the five students who initially volunteered responded to these post system questions. As Ipsative assessment sets out to help students develop the confidence to identify their strengths and weakness (Hughes, 2011, question one was used identify if the students were confident in their abilities to identify these strengths and weaknesses. RQ3 investigates if Ipsative Assessment could be considered a viable option to help ameliorate the causes of dissatisfaction within students. However, for Ipsative Assessment to be a viable option to do this, the processes must be able to help a student learn the principles of programming, therefore question two was used to evaluate how well C:IA helped achieve this. As explored in Chapter two, the ability to identify what work needs to be done in the future, along with how the student has improved since beginning the work are crucial aspects of Ipsative Assessment (Ibabe, 2009; Hughes, 2011). To help explore RQ3, question three and question four investigate if a student has been able to use Ipsative practices successfully. Students who have successfully achieved this will be able to answer with specific goals and topics in mind. Question 5 was asked to understand the experiences that students had with the system, and how it could be improved in the future. Interview coding was used to analyse the responses to these questions. A discussion of results can be seen in Chapter 6.

In addition to interviewing students to obtain their perceptions on how C:IA and Ipsative assessment could reduce the impact of the perceived issues of assessment and feedback, staff were also asked to explore the system. Staff were asked to use the system in a single session and to look at it from the perspective of an assessor. The questions asked to staff were as follows:

1. As a concept, do you think Ipsative assessment would be beneficial to students?
2. Would a system such as C:IA, help a student gain an understanding of the concepts of programming?
3. Does C:IA provide a clear region for self-regulation, reflection and growth? Please expand how/ how not?
4. When exploring the Quiz sections do you find the questions to be suitable for first year students? Are they too hard? Impossible for the students to do? Or manageable but challenging?
5. What do you think the benefits of such a system would be?
6. What do you think the drawbacks of such a system would be?

7. What features do you think should be added to the system?

Table 5 shows how each of these case studies answer the research questions proposed in chapter one. The focus groups, interviews and feedback analysis are used to answer RQ1 by identifying what current practice is at Newcastle University regarding assessment and feedback. In addition to the AEQ, the focus groups and interviews are also used identify the perceived issues of students and therefore answer RQ2. Finally, the development and trial of the 'Computing Ipsative Assessment' system is used to identify whether Ipsative assessment and feedback could help reduce the impact of these identified issues.

| Research Question | Case Study |
|---|---|
| RQ1 | Focus Groups, Interviews, Feedback Analysis |
| RQ2 | Focus Groups, Interviews, AEQ |
| RQ3 | Computing: Ipsative Assessment |

*Table 5 Mapping of research questions to case studies Research Question*

### *3.5 Ethics*

As these case studies required the use of student data, the appropriate ethical approval was required (see Appendix H for ethical approval form). GDPR rules were adhered to through the anonymisation of the data collected.

### *3.6 Limitations*

One limitation to education research studies is the lack of a wider application as experiments often take time and are difficult to replicate. Furthermore, experiments are often useful for testing one component of pedagogical behaviour but often fail when considering additional components and connectivity within other areas. The robustness of an experimental educational study is often brought into question, issues such as: whether the experiments can be replicated, will the results be the same? and what effect has monitoring the students had on their behaviour? (Schanzenbach, 2012) are important to consider. Using peer-reviewed questionnaires, a feedback criteria that is not course-specific and a system which could be adapted to any subject, will ensure that these case studies and experiments could be replicated with other cohorts.

However, it is also important to note the data collected within this research is both qualitative and quantitative. This means that while the data can give information on how feedback can impact a student's experience, this information cannot be applied perfectly to other cohorts as student numbers, experiences and backgrounds will all have impacts.

Historical data analysis is a useful tool to investigate large amounts of data and the changes that can happen over time. However, by its nature, historical data is subjective and limited by the questions asked of it (Cohen et al., 2018, pp325). This research investigates a potential relationship between the quality and consistency of the feedback given to students and the students' final grade. The scope of this research does not include extenuating circumstances for the students, nor does it track feedback given by each marker. This data is still useful however as the assessment and teaching within the module CSC1021 has remained consistent. While there are changes to the scenarios of the coursework, as seen in Appendix F, the learning objectives and goals remain the same and thus provide a form of consistency.

Despite the limitations presented here, this research provides insight into the issues faced by students in the School of Computer Science at Newcastle University. These experiments can be replicated and used with future cohorts in different subjects.

### *3.7 Summary*

This chapter outlined the data collection process for the three Case Studies investigated throughout this research. These case studies involve interviews and focus groups, the Assessment Experience Questionnaire (AEQ), the Feedback Criteria analysis and the protype trial of the Computing Ipsative Assessment tool. The statistical tests and methods described throughout the chapter were used to answer the research questions identified in Chapter one. They identify some of the perceived issues that students have regarding assessment and feedback and whether these issues are present within feedback. These tests also investigate whether Ipsative assessment and feedback could help ameliorate some of the issues identified by students. Finally, the limitations of the data collected, and the methods used are discussed and ways in which these can be managed to ensure consistency and reliability of the results are outlined.

# Chapter 4. Student Perceptions of Assessment and Feedback

## *4.1 Introduction*

Chapter four explores the results of statistical and qualitative tests to answer RQ2 (what are the current perceived assessment and feedback issues experienced by students?) using a novice programming module at Newcastle University as a case study. This investigation uses qualitative methods such as focus groups and interviews targeted at undergraduate students and lecturers within Newcastle University. Additionally a quantitative experiment collected data via the Assessment Experience Questionnaire (AEQ) (Gibbs and Simpson, 2003) was used. The AEQ was analysed using descriptive statistics such as ANOVA and correlations tests, to explore how students perceive assessment and feedback practices within this institution. This chapter then concludes with an analysis of how Ipsative assessment and feedback could ameliorate the discussed issues.

## *4.2 Pre and Post Moderation*

Pre- and Post-Moderation were processes started in the school of Computing at Newcastle University to ensure consistency across the postgraduate demonstrators who were employed to mark assignments. These processes were started to address inconsistencies within feedback given to student caused by the increasing class sizes.

In pre moderation postgraduate demonstrators mark a selected number of scripts, usually around three of their assigned marking totals. Once the postgraduate demonstrators have marked these assigned scripts, a member of staff reviews the marks alongside the feedback given and assesses the quality and consistency across the markers. This period is the instance where mistakes in the marking scheme or misunderstandings of the postgraduate demonstrator are commonly picked up. If the marking is completed to the satisfaction of the member of staff, the demonstrator may mark the rest of their assigned marking. If this is not the case, the postgraduate researcher is given further advice and asked to mark another two or three of their assignments and the process repeats. This process is typical for all modules in the first and second stage of the BSc Computer Science undergraduate programme.

Post-moderation occurs after demonstrators have marked all their assignments and returned them to the module leader. The module leader checks these scripts a final time before "signing them off" and releasing the marks and feedback to the students. Pre- and post-moderation were processes that began before data was collected for this study, the process become more consistent and therefore more developed during the time of this study. This may have had an impact of the perceptions of students due to the increased consistency this may add to the students feedback. The statistical tests further investigate if the improvement of pre- and post-moderation has a relationship with student perceptions on feedback.

To investigate a potential relationship between pre and post moderation and student perceptions of assessment and feedback the following hypothesis were defined.

(H0) Null Hypothesis: Pre and post moderation do not affect student perceptions of feedback.

(H1) Alternate Hypothesis: The controlled structure of pre and post moderation leads to a more positive perception of feedback from students.

### *4.3 Interviews and Focus Groups*

To answer RQ2 and identify the perceived issues that students have, semi-structured interviews and focus groups were conducted with a selection of year one and year three students. In total, four first-year students attended the focus group. A full description of the respondents and their background, along with question justification can be found in Section 3.4.

#### *4.3.1 Focus Group Responses*

Within the focus groups 4 first year students volunteered to take part, the questions asked were as follows:

1. What do you think the purpose of assessment is?
2. What do you think the purpose of feedback is?
3. What types of feedback have you had?
4. What is your preferred method of getting feedback?
5. Do you understand the purpose of each assessment that you get?

Questions 1 and 2 were asked to understand student's perception of assessment and feedback as a concept. Questions 3, 4 and 5 explored the personal experiences of assessment and feedback of the students in the focus group.

For question one, focus group students (FGS) identified assessment was used for progress checking. There was no mention of using assessment *for* learning, it was discussed purely from the perspective of providing certification and checking competencies (Washer, 2007).

FGS1: *Try and make sure you have the capability to progress onto the next stage.*

FGS2: *To gage a rough idea of where you are and how you've come to terms with the material.*

FGS3: *I'd say that it could also be used by metrics for teachers to see if the way they are teaching has been understood and has been taken in.*

FGS4 did not expand on these comments, instead they indicated their agreement. Furthermore, when answering question two only FGS3 stated that feedback was used to *"see where you went wrong and see how you can improve in the future."* The focus for these students were on the summative aspects of assessment and feedback, they considered assessment to be about providing certification only, as barriers to pass to show a mastery of the subject. The reason for this becomes evident when exploring what types of assessment the students had previous to studying in higher education.

FGS2 "*There's definitely less exams overall. We have like less exams now, and have six modules overall in this, while in the past like in GCSE we have had like 10 different subjects. Some of them with multiple exams."*

FGS3 *"The main difference for me, I don't know if it's the same for everyone else but em for my GCSE and A level everything was exam based. I had no coursework based subjects at all, whereas now here there's more coursework."*

FGS4 *"Yeah I think it's more individual, like work by yourself of assessment. You can't really get through with just the lecture material for most assessments."*

FGS1 did not comment for this question. The responses suggest that prior to higher education students only had assessments as end point exams, where the aim of the assessment was to just test a student's understanding of the material with limited experience of assessment *for* learning.

When answering question 3, FGS2 identified that students get given one-to-one feedback from tutors for exams and that NESS was used to give feedback for assignments. FGS1 and FGS3 agreed with FGS2 and identified that they each preferred face to face feedback over that of NESS, which *"did its job"*. FGS4 however identified that they found written feedback to be problematic.

FGS4 "*Sometimes it's great feedback and it can have parts that I understand, or parts that they are saying is wrong. It's like you know you're not completely correct but it's hard to know exactly where you went wrong. Sometimes they just say something quite vague. I think, if you don't understand the assessment very well it can be quite difficult to understand what they're saying."*

This is an identification of an issue that students have regarding feedback and could be why for the NSS statement "feedback on my work has helped me clarify things I did not understand" was low (Table 1, Page 2). The response from FGS4 suggests that the feedback is identifying where the student went wrong, but as it does not explain why they were wrong or clarify that misunderstanding, it is not useful in helping the student to understand further. This may be a factor that other students experience and could be why face to face feedback is preferred, the students get the opportunity to ask further questions. The aspect of students not finding feedback to be very clear also shows a lack of communication as identified by Wilson (Wilson, 2012). It suggests that after receiving feedback students do not ask for further clarification from the assessor, therefore the assessor assumes that the students understand what they have been given.

Students were also asked if they understood the purpose of the assessments that they were doing. Students focused on the assessments that they found to be not relevant, these assignments typically focused on the development of skills beyond that of programming.

*FGS1 "I mean I guess to make sure that we can implement what we have been taught in the lectures, to transfer out, to implement it into actual code."*

FGS2 "*You had to make a video with your group I didn't feel that was very relevant towards the course really. It was about technical information, so it was good from a technical standpoint, but it involved stuff like filming, editing which is quite fun but it's not really relevant to the course I feel like*."

FGS3. *"We had sprints last semester which was in the middle of all our programming deadlines, and it was looking at ethical disasters, and I understand the point of a sprint is to make sure you can work under pressure, but I just feel like it was a bit, I don't know unrelated."*

FGS4 "*I don't think, I think eh software engineering professional module has some assignments that have the most complained about. People saying, they don't really help us and that just makes them useless really. So, a way for us to spend our time."*

The responses from the focus group students show evidence to the common problems in assessment and feedback as explored in Section 2.3. The challenge that students face when extrapolating the skills learned to other assessments, such as problem solving, research skills and communication may suggest a lack of assessment literacy. It could also be a result of the links to other modules and assessments being lost due to the modularisation of courses (Ferrell, 2012).

***4.3.2 Student Interviews***

For the interviews, four first-year students and one third-year student attended. These semi-structured interviews enabled other themes to be explored if students or staff wished to discuss them. For a justification of why student interview questions were diffident to the focus group questions see Section (3.4). The interview questions were as follows.

1. What do you think assessment is for?
2. What do you think feedback is for?
3. How do you use feedback?
4. What feedback types work for you? What feedback types do you prefer?
5. What do you think the problems are with the current assessment and feedback practices? And what are the good things?
6. What has influenced the way you view and use feedback?
7. Is it always clear why you are doing a specific assessment?

These questions each explore a different aspect of assessment and feedback within higher education. Question one, two and seven explore students theoretical understanding of assessment and feedback, while questions three, four, five and six explore student's practical experiences with regards to assessment and feedback.

***4.3.2.1 Student Experiences of assessment***

Two main themes were identified in student responses to question one (what do you think assessment is for?) The first theme identified is similar to the responses of the focus group, staff use assessment points to evaluate the knowledge of the students.

*Student A "Assessment is a way to track progress."*

*Student D "We are learning the material that we are being taught."*

However, unlike the responses within the focus group, the second purpose of assessment identified by students was to help teachers identify how their teaching is progressing. This would allow the teacher to see if there are any gaps to a student's knowledge, which needed addressing, or changes to the material that they need to make.

*Student C "I think it is so that both the student and the staff can see where they are at in a certain topic."*

An issue identified in Section 2.3 (page 14), was a lack of student assessment literacy, however, the responses to this question show that the students interviewed had a good understanding of the purpose of assessment. The comments from the students suggest that they see assessment as guides to their learning and as tools to provide an understanding of their current knowledge. It also suggests that students are aware of how assessment benefits their learning beyond providing certification. It is promising that students are aware that this is a purpose of assessment, that it can be used beyond a 'test' of knowledge (Washer, 2007).

From the responses of question one, it could be suggested that students understood the purpose of assessment. A common theme that arose when discussing assessments with students was the type of assessment that students have completed. Similarly, to the focus group responses, the students identified that there was a switch from exam based practice to a more coursework style practice when moving from secondary education to higher education.

Student D: *"Most of mine were exams."*

Student A commented that at university the assessment was more than just a test, they elaborated that assessment and feedback was also used to help assist with learning.

Student A *"I'd say now that there is a lot more feedback to it maybe that is why I see assessment as more than tests."*

These responses could suggest why students could be seen as being highly mark driven when entering higher education. Students may be used to assessment fulfilling one purpose, that of providing certification, whereas, in higher education, this is a part of its purpose (Washer, 2007).

To further explore this, students were also asked which type of assessments they had come across in their time at university, the students all answered differently. All five students mentioned programming in their response. However, while coding was the primary form of assessment students mentioned, there were further assessment types outlined:

- Student A: Maths and essay writing
- Student B: Essay writing
- Student C: Presentations
- Student D: Sprint Exercises (a sprint exercise is a 24-hour research task)
- Student E: Program design and web development

Good practice in higher education when developing assessment is to have a large variety of assessment types (Brown, 2005) which allow students to gain multiple experiences with different types of assessment and allows the strengths of each student to show. A variety of assessment types allow students who are weak at a particular area, such as code writing to have a chance to excel, by completing assessments that focus on skills they feel comfortable with, such as essay writing. An issue within assessment practices can be that assessors will use one type of assessment, which does not benefit the development of skills nor reflect a real-world scenario (Brown and Knight, 1994; Ferrell, 2012). As students are identifying a range of assessment types within their degree, it shows that assessment variety is being included, and ensures that students with different strengths all get chances to showcase their skills.

The responses to question one suggest that students are aware of why assessment is used on a general level and how it is beneficial to their studies. The next step was to investigate if students understood the purpose of each assignment. Excluding student C, the perception of students for this question was no. Student B reported that it was "hardly ever" clear why an assignment was assigned. It is worth noting that student C was a third-year student, while students A, B, D, and E were first-year students.

When interviewing students, first year students placed their focus on programming assessments. The emphasis for student A was placed not on the coursework assignments, but the formative practical exercises set in class. When asked "Is it always clear why you are doing a specific assessment?' student A reported that, regarding the programming exercises, it was not clear why they were doing them.

*Student A: "Maybe not so much the exercises for programming. I feel like they were just too random."*

When considering the coursework, programming assignments were given a more positive viewpoint by the students. However, when considering assignments that required skills outside of programming, the responses were less positive.

Student E: "*Like in programming its, oh if you do this bit it will teach you this, but for instance, software engineering; some of the practicals or some of the coursework, I sort of understand why we have had to do it, but it's not really clear why*".

Student B: "*Like hardly ever. I can see it in like, sometimes they will explain it to you. And sometimes its 'here is your coursework.'*"

*Student C: "Generally, yes."*

Struyven and colleagues reported that if students find the work to be unsuitable for what they are trying to achieve or believe the assessment to be in an unsuitable format, students will disengage (Struyven et al, 2005). Therefore, the coursework that students had to complete outside of programming assignments may have been a cause of student dissatisfaction. Student A said *"there's more parts to being a computer scientist than what people see"* however, from the responses of students, it is clear that programming is their primary assessment focus

#### *4.3.2.2 Student experiences with feedback*

The responses to the assessment questions suggest that students understand the purpose of assessment, if not how their assessment aligns with this purpose. This section will explore students understanding of feedback and how this compares to pedagogical literature. In their responses to the question one, students have previously mentioned feedback and its uses in facilitating learning. To expand on these answers, interview question 2 asked students, what do you think feedback is for? In their answer's students focused on the formative aspects of feedback, student A, B, D, and E said that feedback was to help you improve in the future*:*

Student A: *"Just to give you an idea of where you can improve. It's constructive criticism."*
Student B: *"I'd say it is to help you on the next assignment. Or to let you know where you are."*
Student E*: "apply ways to hopefully improve in the future."*
Student D*: "To help us learn from our errors."*

Once again, the student who answered differently was student C, a student in their third year. They responded that feedback is to *"Support Assessment."* These responses show that students think feedback should be used for future improvements and correcting misunderstanding, comments made by students on mark justification were minimal (Hughes, 2011).

To understand if good practice identified in the literature was being used, it was first important to understand what types of feedback students had experienced, and this was explored through question 7.

The first type of feedback discussed by all participants was written feedback via the Newcastle eLearning Support System (NESS), this is the primary form of feedback given to students in the school Computing within Newcastle University. The second type of feedback students discussed was oral feedback, given to students in practical lab sessions (see Section 3.4 for an explanation of these lab sessions).

Student A: *"On the NESS system with written feedback. Visual help as well, such as in tutorials."*

Student B: *"Got a lot of written. Not always that great, but you get a lot of it. But a lot of it is normally just 'insert comment here'."*

The 'insert comment here' text is prepopulated text within the NESS system, if this is being displayed to students this is due to the assessor not removing these comments.

Students were then asked what type of feedback they preferred, two of the students preferred written feedback and two preferred face to face feedback. The preferred style was based on what type of assessment the student was considering. For programming and exams, the respondents stated that face to face is usually better. For essay-style assignments, the students reported that written feedback was the preferred method.

To further explore student perceptions of feedback, interview question 3 asked students how they used feedback once they had been given it. The responses to the use of feedback were not as uniform as the understanding of the purpose of assessment. Student A and Student D reported that they tried to use the feedback to develop goals to help with similar work in the future.

Student A: "*When I am doing my next piece of work, I take that feedback and try to make sure I stick to the targets which I set myself from the feedback.*"

*Student D: "My next step is to look again at my work and uh, first of all, try to understand the feedback and see if I do. Usually, I do. Then I do my best to keep it in mind when I do my next piece of work."*

Student B, however, did not report having as positive of a use for the feedback stating that the feedback was discussed with a friend, to compare the quality and author of the feedback.

Student B: "*I usually compare feedback with friends, but it is usually comparing how bad the feedback is and comparing who gave your feedback.*"

This comparison, not only of the quality of the feedback but the author of the feedback suggests that to students, the author is just as important as the content. Finally, student E said that they did not use the feedback due to feeling that the feedback was too generic.

Student E *"I don't really use feedback as much, normally because I'm just getting the same things over and over again."*

While it could be true that the feedback the student is receiving is too generic, it could also be the case that similar mistakes are being made within the work submitted and that they are not being corrected due to the student not using or understanding feedback.

An important fact to consider is how students have been taught to use feedback. Students were also asked, what has influenced the way you use feedback? There was no mention of an academic form of training, any guidance was often from a family member or self-training.

Student D: *"Mostly, the thing that influenced my Dad would go through the work with me."*

Student E: *"Basically, just a case of what I have done."*

Student C: *"Previous experience really."*

The lack of formal training with how to use feedback could be a supporting factor in why feedback can sometimes become used as mark justification. If students are not taught how to use feedback to structure learning, identify gaps in their knowledge and plan for future learning, they cannot gain the most benefit from it. This skill development is achieved by identifying the specific improvements that have been made from the previous assignment. By identifying this change, it can train students to regulate and plan their learning through practice to allow them to identify improvement areas on their own and therefore become more independent learners.

Finally, it was explored within the interviews what students thought was good and bad about the assessment and feedback practices they had experienced. Student A reported that they would prefer some instances of why the work was good, as opposed to lists of ways to improve as *"the negatives can be detailed, but sometimes the positives aren't."*

Student E also expands on this point, identifying that they "need more stuff about why it's good, or where I could actually improve." This can lead to issues of lack of confidence within students if the focus is on what the students did wrong, without giving context for what was correct. This lack of positive reinforcement can cause students to become demotivated and encourage the idea that a student cannot complete the task, therefore emphasising a fixed mindset (Dweck et al., 1998). While it is important to point out where the student made mistakes, it is also important to emphasise where the student did something well. The 'feedback sandwich' is a technique which can assist with this development. The feedback sandwich model ensures that an improvement a student can make has a positive comment before and after it (Leibold et al., 2005). While this runs the risk of improvements being buried within the feedback, it reduces the impact of students feeling too much negativity.

Student C commented on comparing feedback with other students where they "*were told they shouldn't have done the same thing that someone else got praised for."* This shows evidence of feedback being used as mark bartering to find out where students marks have differed in the cohort (Matshedisho, 2020). This theme was also further explored by student C identifying that within their maths module students are given general class feedback, and it would be further helpful to see a class average to understand where they are with regards to other students. A common theme across the students' interviews is the use of feedback from the student perspective for understanding mark loss and mark gain. From the literature perspective, this is not the aim of feedback. This mismatch in expectations of feedback may be a contributing factor to student dissatisfaction.

### *4.3.3 Staff Interviews*

The teaching staff who were interviewed were also asked the questions given to students, with the focus being from the assessors' perspective not the students. The questions were as follows

1. As a developer of assessment, what do you think is its purpose?
2. What is the most challenging aspect of assessment creation?
3. What do you think the purpose of feedback is?
4. What is your preferred method of giving feedback?
5. In your experience, do you think students understand the point of every assignment?
6. Do you think students are using feedback in the way you expect?
7. What would you consider to be the best assessment and feedback practice?
8. What do you think are the challenges of assessing programming?
9. What skills are most important to test during a programming assessment?
10. What are the differences between assessing programming assignments and other forms of assignments?

#### *4.3.3.1 Staff perceptions of assessment*

The responses that teaching staff had when responding to question one was similar to the student responses with two identified themes. The first theme that staff commented on was to test the understanding of the concepts taught within the lectures i.e. summative assessment.

Teacher A: *"It is assessing student abilities"*.

Teacher A also indicated that the purpose of assessment changed depending on the form the assessment took. For coursework, it was better when *"Knowledge and learning are applied through experience,"* whereas exams ensured that students worked on the assessment in a controlled environment. Teacher A also discussed a surface learning approach (Struyven et al., 2005) that students take for exams where they *"remember things for that morning"* and do not gain a deep understanding. While this teacher identified differences regarding formative and summative practices, the difference was discussed in the feedback to the students, not in the design of assessment.

Teacher B reported that assessment was used *to "test the students understanding of the concepts"* that were introduced in class, whereas, Teacher D reported that assessment was used to find out "*what they know and do not know".* Thus, providing certification and facilitating learning.

In question 5 staff were asked, do you think students understand the point of every assignment they have been given? The consensus was no. Staff members said that there was an effort to explain to students how the assignments were relevant to their work, however, their efforts were met with various degrees of success. These attempts involved forms such as objectives in the coursework, guidance through lectures and linking skills with other modules. An issue with these methods is that they require engagement with the material beyond the basic interaction that students give to coursework specifications. As discussed previously (Section 2.4.2 page 23) if students do not believe that the work is of appropriate form or useful too them, they will disengage, which results in these materials being underutilised (Struyven et al., 2005).

Teacher A *"Assuming they come to the lectures, they hear some pretty extensive explanations."*

Teacher B *"I think some students view certain ones such as reflective reports, for instance. I have a strong suspicion that some students see that as a waste of time."*

Teacher E *"In the specifications I will say, these are the objectives ……. students don't always listen if you tell them that during the lecture."*

Teacher C explained that students generally understand the purpose of the assignments but may not appreciate why it is important. Despite efforts to engage students with objectives, staff reported that they still received questions such as "why did I get this mark?" or "I did this fantastically why didn't I get the mark?" when the task/function the student was referring too was not what was being assessed. It suggests that students see little value in knowledge that does not contribute directly to mark gain or that the value of this work has not been fully explained (Nilson, 2013).

#### *4.3.3.2 Staff perceptions of feedback.*

Similar to exploring themes of assessment, question two asked what the purpose of feedback was. There were again two categories of responses, the first being to help students develop.

Teacher A *"It's to help students develop further."*

Teacher C *"The purpose of feedback was to help students develop their understanding based on the fact the assessment gives you an indication of their current understanding."*

Teacher E *"A student can understand where they have gone wrong and how they can improve in the future."*

Secondly, staff members also focused on mark driven aspect of feedback. They discussed that feedback is used to help students understand what marks they gained or lost.

*Teacher B "What they did correctly and the things that they did not do correctly on a specific assignment."*

*Teacher E "Being able to understand where they have lost marks, what the mark actually corresponds to and what was expected."*

These responses fall into two distinct categories. Without mentioning the topics, the staff members have described the processes of summative and formative feedback. The mark driven nature of students could also be linked to the assessors' agreement that feedback should be used to help clarify mark allocation.

When exploring what forms of feedback staff preferred to give to students the two same threads appeared, three out of the five staff members (teachers C, D and E) explored that NESS was the primary form of giving written feedback. However, Teacher C expressed that they would prefer not to give it via NESS but was often necessary. Teacher A explored giving feedback through demonstrations, identifying where common weak points were and having a dialogue with the students. Teachers A, B, C and E explore that the feedback needs to be beneficial to the students. Interestingly where students focused on the medium in which feedback was delivered, staff focused on the structure of the feedback.

Teacher B *"I will say for instance this is question one, this is all your feedback for question one, this is the feedback for question two, so I usually do it chronologically."*

Teacher C *"To make feedback most useful, it's beneficial for the student, you have to make it engaging and you have to do it at the right time and that is almost always not the time frame that's given for written feedback."*

Teacher E *"A few notes delivered via NESS which explain what they have done well, what they haven't done well and combine that with a sample answer that basically gives them what I expect."*

It is within this exploration of feedback we begin to see the differences in the perceptions of staff and students regarding feedback practices. Theoretically, both perceptions aligned, in practice, however, there is a divergence in perceptions. The staff focused more on the feedback being used for developmental purposes. It is interesting to note that the form of feedback delivery mattered little to the staff members. The focus was on how the students could improve, not whether the feedback was written or oral. These differences identify a potential source of dissatisfaction. By placing the focus on the medium of delivery and not the structure students may miss some of the nuances within the feedback.

Question 6 asked staff members if they thought students were using feedback in the way that they expected them too. The answers received were: *"no"*, *"I hadn't thought about it"* and "it's *hard to measure"*. When exploring this further with staff, they reported that feedback was sometimes used more for mark bartering than the expected reflection on the work that was being done. This supports a study explored in chapter two by Matshedisho, where marking schemes were replaced by marking rubrics. Despite the removal of individual marks, Matshedisho reported that students were still primarily mark driven. This mindset is also understandable when investigating how students viewed assessment in their pre-university experience, where assessment was used primarily as a testing method and not as a way to facilitate learning (Matshedisho, 2020). A cause for student dissatisfaction could be the mismatch of expectations, where assessors expect students to focus on the knowledge gain and future uses of feedback whereas students have a high focus on how feedback can be mapped to marks.

Questions 8, 9 and 10 explored the creation of programming assessments specifically and the differences to other forms of assessment.

Question 8 asked staff what the most challenging aspect of creating programming assessments was. When considering all forms of assessment, each staff member reported that fairness and finding something accessible to all students, which would allow a reasonable attempt at a grade but also to challenge students who excelled at the subject was difficult.

Teacher D stated that it was a challenge to *"set something which is not so large that it is going to take an enormous amount of time to do, but not totally trivial"*. When this is not achieved, it creates a "double bump" a concept explored by Scott (Scott, 2003). The double bump occurs when the basics and the advanced material is assessed, but the concepts with medium difficulty (Levels 3 and 4 of the Bloom Taxonomy) are not assessed, thus alienating students with an average performance. If the double bump is not considered in assessment development, it could also be a cause of dissatisfaction for 'at risk' students who cannot complete the harder levels but do not have access to simpler material.

A further challenge identified by staff was the need to scale the difficulty of assessments, exams specifically, based on the performance of the previous year's abilities. If the previous cohort found the work too easy, then the difficulty had to be increased. If the previous cohort found the work to be too difficult, then the current assessment had to have its difficulty decreased, however this was something that had to be considered on a cohort by cohort basis.

Interestingly when asked about assessing programming specifically, the answers were different for each tutor. Teacher A identified that plagiarism was a challenge to identify *"There's some* *pretty clear challenges around plagiarism."* This was also identified by Daley and Waldron, as there can be similar solutions to basic programming tasks, such as measurement conversions, plagiarism can become difficult to detect (Daley and Waldron, 2004; Bradley, 2020). Moss and colleagues identified reasons why students plagiarise, three of these reasons are as follows: (i) to artificially inflate grades, (ii) extenuating external circumstances such as an ill family member may cause a student to plagiarise to maintain grade levels and (iii) a lack of interest in the material (Moss, 2018). Each of these approaches show a student using a surface learning approach alongside plagiarism to attempt to achieve a grade which they would receive with a mastery of the material. There are two main issues that stem from a student plagiarising, the first is an issue with academic misconduct. The second is a student's lack of understanding of the material. A student who plagiarises and uses a surface learning approach will not gain an understanding of the material that they are copying. A core skill in learning how to code is creativity and problem-solving and therefore require practice and time to master.

Teacher B said that the difficulty was getting students to understand the standard solution and handling the non-standard ones. Teacher C reported that a difficulty when assessing programming was getting students to distinguish the functional and non-functional requirements. Students often fail to recognise that there is more to writing a program than producing the output. If the solution successfully produces the functional requirements that can only be part of the solution. Program design, good programming practice, and code comprehension are important factors to consider when creating assessment (Bailey and Stefaniak, 2001). The lack of recognition that design and good practice is something to consider when submitting a piece of work can stem from a lack of understanding regarding the assessment criteria. These are skills which cannot be gained from a surface learning approach, and instead requires a mastery of the subject. This was also explored by staff members who identified that the most challenging aspect to assessment that students faced was problem-solving and the ability to break a problem down into its individual parts from the original specification.

Question 10 investigated the different skills that programming assessments should assess compared to that of traditional university assignments, such as essay writing and report writing. For staff the main differences were the structure of how the student went about the work.

*Teacher B "So for written work, it's best practice to back up every point you make with something from the literature if you are trying to make an argument for doing something a certain way. Whereas with programming assignments they just have to do it if you see what I mean."*

There was also an expectation that within programming, there was a greater possibility for creativity and deviation from the normal that was not present within traditional assignments. From the responses of the staff, this indicates that the *type* of assessment is not as important as the requirement for assessment to follow good practice.

### *4.3.4 Interview and Focus Group Summary*

The interview and focus group responses provided an initial understanding of the causes of student dissatisfaction in the school of Computing, however, due to the limited amount of responses from both staff members and students it is difficult to generalise these results to a wider context.

These interviews identified the following:

- Feedback was sometimes considered to be too negative.
- Feedback could be unhelpful to students who did not understand the material as a whole, as it would often say that something was incorrect but not why it was incorrect.
- A lack of communication between teacher and student may result in students misunderstanding the feedback and its purpose.
- Identifying skills between modules and assessments were difficult for students.
- Good practice was used with regards to a variety of assessment types being used.
- It wasn't always clear to the students why they were doing a specific assessment.
- NESS (the environment used to give feedback to students) introduced issues such as "Insert comment here" which reduced students confidence in the feedback.

### *4.4 Assessment Experience Questionnaire*

The Assessment Experience Questionnaire (AEQ) (Gibbs et al., 2003) was adapted and given to students to investigate student perceptions regarding assessment and feedback. In total, 535 responses were collected and analysed during the three academic years of 2016 to 2019. Table 6 shows the distribution of the number of students who responded by cohort year. The questionnaire was given to students during their CSC1021 module after they had received the results and feedback of the first assessment submission.

| Cohort Year | Number of student responses |
|---|---|
| 2016/2017 | 161 |
| 2017/2018 | 147 |
| 2018/2019 | 227 |
| **Total** | 535 |

*Table 6 Number of Student responses for AEQ*

As the information was anonymised, no personal data is available for the students who responded. Table 6 shows the percentage of students who responded, in total 54.8% of students who could have responded filled in the AEQ.

| Academic year | Percentage of responses from stage one (%) |
|---|---|
| 2016/2017 | 61 |
| 2017/2018 | 52 |
| 2018/2019 | 74 |

*Table 7 Assessment Experience Questionnaire Respondents*

While not every student answered every question, it was decided to include all student responses and to analyse the questions that had been answered. The exception to this was when a correlation was needed between answers that had not been reported. These answers still gave an understanding of the perceptions of students in those areas. For example, not answering questions on feedback practices does not disregard a student's opinions on assessment practices.

Through the use of descriptive statistics, it can be seen that students within first year show signs of agreement with low satisfaction of assessment and feedback (as shown in the NSS results as seen in Table 1, Section 1.1). Table 8 shows the mean score for the questions on the AEQ. The AEQ uses a Likert scale system with the following values 1 = 'strongly disagree', 2=' disagree' 3= 'neutral' 4 = 'agree' 5=' strongly agree'. The initial descriptive statistics from Table 8 give a basic insight into student perceptions regarding assessment and feedback practices within higher education. The mean responses to statement 16 (2016: 3.01, 2017:2.35, 2018 3.20) and statement 20 (2016: 3.27, 2017: 3.71, 2018: 3.14) show that the perceptions of students align with the NSS statement “feedback on my work has helped me clarify things I did not understand.” While neither response is on average negative, the neutral responses suggest that students are not satisfied with feedback helping to clarify misunderstandings.

**Report**

Mean

| | | Cohort Year | | |
|---|---|---|---|---|
| Statement Number | Statement | 2016/2017 | 2017/2018 | 2018/2019 |
| 1 | I do the same amount of study each week regardless of whether an assignment Is due or not | 2.13 | 2.29 | 1.96 |
| 2 | I can be quite selective about what I study, learn, and still do well. | 3.43 | 3.42 | 3.26 |
| 3 | I only study things that are going to be covered in the assignment. | 2.80 | 2.73 | 3.15 |
| 4 | I have to study regularly if I want to do well on the course | 3.37 | 3.72 | 3.63 |
| 5 | On this course, it is possible to do quite well without studying much | 2.74 | 2.04 | 2.39 |
| 6 | IN weeks when the assignments are due I put in many more hours. | 4.12 | 4.16 | 4.34 |
| 7 | Tackling the assignment really makes me think | 4.07 | 4.07 | 3.84 |

| Statement Number | Statement | 2016/2017 | 2017/2018 | 2018/2019 |
|---|---|---|---|---|
| 8 | I learn more from doing the assignments than from studying the course material. | 4.06 | 3.59 | 3.42 |
| 9 | In completing the assignment you can get away with not understanding and still get high marks. | 2.78 | 2.36 | 2.69 |
| 10 | The assignments give very clear instructions about what you are expected to do | 2.84 | 2.55 | 2.81 |
| 11 | When I tackle an assignment it is not at all clear what would count as a successful answer | 3.06 | 3.11 | 3.31 |
| 12 | The assignments are not very challenging | 2.45 | 2.21 | 2.20 |
| 13 | On this course I get plenty of feedback on how I am doing. | 2.96 | 3.34 | 2.90 |
| 14 | The feedback comes very quickly. | 2.38 | 2.43 | 2.16 |
| 15 | There is hardly any feedback on my assignments when I get them back | 2.87 | 2.13 | 2.81 |
| 16 | When I get things wrong or misunderstand them I don't receive much guidance in what to do about it | 3.01 | 2.35 | 3.20 |
| 17 | I would learn more if I received more feedback | 3.72 | 3.66 | 3.83 |
| 18 | Whatever feedback I get comes too late to be useful. | 2.99 | 2.69 | 3.45 |
| 19 | The feedback mainly tells me how well I am doing in relation to others. | 2.49 | 2.35 | 2.30 |
| 20 | The feedback helps me to understand things better | 3.27 | 3.71 | 3.14 |

| Statement Number | Statement | 2016/2017 | 2017/2018 | 2018/2019 |
|---|---|---|---|---|
| 21 | The feedback shows me how to do better next time | 3.38 | 3.96 | 3.31 |
| 22 | Once I have read the feedback I understand why I got the mark I did, | 2.91 | 3.51 | 2.90 |
| 23 | I don't understand some of the feedback | 3.04 | 2.80 | 3.22 |
| 24 | I can seldom see from the feedback what I need to do to improve. | 2.92 | 2.59 | 3.00 |
| 25 | I read the feedback carefully and try to understand what the feedback is saying. | 3.93 | 4.18 | 3.85 |
| 26 | I use the feedback to go back over what I have done in the assignment | 3.23 | 3.62 | 3.29 |
| 27 | The feedback does not help me with any subsequent assignments | 2.71 | 2.37 | 2.78 |
| 28 | The feedback prompts me to go back over material covered in the course. | 2.88 | 3.29 | 2.92 |
| 29 | I do not use the feedback for revising. | 3.17 | 2.76 | 3.15 |
| 30 | I tend to only read the marks. | 2.30 | 2.19 | 2.54 |

*Table 8 Average means per cohort year*

When completing the assignments set, students in this case study report that they do not use a strategic form of learning (Nilson, 2013) to complete the assignments. Evidence for this can be seen from the average mean responses to statements 2, 3, 5 and 9. The responses to these statements show that from the perspective of the students, the assessment encourages them to have a deep understanding. The responses to the statements also show that the good practice identified from the literature is being followed within this institution, such as encouraging students to work at the higher levels of Bloom’s taxonomy (Bloom, 1956).

There are conflicting results concerning the quantity of feedback. The results from statement 15 suggest that students get an amount of feedback that they consider to be enough, while the results from statement 13 show that students do not consider the feedback that they receive to be "plenty". The amount of feedback a student receives is less important than the quality of feedback (Price et al. 2010). However, there needs to be enough feedback to help a student learn. Investigating the wording of these statements gives a more detailed understanding of student perceptions. The words "hardly" and "plenty" suggest two ends of an extreme, where hardly is defined by not enough, and plenty is an excess. When considering the response to question 17 however, it can be seen that students do not necessarily agree that getting more feedback would help them learn more.

A clear negative from the students was regarding the speed at which feedback was returned. Statement 14 has a mean score of 2.38, which shows that students are dissatisfied with the speed at which their feedback is returned. The speed at which feedback gets returned to students a crucial part of processing feedback. Marin states the feedback needs to be returned promptly for students to gain an impact from it (Marin, 2009). The disagreement and neutral results from statement 18 tell us that feedback is returned in a time scale to be beneficial to students for the next assignments. Finally, an initial exploration statements 26, 28, 29 and 30 show that students are engaging with the feedback at some level.

### *4.4.1 ANOVA Test*

To further explore the potential impacts of these results, one-way analysis of variance (ANOVA) test will be used to investigate the changes within the year group and identify potential impacts to these results. The students were split by the academic year in which they took the questionnaire so that responses of each year could be compared.

#### *4.4.1.1 Perceptions of assessment*

**Result:** Table 8 shows the results of a one-way between-groups analysis of variance was conducted to explore the impact of year group on statement 10 "The assignments give very clear instructions about what you are expected to do". Students were divided up by year group in which they took the questionnaire. The groups were split by the academic year as follows: 2016/2017 (M=2.84), 2017/2018 (M=2.55) and 2018/2019 (M=2.81). Using Post Hoc tests, it was found that there was not a statistically significant difference in any of the years.

Wakeford (2003) states that an important aspect of developing assessment is the clarity and validity of it. The results of this analysis show that students do not perceive assessment specifications to be clear. For assessment to be clear, the aim of what the student should produce needs to be transparent (Hughes, 2011). This transparency does not, however, mean that each process should be explained to the student. Clarity should ensure that a student can understand what the product should be. This result provides part of an answer to RQ2, what are the current perceived assessment and feedback issues experienced by students? The response shows that a perceived issue students' have, may stem from a lack of clarity about what they must do to complete their assessment, in this case, a programming assessment.

**Multiple Comparisons**

Dependent Variable: The assignments give very clear instructions about what you are expected to do

Tukey HSD

| (I) Cohort Year | (J) Cohort Year | Mean Difference (I-J) | Std. Error | Sig. | 95% Confidence Interval Lower Bound | Upper Bound |
|---|---|---|---|---|---|---|
| Academic Year 2016/2017 Stage1 | Academic Year 2017/2018 Stage1 | .291 | .129 | .064 | -.01 | .60 |
| | Academic Year 2018/2019 Stage1 | .035 | .117 | .953 | -.24 | .31 |
| Academic Year 2017/2018 Stage1 | Academic Year 2016/2017 Stage1 | -.291 | .129 | .064 | -.60 | .01 |
| | Academic Year 2018/2019 Stage1 | -.256 | .120 | .084 | -.54 | .03 |
| Academic Year 2018/2019 Stage1 | Academic Year 2016/2017 Stage1 | -.035 | .117 | .953 | -.31 | .24 |
| | Academic Year 2017/2018 Stage1 | .256 | .120 | .084 | -.03 | .54 |

*Table 9 Statement 10 The assignments give very clear instructions about what you are to do*

Furthermore, when exploring statement 11 "when I tackle an assignment, it is not at all clear what would count as a successful answer" there were no statistically significant changes for each of the year groups. However, when investigating the mean responses for this statement 2016/2017 (M=3.06), 2017/2018 (M=3.11), 2018/2019 (M=3.31), it can be seen that they are more positive than question 10. This result does suggest that while students are aware of what the product should be, e.g. the end program which has a specific set of functions, they are not necessarily clear on the individual steps needed to reach that functionality. Identifying what an end product should be as well as the criteria needed to complete the assessment show the use of good practice.

This is also emphasised by placing the importance of the assignment not on the end product, but the process and learning the student has done to reach that stage. By informing students of the desired product along with clarity about exactly what that should be, students have to focus on the process to get to the product and think about what needs to go into each step (Black et al., 1998; Gibbs and Simpson, 2003; Boud 2009, and Newcastle University, 2019).

There are two possibilities of why this lack of understanding of what a successful answer is could occur. The first is that students may lack a good level of assessment literacy. It may be that it is not clear to students how what they have learned can be applied to the current assessment. The second could possibly be that the coursework specification or the subject itself was not clear to the students, and therefore it was unclear how the assessment was intended to be completed. This may imply that good practice such as clear criteria referenced assessments (Hughes, 2011), may not be used when developing assessment, or that the use of good practice does not align with what students find to be clear. When investigating the syllabus and description of the module (Appendix A), we can see that the coursework outcomes align with the requirements of the syllabus, as stated in the module outline form (MOF). The coursework throughout the years of the study remained consistent, with only minor changes.

While there were some changes made to the assignment, they were largely cosmetic. Figure 9, 10 and 11 show the changes made to the specification (Appendix F shows the full specification). The 2017 specification gave further details concerning the constructs that had to be used to complete the assignment. When investigating the changes in 2018, the specificity and detail of the solutions are removed, leaving a larger scope for interpretation by the students.

From the perspective of the students, the analysis of statement 11 shows that changing the specification to be more specific has minimal impact on student perceptions as a whole with the consensus in each year being 'disagree'.

Part 1

BOGOFF want you to produce a `TaxCalculator` class with two public methods:

1. A method that, given a child's income, returns the tax payable on that income.
2. A method that, given a child's income, returns the income left after the tax payable has been deducted.

*Figure 9 2016 Assignment One specification*

Part 1

SOCS want you to produce a `MarkCalculator` class with two public methods:

1. A method `computeMarks`that, given an array of student exam and coursework marks, returns an array of returned module marks.
2. A method `computeResult` that, given an array of student exam and coursework marks, returns a Stage Result for that student.

*Figure 10 2017 Assignment One specification*

Part 1

Produce a `TaxCalculator` class with one public method:

1. A method that, given a child's income, returns the tax payable on that income.

*Figure 11 2018 Assignment One specification*

#### *4.4.1.2 Perceptions of feedback quality*

Table 9 shows the results of an ANOVA test that was conducted to explore the impact of year groups on statement 16, "when I get things wrong or misunderstand them I don't receive much guidance on what to do about it". Students were divided up by year group in which they took the questionnaire; the groups were split by the academic year as follows: 2016/2017, 2017/2018 and 2018/2019. There was a statistically significant difference at the $p<.05$ level in mean scores between each year. The magnitude of the differences within the means was a medium effect, with an ETA-Squared of 0.09. Post Hoc comparisons using the tukey HSD test indicated that the mean score for 2016/2017 (M=3.01) was significantly different from 2017/2018 (M=2.35). There was also a statistical significance between the years 2017/2018 (M=2.35) and 2018/2019 (M=3.20). Means did not differ significantly between groups 2016/2017 and 2018/2019.

**Multiple Comparisons**

Dependent Variable: When I get things wrong or misunderstand them I don't receive much guidance in what to do about it

Tukey HSD

| (I) Cohort Year | (J) Cohort Year | Mean Difference (I-J) | Std. Error | Sig. | 95% Confidence Interval Lower Bound | Upper Bound |
|---|---|---|---|---|---|---|
| Academic Year2016/2017Stage 1 | Academic Year2017/2018Stage 1 | .659* | .129 | .000 | .35 | .96 |
| | Academic Year2018/2019Stage 1 | -.196 | .116 | .211 | -.47 | .08 |
| Academic Year2017/2018Stage 1 | Academic Year2016/2017Stage 1 | -.659* | .129 | .000 | -.96 | -.35 |
| | Academic Year2018/2019Stage 1 | -.855* | .120 | .000 | -1.14 | -.57 |
| Academic Year2018/2019Stage 1 | Academic Year2016/2017Stage 1 | .196 | .116 | .211 | -.08 | .47 |
| | Academic Year2017/2018Stage 1 | .855* | .120 | .000 | .57 | 1.14 |

*. The mean difference is significant at the 0.05 level.

*Table 10 Statement 16 When I get things wrong or misunderstand them I don't receive much guidance in what to do about it*

The medium effect size of 0.09 suggests that the significant changes in the mean averages is large enough for statement 16 to be a factor in the perception of students. The response of students for the years 2016 and 2018 was neutral, while the 2017 cohort responded disagree. It is worth noting that when the average response was neutral, this was in the lower quartile <.25

showing a closer response to disagree than agree. These results suggest that feedback to the students showed not just areas where mistakes were made but how the student could also improve. Using feedback to identify mistakes is a common and important practice. What is also good practice when writing feedback is to show where students can improve as it helps guide students to plan their learning. Students need to be able to answer not just what their current progress is, but what they need to do next and how to get there (Ott et al., 2016).

When considering the change in perceptions of feedback, three factors could have affected the student responses. By 2017 pre- and post-moderation were existing processes that had been practised and developed within the school of Computing. Therefore staff members running pre- and post-moderation would be aware of what features needed to be identified within the feedback and could train markers to further comment on areas for improvement, especially when considering the same assignment had been used multiple times.

The second could be the result of the strengths and weakness of the students. If the 2017 cohort performed less well than the 2016 and 2018 cohort, it could be that students needed more guidance on how to improve their work and this was reflected in the feedback, thus gaining a better perception of the feedback used. This is reflected in the average grades for the first project, 2016: 77%, 2017: 64%, 2018: 69%. From these grades, we can see that the lowest-performing cohort on average was 2017.

It is important to also look at the changes made to the coursework specification, as discussed in the analysis of statement 10 and 11 (page 79). It is the further detail that was discussed in 2017, such as the type of data structure required to be used and the expected return type, that could have had an impact on the perception of feedback. The added detail to the specification of the project implementation could have two different impacts which would improve the response to statement 16. The first impact can be viewed from the perspective of the markers. By 2017 pre and post moderation was a process that had been implemented and improved since 2014, resulting in markers having more experience with error identification due to the similarities of the coursework. Furthermore, the constructs identified give markers more specific areas to target, such as, a student misusing arrays. Due to this there will be less variation in the solutions presented and the feedback.

As explored in Section 4.3, staff member C reported that where feedback is concerned, students often focused on the functional aspects of the code, as opposed to the other features such as code design. It could be that the specificity of array inputs and array outputs left students more satisfied with feedback and how they can generally improve, as these are tangibly correct or incorrect features that focus on the functionality of the code rather than the design or style.

Figures 9, 10 and 11 show that the specification for 2018 (Full specifications in Appendix F), is similar to that of 2016, with the removal of a method to return the income left after the tax payable has been deducted. This function is still expected to be completed by the student but is no longer explicitly defined. When investigating the change in perceptions to 2018 in table 7 we can see a decline from 2017 perceptions. In addition to the change of scenario the specification also removed the instructions regarding which data structure and return type was required. The decline in student perspective could suggest that the more specific the assessment criteria, the more specific the feedback can be. If students can easily match the feedback to the specification and the criteria this may result in better satisfaction.

The data presented suggests, for this cohort of students, that the addition of more detail within the specification, which follows good practice, has no impact on the students' perception of assessment but may have an effect on student perceptions of feedback. The main change within the delivery of the material was the changes to the coursework specification. Teaching staff on the module and the module content remained consistent.

#### *4.4.1.3 Developmental use of feedback*

To further explore student perceptions on feedback, statement 20 in the AEQ investigates the use of feedback for developmental learning. For this statement the phrase 'understand things better' has been interpreted as clarifying misconceptions that students have and informing the student of good practice when not followed.

**Result:** Table 9 shows the results of an ANOVA test that was conducted to explore the impact of a year group on statement 20 "The feedback helps me to understand things better." Students were divided up by year group in which they took the questionnaire, the groups were split by the academic year as follows: 2016/2017, 2017/2018 and 2018/2019. Despite having statistical significance, the actual mean difference was quite small. The effect size using ETA-Squared was 0.06. There was a statistically significant difference at the $p<.05$ level in mean scored. Post Hoc comparisons using the Tukey HSD test indicated that the mean score for 2016/2017 ($M=3.27$) was significantly different from 2017/2018 ($M=3.71$). There was also a statistical significance between the years 2017/2018 ($M=3.71$) and 2018/2019 ($M=3.14$). Means did not differ significantly between groups 2016/2017 and 2018/2019.

**Multiple Comparisons**

Dependent Variable: The feedback helps me to understand things better

Tukey HSD

| (I) Cohort Year | (J) Cohort Year | Mean Difference (I-J) | Std. Error | Sig. | 95% Confidence Interval Lower Bound | Upper Bound |
|---|---|---|---|---|---|---|
| Academic Year 2016/2017 Stage1 | Academic Year 2017/2018 Stage1 | -.440* | .114 | .000 | -.71 | -.17 |
| | Academic Year 2018/2019 Stage1 | .137 | .103 | .378 | -.10 | .38 |
| Academic Year 2017/2018 Stage1 | Academic Year 2016/2017 Stage1 | .440* | .114 | .000 | .17 | .71 |
| | Academic Year 2018/2019 Stage1 | .577* | .106 | .000 | .33 | .83 |
| Academic Year 2018/2019 Stage1 | Academic Year 2016/2017 Stage1 | -.137 | .103 | .378 | -.38 | .10 |
| | Academic Year 2017/2018 Stage1 | -.577* | .106 | .000 | -.83 | -.33 |

*. The mean difference is significant at the 0.05 level.

*Table 11 Statement 20 The feedback helps me to understand things better*

While the differences between the means in each year are small and do not move between boundaries, there are two factors to note. For the years which have a statistical difference at the $p=0.05$ level (2016 to 2017 and 2017 to 2018) the range within the boundary changes from the lower quartile < .25 to almost the upper quarter >.75. An effect size of 0.06 as Pallant explains, is a medium effect size and therefore the change in the mean is large enough for the shift in perspective to be significant (Pallant, 2010, pp 254). This neutral response from students suggests that it is difficult to tell whether feedback "helps students to understand things better" or not. However, as statement 16 (page 83) has a neutral or disagree response, we can see that students perceive that there is guidance on what to do when they misunderstand concepts. It could be that for simpler concepts, feedback does describe what the correct answer is and explain the why behind it, therefore helping to clarify misunderstandings. Whereas the more complicated aspects of the work may not be suitable to discuss in feedback, however the staff member could provide further resources for the student to use. The increase of mean occurs during the 2017 cohort year, this gives a further suggestion that the change to the assignment specification in 2017 was more challenging for students, therefore required more explanation within the feedback. Additionally, the specified data structures within the 2017 specification may allow for pre prepared feedback statements on the correct implementation to be used and therefore students would be able to receive more guidance.

There are some important things to note within the analysis of the results to this question. There is no preface to the question enquiring what mark the student received on their assignment, and it is not clear if the feedback the student received set out to help the student understand things better. For example, a student who received a mark of 95% may not require a further understanding as much as a student who received a mark of 50%. Feedback to students who received very high marks can still be beneficial if the feedback uses positive reinforcement to highlight what the student did well.

Encouraging students to reflect on the work where they have made mistakes or areas of general improvements is of vital importance. Simply informing students that something is wrong or using feedback for mark justification is not enough to support student learning (Ferrell, 2012). Good practice should encourage students to reflect on their work and identify areas in which they can improve (Kolb, 2014; Ott et al, 2016; Tai and Adachi, 2019, pp64).

The relationship between "when I get things wrong or misunderstand them I don't receive much guidance on what to do about it" and "the feedback helps me to understand things better" was investigated using Pearson's product-moment correlation coefficient, to investigate whether the type of feedback a student receives could change perspectives. There was a negative correlation between the two variables, $r = -.298$, $p < 0.001$. This a small negative correlation between these two statements, is to be expected. If a student has had mistakes identified to them, it would be expected that the feedback would also elaborate on their mistakes and help the student understand the material further. It is also important to consider that feedback may not be an appropriate place to explore the misunderstandings that a student has. For example, it may be more appropriate to signpost a student to resources as opposed to explaining a core misunderstanding and taking away emphasis from other bits of feedback. For a computer science student, it could be suggested that they ask for help and clarification within their practical lab sessions.

#### *4.4.1.4 Mark Driven students*

Ferrell, Nicol and colleagues, along with Whalley and colleagues, outline many studies that show why and how students are mark driven and explore the student-facing aspects of mark driven feedback. Ferrell also identifies that there are large quantities of summative feedback frameworks but a lack of formative feedback frameworks (Nicol et al., 2007; Whalley et al. 2011; Ferrell 2012). From these studies, it is clear that feedback, from the student's perspective, should help explain where marks were gained and where marks were lost. If feedback does not do this, it could be a cause of student dissatisfaction. Here we can see a mismatch between student expectations and desires (to be provided with clear mark identification) and what is considered good practice in feedback writing (to not be used as mark justification). However, as assessment should be criteria referenced, students who are assessment literate should be able to identify which aspects of the criteria they did not meet, and therefore why marks were not obtained (Hughes, 2017).

**Result:** Table 12 shows the results from a one-way between-group analysis of variance that was conducted to explore the impact of year group on statement 22 "Once I have read the feedback I understand why I got the mark that I did". Students were divided up by year group in which they took the questionnaire, the groups were split by the academic year as follows: 2016/2017, 2017/2018 and 2018/2019. There was a statistically significant difference at the $p<.05$ level in mean scores. Despite having statistical significance, the actual mean difference was quite small. The effect size using ETA-Squared was 0.05. Post Hoc comparisons using the Tukey HSD test indicated that the mean score for 2016/2017 (M=2.91) was significantly different from 2017/2018 (M=3.51). There was also a statistical significance between the years 2017/2018 (M=3.51) and 2018/2019 (M=2.90). Means did not differ significantly between groups 2016/2017 and 2018/2019.

**Multiple Comparisons**

Dependent Variable: Once I have read the feedback I understand why I got the mark I did.

Tukey HSD

| (I) Cohort Year | (J) Cohort Year | Mean Difference (I-J) | Std. Error | Sig. | 95% Confidence Interval | |
|---|---|---|---|---|---|---|
| | | | | | Lower Bound | Upper Bound |
| Academic Year2016/2017Stage 1 | Academic Year2017/2018Stage 1 | -.604* | .131 | .000 | -.91 | -.29 |
| | Academic Year2018/2019Stage 1 | .009 | .118 | .997 | -.27 | .29 |
| Academic Year2017/2018Stage 1 | Academic Year2016/2017Stage 1 | .604* | .131 | .000 | .29 | .91 |
| | Academic Year2018/2019Stage 1 | .613* | .123 | .000 | .32 | .90 |
| Academic Year2018/2019Stage 1 | Academic Year2016/2017Stage 1 | -.009 | .118 | .997 | -.29 | .27 |
| | Academic Year2017/2018Stage 1 | -.613* | .123 | .000 | -.90 | -.32 |

*. The mean difference is significant at the 0.05 level.

*Table 12 Statement 22 Once I have read the feedback I understand why I got the mark I did.*

In table 11, we can see that the same pattern occurs in statement 16 as statement 20. Student perceptions of feedback shift to be more positive in the year 2017. An ETA-squared of 0.05 is a medium effect size. This medium effect size suggests that the average response shifts from disagree to neutral is a sufficient size to be considered a noteworthy factor of student perceptions. A potential contribution to this change in effect size may be a lack of assessment literacy within the students, specifically that of understanding assessment specifications. If the assessments are criteria referenced (as seen in Figures 9,10,11) and the feedback refers to these criteria it should be clear to the students how the marks of the assignment align (Hattie et al, 2006; Hughes, 2011) .

#### *4.4.1.5 Forward facing feedback*

Wong and colleagues state that clarity of feedback is crucial for a student's learning, as not only can unclear feedback stop student progression but also cause regression of student understanding (Wong, 2010). This clarity can also be applied when considering how well the feedback aligns with the criteria of assessment. If it is not clear to the students how their mark relates to the feedback, this could be a cause of dissatisfaction (Hughes, 2017). This, however, also shows that while feedback is giving direction to students and suggesting improvements, it is not being used by markers as a way to justify the marks given to students.

**Result:** Table 13 shows the results from a one-way between-groups analysis of variance that was conducted to explore the impact of year group on statement 27 "the feedback does not help me on any subsequent assignment". Students were divided up by year group in which they took the questionnaire, the groups were split by the academic year as following: 2016/2017, 2017/2018 and 2018/2019. There was a statistically significant difference at the $p<.05$ level in mean scores. Despite having statistical significance, the actual mean difference was quite small. The effect size using ETA-Squared was 0.03. Post Hoc comparisons using the Tukey HSD test indicated that the mean score for 2016/2017 (M=2.71) was significantly different from 2017/2018 (M=2.37). There was also a statistical significance between the years 2017/2018 (M=2.37) and 2018/2019 (M=2.78). Means did not differ significantly between groups 2016/2017 and 2018/2019.

The responses to this statement have an average mean response rate of disagree. This suggests that while students are not confident with feedback telling them what they did wrong, it does inform them how they can improve on their work next time. As previously discussed, a large component of good feedback is forward-facing feedback, as feedback should also help with longitudinal development (Ott et al., 2016). An effect size of 0.03, however suggests that the changes in averages are minimal. The changes to the assessment specification in 2017 have had limited impact on this response rate. This is understandable as it is possible to correct a misunderstanding of a student and still inform the student how the solution should have been implemented. This shows that when considering the usage of feedback, students are aware of how to abstract the guidance they are given and to apply it to future assignments.

**Multiple Comparisons**

Dependent Variable: The feedback does not help me with any subsequent assignments

Tukey HSD

| (I) Cohort Year | (J) Cohort Year | Mean Difference (I-J) | Std. Error | Sig. | 95% Confidence Interval Lower Bound | 95% Confidence Interval Upper Bound |
|---|---|---|---|---|---|---|
| Academic Year 2016/2017 Stage1 | Academic Year 2017/2018 Stage1 | .334* | .118 | .013 | .06 | .61 |
| | Academic Year 2018/2019 Stage1 | -.073 | .107 | .772 | -.32 | .18 |
| Academic Year 2017/2018 Stage1 | Academic Year 2016/2017 Stage1 | -.334* | .118 | .013 | -.61 | -.06 |
| | Academic Year 2018/2019 Stage1 | -.407* | .110 | .001 | -.67 | -.15 |
| Academic Year 2018/2019 Stage1 | Academic Year 2016/2017 Stage1 | .073 | .107 | .772 | -.18 | .32 |
| | Academic Year 2017/2018 Stage1 | .407* | .110 | .001 | .15 | .67 |

*. The mean difference is significant at the 0.05 level.

*Table 13 Statement 27 The feedback does not help me with any subsequent assignments*

For example, when given feedback on how to implement an array, a student should be able to identify how to implement arrays in other situations as opposed to just the specific array mentioned within that assignment. This is important as the skills necessary to have a deep understanding of programming are often the transferable skills, such as problem solving and the mastery of concepts, rather than specific syntaxes (Lahtinen et al., 2005). From the statistical tests and the minimal effect size, we can see that good practice is being consistently followed in the form of forwarding facing feedback.

The results from statement 27 "the feedback does not help with any subsequent assignment" was disagree for all three years suggesting that from the perspective of students', feedback does help with future assignments. However, when considering the neutral responses to question 21 and responses from student interviews, (Section 4.3.2), it can be theorised that feedback identifies the mistakes that a student has made, but not how to correct them in the future.

**Multiple Comparisons**

Dependent Variable: The feedback shows me how to do better next time

Tukey HSD

| (I) Cohort Year | (J) Cohort Year | Mean Difference (I-J) | Std. Error | Sig. | 95% Confidence Interval Lower Bound | 95% Confidence Interval Upper Bound |
|---|---|---|---|---|---|---|
| Academic Year 2016/2017 Stage1 | Academic Year 2017/2018 Stage1 | -.581* | .113 | .000 | -.85 | -.31 |
| | Academic Year 2018/2019 Stage1 | .064 | .102 | .804 | -.18 | .31 |
| Academic Year 2017/2018 Stage1 | Academic Year 2016/2017 Stage1 | .581* | .113 | .000 | .31 | .85 |
| | Academic Year 2018/2019 Stage1 | .646* | .106 | .000 | .40 | .90 |
| Academic Year 2018/2019 Stage1 | Academic Year 2016/2017 Stage1 | -.064 | .102 | .804 | -.31 | .18 |
| | Academic Year 2017/2018 Stage1 | -.646* | .106 | .000 | -.90 | -.40 |

*. The mean difference is significant at the 0.05 level.

*Table 14 Statement 21 The feedback shows me how to do better next time*

**Result:** Table 14 shows the results from a one-way between-groups analysis of variance that was conducted to explore the impact of year group on statement 21 "the feedback shows me how to do better next time". Students were divided up by year group in which they took the questionnaire, the groups were split by the academic year as follows: 2016/2017, 2017/2018 and 2018/2019. There was a statistically significant difference at the $p<.05$ level in mean scores. Despite having statistical significance, the actual mean difference was quite small. The effect size using ETA-Squared was 0.08. A medium effect size of 0.08 suggests that while the mean values do not change Likert bounds the changes in perception to statement 21 are noteworthy. Post Hoc comparisons using the tukey HSD test indicated that the mean score for 2016/2017 (M=3.38) was significantly different from 2017/2018 (M=3.96). There was also a statistical significance between the years 2017/2018 (M=3.96) and 2018/2019 (M=3.31). Means did not differ significantly between groups 2016/2017 and 2018/2019.

It is noteworthy that the year which has the closest mean to agree was 2017. As discussed in earlier questions, this was when the assignment became more specific. By focusing on the specific methods in the feedback, such as the development of arrays, it will be clearer for students to see how this can be applied to future assignments.

To further explore if feedback focused on the negative aspects of the work as opposed to the facilitation of learning and forward-facing feedback, the perceptions of students regarding feedback being used to improve was investigated.

**Result:** Table 15 shows the results from a one-way between-groups analysis of variance was conducted to explore the impact of year group on statement 24 "I can seldom see from the feedback what I need to do to improve". Students were divided up by year group in which they took the questionnaire, the groups were split by the academic year as following: 2016/2017, 2017/2018 and 2018/2019. There was a statistically significant difference at the $p<.05$ level in mean scores. Despite having statistical significance, the actual mean difference was quite small. The effect size using ETA-Squared was 0.03. Post Hoc comparisons using the Tukey HSD test indicated that the mean score for 2016/2017 (M=2.92) was significantly different from 2017/2018 (M=2.59). There was also a statically significant difference between 2017/2018 (M=2.59) and 2018/2019 (M=3). Means did not differ significantly between groups 2016/2017 and 2018/2019.

Duncan explores that forward-thinking is an important aspect of feedback and feedback that informs the students on how to improve is often more beneficial to longitudinal development than mistake identification (Duncan, 2007). Students must understand how to develop in the future as opposed to just identifying where mistakes were made in their work.

While the average means for statement 24 shifts into the neutral category in the academic year 2018/2019, it is a small change with an effect size of 0.03. It is a positive result that students are consistently reporting feedback can be used for future improvement. However, an issue which has impacted the results is that students occasionally identified that they did not understand the word 'seldom' and therefore did not answer due to this. While the effect size of this statement was small, the average means of the statement follow a similar pattern to previous statements. There was an improvement in perception from the years 2016 to 2017, and a decline in perceptions from 2017 to 2018.

**Multiple Comparisons**

Dependent Variable: I can seldom see from the feedback what I need to do to improve.

Tukey HSD

| (I) Cohort Year | (J) Cohort Year | Mean Difference (I-J) | Std. Error | Sig. | 95% Confidence Interval Lower Bound | Upper Bound |
|---|---|---|---|---|---|---|
| Academic Year 2016/2017 Stage1 | Academic Year 2017/2018 Stage1 | .333* | .116 | .012 | .06 | .60 |
| | Academic Year 2018/2019 Stage1 | -.073 | .104 | .766 | -.32 | .17 |
| Academic Year 2017/2018 Stage1 | Academic Year 2016/2017 Stage1 | -.333* | .116 | .012 | -.60 | -.06 |
| | Academic Year 2018/2019 Stage1 | -.405* | .108 | .001 | -.66 | -.15 |
| Academic Year 2018/2019 Stage1 | Academic Year 2016/2017 Stage1 | .073 | .104 | .766 | -.17 | .32 |
| | Academic Year 2017/2018 Stage1 | .405* | .108 | .001 | .15 | .66 |

*. The mean difference is significant at the 0.05 level.

*Table 15 Statement 24 I can seldom see from the feedback what I need to do to improve*

This could also be due to the change in the style of the specification from 2016 to 2017. The more generalised specifications of 2016 and 2018 may not be able to have as clear feedback with regards to the criteria specified within the assignment. For example, a marker may be able to give more guidance on the use of arrays, as opposed to the use of lists. The more solutions used, the more difficult it is to produce consistent and detailed feedback.

A further possibility is that the 2017 cohort performed worse than the 2016 and 2018 cohorts. The average module grades for the programming module CSC1021 were investigated to see if this was true, they were as follows: 2016: 72%, 2017: 67%, 2018: 65%. As can be seen by the average CSC1021 grades, 2017 is not the weakest cohort, while the 2017 students performed less well than the 2016 cohort, on average they performed better than the 2018 cohort; therefore, this theory does not hold.

While there is a conflict regarding the usage of feedback in future assignments, it is important to investigate the wording of the questions. The main differences in the perception are with "to do better next time" and "improve in the future". While the perspective of statement 21 was a neutral result, the statement 24 had an average of agree. Therefore, it may be that while feedback helps guide students on how to improve specific concepts, it does not help with future assignments. The assessment literacy of students may impact these statements. It could be that while they see the general use of feedback given, the applications in other assignments are less clear.

#### *4.4.1.6 Feedback Quantity*

**Result:** Table 16 shows the results of a one-way between-groups analysis of variance that was conducted to explore the impact of year group on question 15 "there is hardly any feedback on my assignments when I get them back". Students were divided up by year group in which they took the questionnaire, the groups were split by the academic year as follows: 2016/2017, 2017/2018 and 2018/2019. There was a statistically significant difference at the $p<.05$ level in mean scores $p=0.000$. Despite having statistical significance, the actual mean difference was quite small. The effect size using ETA-Squared was 0.09. Post Hoc comparisons using the Tukey HSD test indicated that the mean score for 2016/2017 ($M=2.87$) was significantly different from 2017/2018 ($M=2.13$). There was also a statistical significance between the years 2017/2018 ($M=2.13$) and 2018/2019 ($M=2.81$). Means did not differ significantly between groups 2016/2017 and 2018/2019.

The large effect size of 0.09 suggests that the changes of students' perceptions within the disagree boundaries are important to note. It is clear from the results that students perceive there to be feedback of sufficient quantity. It is promising that students disagree with statement 15 and that the reported quantity of feedback was satisfactory. It is worth noting that for good practice to be applied quantity is not always better than quality. In fact, too much feedback can overwhelm the students (Price et al 2010). Additionally, the neutral results of statement 17 "I would learn more if I received more feedback" (2016/2017: 3.71, 2017/2018: 3.66, 2018/2019: 3.38) could suggest that either students have low confidence in the feedback and do not think more would be beneficial, or that the feedback provided is a sufficient quantity. To investigate these possibilities correlation tests are used to investigate the relationship between AEQ statements

**Multiple Comparisons**

Dependent Variable: There is hardly any feedback on my assignments when I get them back

Tukey HSD

| (I) Cohort Year | (J) Cohort Year | Mean Difference (I-J) | Std. Error | Sig. | 95% Confidence Interval Lower Bound | 95% Confidence Interval Upper Bound |
|---|---|---|---|---|---|---|
| Academic Year2016/2017Stage1 | Academic Year2017/2018Stage1 | .738* | .117 | .000 | .46 | 1.01 |
| | Academic Year2018/2019Stage1 | .057 | .105 | .850 | -.19 | .30 |
| Academic Year2017/2018Stage1 | Academic Year2016/2017Stage1 | -.738* | .117 | .000 | -1.01 | -.46 |
| | Academic Year2018/2019Stage1 | -.681* | .109 | .000 | -.94 | -.43 |
| Academic Year2018/2019Stage1 | Academic Year2016/2017Stage1 | -.057 | .105 | .850 | -.30 | .19 |
| | Academic Year2017/2018Stage1 | .681* | .109 | .000 | .43 | .94 |

*. The mean difference is significant at the 0.05 level.

*Table 16 Statement 15 There is hardly any feedback on my assignments when I get them back*

### *4.4.2 Correlations between AEQ statements*

Table 17 shows the results from correlation tests ran with statement 30 “I tend to only read the marks”. The aim of this was to identify if disengagement occurs when the students do not understand the feedback (Struyven et al., 2005). Using Pearson's product-moment correlation coefficient, we can investigate the link between statement 30 and other influencing factors. It is important to note that while correlation tests can inform us of a relationship between the two statements, it does not imply causation.

| **Correlation Number** | **Correlation Statements** | **Statement Number** | **Pearson's coefficient** |
|---|---|---|---|
| 1 | “I tend to only read the marks”/ “The feedback helps me to understand things better” | 30 and 20 | -.105 |
| 2 | “I tend to only read the marks” / “I don’t understand some of the feedback” | 30 and 23 | .179 |
| 3 | “I tend to only read the marks” / “Whatever feedback I get comes too late to be useful” | 30 and 18 | .225 |
| 4 | “I tend to only read the marks” / “I do not use the feedback for revising" | 30 and 29 | .223 |

*Table 17 Student Disengagement*

The small correlation between statement 23 and statement 30 (r=.179) suggests that while there is a small aspect of disengagement from the feedback there is a minimal relationship with its clarity. There is also a small relationship between disengagement (statement 30) and how students perceive the feedback can be used (statement 29) (r=.223). There is a stronger relationship between statements 18 and 30 which concern disengagement due to feedback timing (r=.225). The negative correlation between statements 20 and 30 (r=-.105) are expected as if students find the feedback to be beneficial there should be limited disengagement.

Table 18 shows the results of a Pearson's product-moment correlation coefficient, to investigate the impact of the quantity of feedback on students' perceptions of its usefulness.

| **Correlation Number** | **Correlation Statements** | **Statement Number** | **Pearson's coefficient** |
|---|---|---|---|
| 1 | "When I get things wrong or misunderstand them I don't receive much guidance on what to do about it" / "There is hardly any feedback on assignments when I get them back" | 16 and 15 | .464 |
| 2 | "When I get things wrong or misunderstand them I don't receive much guidance on what to do about it" / "On this course, I get plenty of feedback on how I am doing" | 16 and 13 | .366 |
| 3 | "When I get things wrong or misunderstand them I don't receive much guidance on what to do about it" / "I would learn more if I received more feedback" | 16 and 17 | .319 |

*Table 18 Quantity of feedback*

There was a medium correlation between statement 15 and statement 16 (r=.446). This relationship suggests that there is a connection between the quantity of feedback a student receives and their perception concerning its usefulness. However, the correlation of statements 13 and 16 (r=.366) suggest the opposite. While not as strong as a relationship as statement 15 and 16 there is still a positive correlation. We can also see in the correlation of statement 16 and 17 (r=.319) a similar strength relationship for students identifying that receiving more feedback would help them learn more. This is not a positive indication of student's perceptions of feedback. In Section 4.4.1 (page 79) two potential reasons for a neutral perception of students were discussed. The first was a lack of confidence in the feedback. The second possibility was that students perceive feedback to be of sufficient quality and quantity to clarify misunderstandings. The low correlation between statements 16 and 17 suggest that the former is more likely. An issue identified by the ANOVA tests and student interviews was that feedback focused too much on the negatives and what to do next time (page 66), as opposed to identifying what went well or why something was wrong. To investigate this further, correlation tests were used to identify a potential relationship between the usefulness of feedback and the mark driven nature of students.

| **Correlation number** | **Correlation Statements** | **Statement number** | **Pearson's coefficient** |
|---|---|---|---|
| 1 | When I get things wrong or misunderstand them I don't receive much guidance on what to do about it" / "The feedback helps me to understand things better" | 16/20 | .298 |
| 2 | "When I get things wrong or misunderstand them I don't receive much guidance on what to do about it" / "The feedback shows me how to do better next time" | 16/21 | .328 |
| 3 | "When I get things wrong or misunderstand them I don't receive much guidance on what to do about it" / "Once I have | 16/22 | -.287 |

| | | | |
|---|---|---|---|
| | read the feedback I understand why I got the mark I did" | | |

Table 19 Feedforward questions correlation results

Statement 16 has a positive correlation with statement 20 (r=2.98) and statement 21(r=.328) but a negative correlation with statement 22 (r=-.287). These correlations suggest that when students consider statement 16, they consider it from the perspective of what to do to gain the marks. Therefore, these correlations also give evidence of a strategic mindset being used by students, where the focus is on the marks and how to obtain them (Nilson, 2013).

Finally Table 20 shows a summary of the findings of students perceptions throughout chapter 4.

| Finding | Description |
|---|---|
| 1 | Students identified that staff use assessment to assess class progress and knowledge / general progress |
| 2 | There was a switch from heavy exam-based systems to coursework based system moving from further education to higher education. This may be a factor in the mark driven nature of students. |
| 3 | Assessors follow good practice by using a variety of assessment types. |
| 4 | It wasn't always clear from the student perspective why they were doing a specific assessment. |
| 5 | From the perspective of students feedback is used to give guidance on how to improve a student work. |
| 6 | Students identified that the "Insert Comments Here" auto filled by NESS reduced their confidence in the feedback |
| 7 | Feedback can focus too much on the negatives and what the student did wrong, but not necessarily why they were wrong. |
| 8 | As opposed to student's staff identified that they give large and details explanations of why assessments were set. |

| | |
|---|---|
| 9 | Perceptions from the AEQ align with the responses to the NSS question "Feedback on my work has helped me clarify things I did not understand" in that students did not report that they were satisfied on similar questions. |
| 10 | The responses to the AEQ suggest that students engage with the material and do not practice a surface learning approach. |
| 11 | Students are aware of what the end produce of an assessment should be but not necessarily the steps to get there. |
| 12 | There is evidence to support that the increasing specificity of the assessments improved student perceptions. |
| 13 | Student perceive that there to be a sufficient quality of feedback. There was a medium correlation between perceptions of the use of feedback and the amount of feedback received. |

*Table 20 Summary Of findings*

### *4.5 Ipsative Assessment and student perceptions*

This section will explore how Ipsative assessment may be used to reduce the impact of the issues identified in this chapter as well as how Ipsative assessment may be used to improve student perceptions as a whole.

A potential issue identified through the analysis of the AEQ was a lack of assessment literacy with students (page 89) in regard to understanding where their knowledge levels were and how to plan for future assessments. Ipsative assessment could help develop this skill as it has a significant focus on using assessment to help students understand their knowledge level, and to help guide them in future regulation of their learning by reflecting on their past experiences and how they have improved (Hughes, 2014).

By personalising student learning and identifying skills which can be developed across multiple assignments, Ipsative assessment could help to reduce the impact of the modularisation (page 69) and segregation of current higher education module designs (Ferrell, 2012). Ipsative assessment does not, however, help with the mindset of students viewing all assignments outside of programming as a waste of their time. Viewing other assignments that do not involve programming exercises as valid would require a change in mindset that the study of Computer Science involves more than the ability to program (Bailey and Stefaniak, 2001). As previously explored in Section 2.6, Ipsative assessment is a personal experience that investigates the improvement a student has made from their previous assignments. By linking this with the skills developed in the current assessment with a previous assessment, a student can begin to see the connections and possible skill transference to other modules and assessments. This may help students see value in assessments beyond that of programming (an issue identified in Section 4.3 page 65). For example, essay writing teaches a student how to research, how to find information that they do not know and how to use academic resources effectively. The development of these 'soft' skills can help students when programming, by teaching students how to find solutions to problems and how to investigate coding concepts to help with their current projects effectively. They are all skills that need to be developed for computer science careers and require more experiences than code writing.

The responses to the interview questions regarding the experiences with feedback (specifically how students use feedback) show the development of the Ipsative mindset and a self-regulated form of learning (page 67). The ability to identify areas of improvement and plan goals to achieve them is an important aspect of facilitating learning (Housand & Reis, 2008; Eugenia, 2018; Tai and Adachi, 2019). It allows students to identify gaps in their knowledge state and work towards learning the misunderstood material, therefore increasing their self-regulation and self-reflection skills. Ipsative assessment is a beneficial way for students to further improve these skills. By developing the skills that allow a student to self-assess and self-regulate their learning, Ipsative assessment will teach students how to see areas of improvement from past work, and how the skills that the student did well in can be applied in other areas (Hughes, 2014).

As Ipsative feedback prioritises knowledge (or learning) gain from the student’s perspective, there are two main benefits to using an Ipsative system when considering the use of feedback. The first benefit is the reflective aspects for a student, by focusing on specific improvements that they have made rather than the criteria they have not been successful at, it can motivate students to learn more. Secondly the aspects of Ipsative feedback which feedforward give specific targeted, personalised goals to direct students learning. These goals will then allow students to identify what needs to be improved and enable them to self-regulate their learning (Hughes, 2017, pp9). From this, students across the course can be challenged without feeling that any of the material is inaccessible. While the knowledge gap for the same practice may be too big for some students, the personalised learning and goal setting can be engineered to give the same learning distance for each student and therefore similar amounts of work (Hughes, 2017, pp9).

An issue identified within the analysis of the AEQ was that it was not always clear from the assessment specification and the feedback received where the marks for the assessment came from. The practice of mapping the specification to the comments that students receive from feedback can be assisted by Ipsative assessment. Ipsative assessment would assist with this practice by guiding students in identifying specific areas of weakness from this assessment and apply that feedback at to future specifications.  This makes the self-referential aspect of Ipsative feedback more plausible and can help guide the students in their future learning (Hughes, 2017, pp9).

From the analysis of the questions regarding assessment and feedback, a source of student dissatisfaction maybe that well-performing students do not receive as much guidance or ways to improve as 'at risk 'students. As Ipsative assessment is a personalised experience, it can help ameliorate these issues. Ipsative assessment focuses less on the meeting of criteria and instead places focus on the improvements a student makes and therefore there will always be feedback on how the student could improve (Hughes, 2014). A student who receives a mark of 95% still has room for improvement, if less so than a student who achieved a mark of 50%. By placing more emphasis on the skills and techniques a student needs to develop, Ipsative assessment could help move focus away from the mark driven nature often seen in students and instead place the focus on how to improve and develop further in that specific area and shifting to a deep learning mindset (Hughes, 2017, pp2).

The example of a tailoring feedback to help students understand a specific aspect of an assignment can be seen in Section 4.4.1.1 (page 82). By including a specific function that a student has to use in the 2017 specification, such as array, results in the feedback being more specific and directed for each student. This will enable an Ipsative feedback form which allows these mistakes or misunderstandings to be turned into specific goals, where the students can tailor their learning to resolve a particular issue. When using Ipsative feedback, the feedback will become specific to the students, with direct guidance regarding what they misunderstood and how to correct that specific instance (Hughes, 2017).

### *4.6 Summary*

This chapter explored the qualitative and quantitative methods used to answer RQ2 and RQ3 as defined in chapter one. Interviews with students at Newcastle University were conducted to explore the perceptions of students regarding assessment and feedback practices within higher education. Staff members were also interviewed to explore the differences in perceptions and to identify how current practices compare with good practice identified by the literature. To further explore the perceptions of feedback, the Assessment Experience Questionnaire was given to students.

The responses to the interviews show that students have a strong theoretical understanding of assessment literacy and are aware of how assessment and feedback should be used. However, this understanding appears to be rarely put into practice. This lack of practice can be seen when exploring how students handle feedback. While students identified that feedback should help explore future learning, the use of the feedback was focused on mark gain/mark loss with mention of comparisons to other students.

Staff also reported that there was a lack of willingness to engage with the material beyond what was the necessary minimum. This was further seen with the lack of student engagement outside of programming assessments. While staff reported that students would not engage with material outside that of programming, students reported that the purpose of those assessments were not always clear. This was also seen within the correlation statements where there was a negative correlation between feedback clarifying misunderstandings and identifying where marks were lost. Similarly, to developing a student's assessment literacy, Ipsative assessment will develop a students' ability to identify how skills in different areas can be connected. This occurs by giving specific targeted tasks on how to improve as opposed to generalised feedback on that specific assessment.

The perceptions of the clarity of assessment did not significantly change throughout the three years of this study. Even when the specificity of the assignments was increased, the general perception was still negative if slightly improved. Statement 16 of the AEQ showed that perceptions of feedback changed to be more positive in the years where there was more specificity to the assignments. In general, students had a positive perspective on the use of feedback to guide them through misunderstandings. The responses to the AEQ and student interviews suggest that while feedback helps students identify mistakes and what not to do in the future, there is a limited exploration for why something is wrong, or what exactly the student should do to rectify this mistake. This once again shows a desire for feedback to identify exactly where marks were lost within the work. An Ipsative approach would inform the students where they went wrong and how to fix it by focusing on areas of improvement and not how it relates to the marks.

The results of the AEQ found that as a whole, pre-and post-moderation have had little impact on the perceived quality of feedback to the students. The ANOVA results have shown that student perceptions of feedback have not drastically changed with the introduction of new cohorts. It is worth considering that students have not seen a different method of feedback, therefore they are not reporting on improvements, but their perceptions of the feedback as it. Furthermore, pre- and post-moderation were not new techniques introduced during this research, but techniques that were developed and built upon. Therefore, there would not have been drastic changes made to the feedback given to students. The effect of pre-and post-moderation will be further explored in chapter five.

# Chapter 5. Quantitative analysis of historical data

### *5.1 Introduction*

This chapter explores the analysis of historical feedback given to students on the CSC1021 module 'Programming 1'. This analysis aims to investigate if the perceived problems identified by students in chapter four are present within historical student feedback. The exploration was achieved using a qualitative experiment applying the feedback criteria developed in Section 2.7 to the historical data.

### *5.2 Collected Data*

942 sample academic records of students who took the module CSC1021' Programming 1' and were studying for a Computer Science Degree, were reviewed from the academic years 2012/2013, 2013/2014, 2014/2015, 2015/2016 and 2016/2017. CSC1021 is a first-year module that is offered to both Computer Science Students and Engineering Students. As this research focused on the student experience in Computer Science, the feedback given to the Engineering students were not considered. It is the same module explored with participants in Chapter 4 using the AEQ questionnaire and interviews. A detailed explanation of CSC1021 can be seen in Section 3.4. This module has a pass mark of 40% with a requirement that students pass both the coursework (two practical assessments) and exam components of the module. Table 21 shows the breakdown of the 942 students into their year group, of the 942 students sampled, 846 passed the course.

| | | Frequency |
|---|---|---|
| Valid | 2012/2013 | 115 |
| | 2013/2014 | 181 |
| | 2014/2015 | 191 |
| | 2015/2016 | 216 |
| | 2016/2017 | 239 |
| | Total | 942 |

*Table 21 Number of students per cohort*

For this review, only feedback from assignment two was analysed. In the years 2012 and 2013 feedback was given to students via paper-based feedback sheets (see Appendix D) for assignment one, therefore this data was unavailable. Appendix A and Appendix E show the module outline and the specification for coursework two respectively. The aim of coursework two is to create a pizza delivery class or hotel booking system using the aims and skills identified in the module outline. These include but are not limited to: programming fundamentals, problem-solving skills via the use of Java code development, program design, modularisation through the development of specific classes as outlined within the coursework specification, and finally, the design, running and documentation of tests for their programs.

For the purposes of the qualitative review, final module marks were investigated on CSC1021 from 2012 to 2016. A module mark is a calculated mark which uses all assessment marks, including coursework and exams, as well as any discretionary methods, such as personal circumstances or compensation.  Yorke explores that in each subject there will be a range of marks specific to that discipline. Mathematics and Science-based subjects have a wider range of mark distributions than those of humanity subjects such as English and History. This often occurs due to the nature of the topics as Mathematics and Science based subjects have narrower definitions of what is classed as a correct or incorrect answer. Where there is little interpretation of how a scientific method or formular could be used, subjects such as English allow for a much broader discussion on the part of the student. The interpretation of an author's or an artist's work has a much broader spectrum than the solutions of science-based questions (Yorke, 2000). A histogram of all final module grades throughout the 5 years of the data collected can be seen in Figure 12, Figure 13 shows the distribution of the grades by individual year. Within all Histograms a similar pattern occurs, the grades within the module have a double peak in the normality plots. The double peak can be seen in both the individual years and the collated grades as a whole. The first peak happens at around a mark of 30% to 40% and the second peak at 70% to 80%, as these are student grades this is an expected result as the distribution will not be normal (Arthurs, 2019). It also gives evidence to the double peak of material mastered within Bloom's taxonomy (Scott, 2003). As mentioned in chapter two, it is often the case that assessors can create assessments which test the novice material along with the expert material, but often lack assessment of the material in the intermediate range. The effect of this can be seen in the double bump of student grades, where there is a peak for material that covers the novice range with an additional peak at the 'expert range.' However, in the 2014 and 2015 histogram the double peak is less pronounced with a much larger cluster in the middle grade range.

| Year | Mean | N | Std. Deviation |
|---|---|---|---|
| 2012 | 62.72 | 115 | 19.317 |
| 2013 | 64.65 | 181 | 17.488 |
| 2014 | 68.41 | 191 | 17.057 |
| 2015 | 70.56 | 216 | 15.058 |
| 2016 | 73.89 | 239 | 14.559 |
| Total | 68.87 | 942 | 16.823 |

Table 22 Mean Student Grades

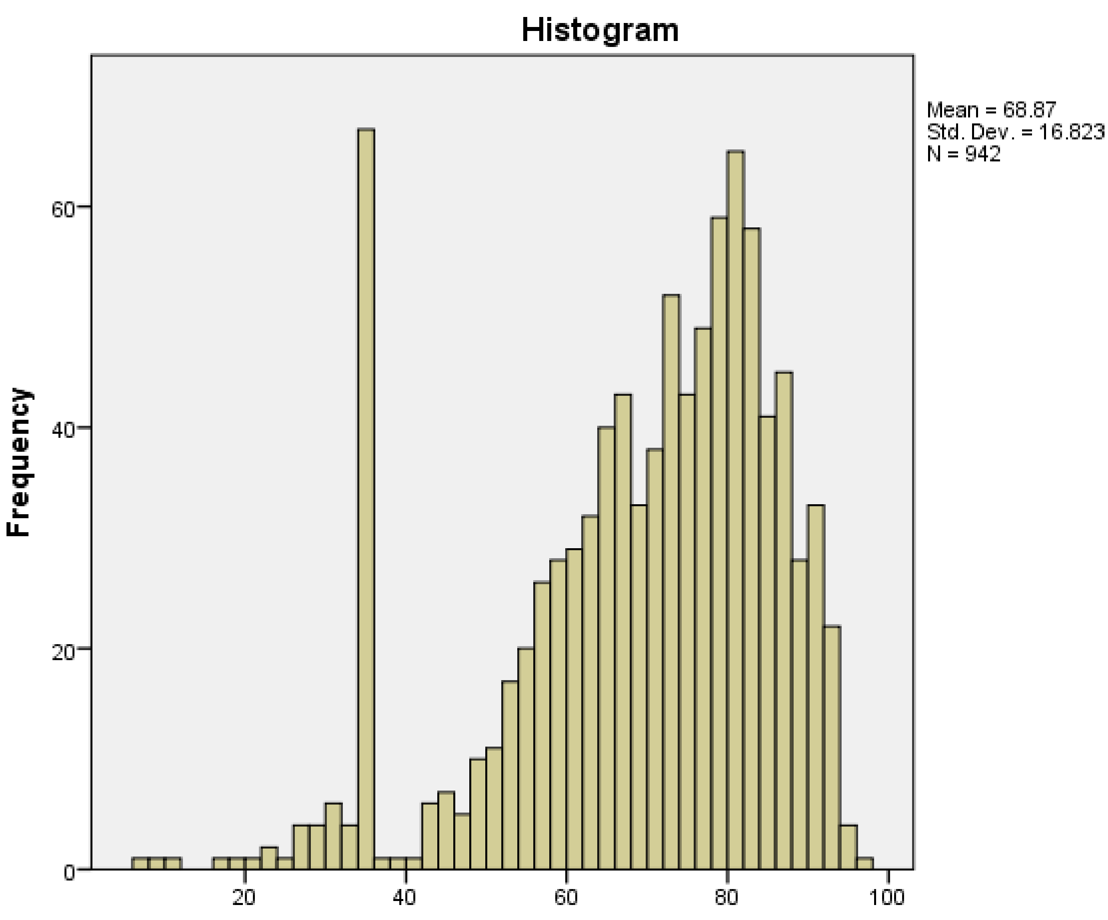


Figure 12 Histogram of student grades for 2012 - 2016

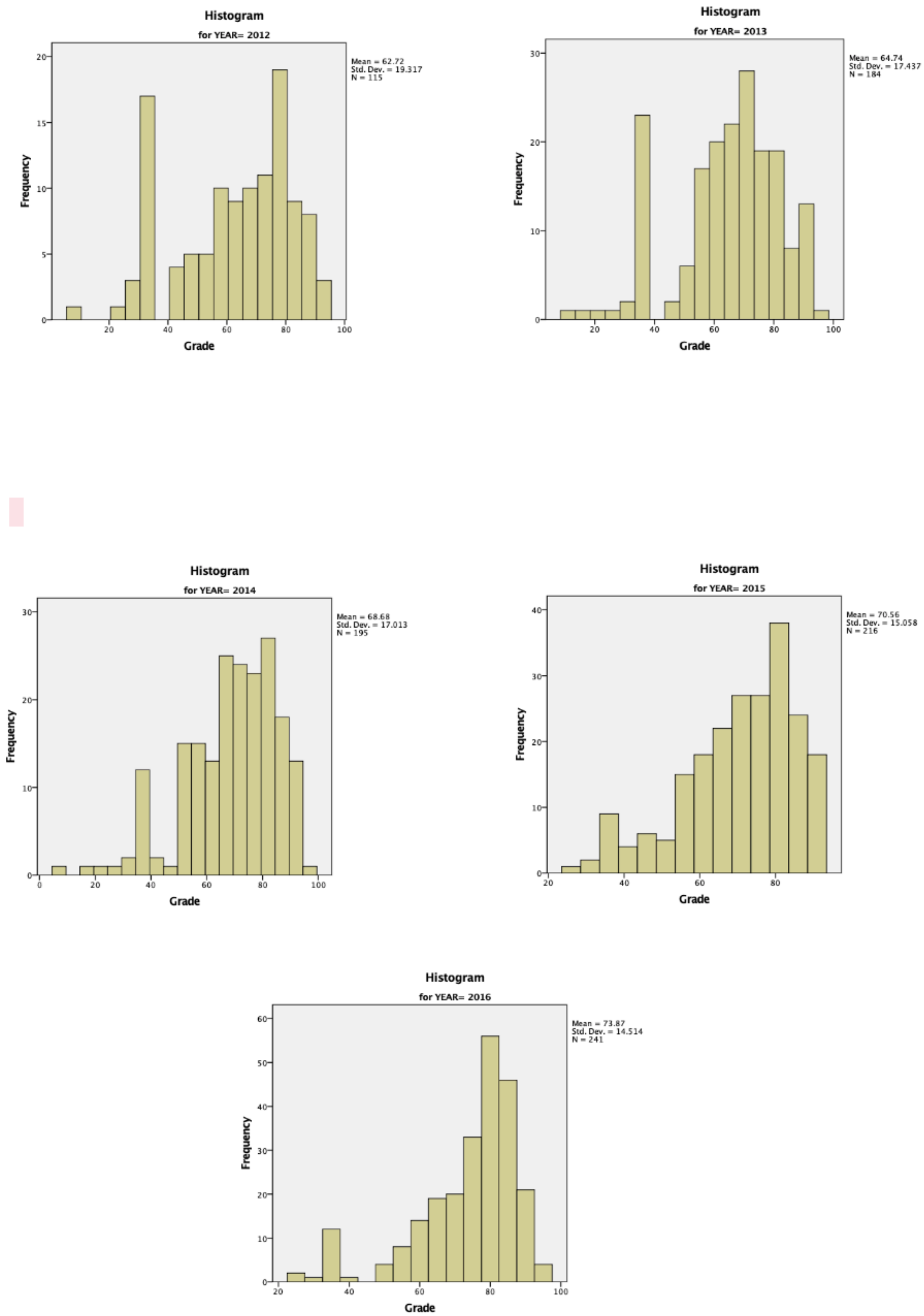


*Figure 13 Histograms of student grades per year*

Table 2y shows that the mean grade for each of the cohorts increased each year that the data was analysed (2013: 1.93% ,2014: 3.76%, 2015: 2.15%, 2016: 3.33%). We cannot directly compare these years with each other as multiple factors can impact the grade a student receives. For example, due to the increasing demand of people with computing skills, there has been an increase of teaching computing science within primary and secondary education (The Royal Society, 2017). Due to this, students could be entering higher education with previous experience within computer science and therefore have a better understanding than the years previously, which may have an impact on student grades. It is however, interesting to first investigate a potential correlation between how many feedback criteria a piece of feedback meets and then the final grade a student receives. It is also important to investigate if the usage of pre- and post-moderation processes used to improve the consistency and quality of student feedback has had an impact on student grades. Following this, a multiple regression test was used to investigate what could explain the variance within these results and if feedback was a contributing factor. A second factor to consider is a negative correlation between the number of feedback criteria a submitted piece of work met and the associated students' grade. A weaker assignment submitted by a student, would expect more feedback than that of a stronger submission. For example, a student who receives 40% on an assignment would expect more feedback identifying mistakes, explaining misunderstood concepts, and informing them where they could improve. An assignment that received 70% or above, while still expecting feedback on ways in which they could improve, would expect less feedback on how to improve than that of a 40% grade submission.

The first stage of the investigation was to look at the normality of the student's grades. Table 23 shows that this data is not normal, there are two reasons for this. The first reason is the double peaks. It is the nature of higher education that there will be a cluster of students around the failing grade. It is rare for a student to get no marks, but students still do fail and a second peak will occur in the passing students. The second reason draws more from the nature of statistics than that of the subject of those statistics. Pallant states that for data over 200 entry points, it is expected for the assumption of normality to be violated. This can also be seen when removing the students who fail, thus removing the double peak, the normality of the results is still not normal (Pallant, 2010, pp63).

If the results for the tests of normality show that the significance 'p' is less than 0.005, the results violate the assumptions of normality. In the case of student grades, when considering all cohorts $p = .000$, therefore normality is violated. In this experiment, the number of participants was larger than 850, as can be seen in Table 22, due to this, it was expected that the data would not be normal.

Table 23 shows the results of the Kolmogorov and Shapiro tests, which can also be examined for the normality of the results, if these values are greater than 0.5 the data is normal. In this case, the values are both .000, which once again highlights a violation of the assumptions of normality (Pallant, 2010). As this is an expected result, the study continued with the selected statistical tests. It is however important to note that without the extenuating conditions such as the educational data and data with over 200 participants, if normality is violated the statistical tests should not be used.

**Tests of Normality**

| | Kolmogorov-Smirnov[a] | | | Shapiro-Wilk | | |
|---|---|---|---|---|---|---|
| | Statistic | df | Sig. | Statistic | df | Sig. |
| Grade | .083 | 846 | .000 | .973 | 846 | .000 |

a. Lilliefors Significance Correction

*Table 23 Normality tests for student grades*

### *5.3 Feedback Analysis*

The first step of the investigation was to explore descriptive statistics. The descriptive statistics were used to answer how much of the written feedback given to students met the feedback criteria. The second step was to conduct a further breakdown of feedback criteria score per year to investigate if pre- and post-moderation had an impact on feedback quality and consistency, the following hypothesis were defined to investigate this.

(H0) Null Hypothesis: Pre- and post-moderation had no effect on the quality and consistency of the feedback given to students.

(H1) Alternate Hypothesis: Pre- and post-moderation had a positive impact on feedback given to students giving more structure and detail.

| | FC1 | FC2 | FC2a | FC3 | FC4 | FC5 | FC6 | FC7 |
|---|---|---|---|---|---|---|---|---|
| 2012 | 66 | 75 | 83 | 73 | 62 | 60 | 15 | 28 |
| 2013 | 70 | 93 | 86 | 74 | 67 | 61 | 40 | 38 |
| 2014 | 73 | 76 | 85 | 82 | 78 | 79 | 63 | 62 |
| 2015 | 70 | 75 | 92 | 93 | 79 | 85 | 17 | 47 |
| 2016 | 78 | 97 | 99 | 100 | 77 | 82 | 15 | 38 |

*Table 24 Feedback Criteria Met Percentages*

The feedback criteria stated in Table 24 (introduced in Chapter 2 Section 2.7) were as follows

FC1 Must give ways to improve the work, with recommendable transferable skills.

FC2 Does not comment on the students themselves but instead comments purely on the work.

FC2a The comments should be non-judgemental.

FC3 Refers to the work itself with given examples.

FC4 Must clarify what good performance is and how the student can reach it.

FC5 Must recommend a manageable workload.

FC6 Feedback given to students must be clear.

FC7 When the student gets something wrong, the feedback informs the student why this was wrong.

The results of the frequency analysis show that across each year, the feedback criteria with the least 'present' mark was: FC6, (Feedback given to students must be clear). The feedback criteria that was most present was FC3 (Refers to the work with given examples). Descriptive statistics show that feedback often tells the students what they did wrong and what is classed as good practice. This can be seen in the high response rate to FC1, FC3 and FC4. By identifying what is classed as good practice and how the submission does not meet this, it informs the students where they can improve. What the feedback does not do is explain why their solution was wrong. From student interviews and the responses to the AEQ, it was identified that while feedback informed students where they had made mistakes, and what the solution should look like, the feedback did not clarify student misunderstandings. Ott and colleagues show that in addition to informing students of where they need to go and how to get there, it is also important to give students clarification on how they are currently performing, this can involve why what they are doing is wrong (Ott et al., 2016). Furthermore, student dissatisfaction can be caused by not giving enough clarity on why the work was incorrect. Good clarity can be achieved by informing students as to why mistakes were made, so they know how to avoid making those mistakes in the future.

From the frequency analysis, we can see that within each year, historical feedback does help give direction to future improvements and identify where mistakes were made. What the feedback was sometimes lacking, is an explanation of why what the student did was wrong, this can be seen when exploring the results of FC7 (when a student gets something wrong the feedback informs the user why this was wrong). Table 24 shows that in the lowest result which occurs in 2012, only 28% of the feedback analysed explained to students why what they did was incorrect. The year in which feedback met FC7 the most was 2014 with 62% of the feedback analysed achieving this feedback criterion. It is likely from these results that pre- and post-moderation had the most impact on this area. However, the amount of feedback that meets this criteria point is consistently seen to be below 50% in Table 24 and Figure 14.

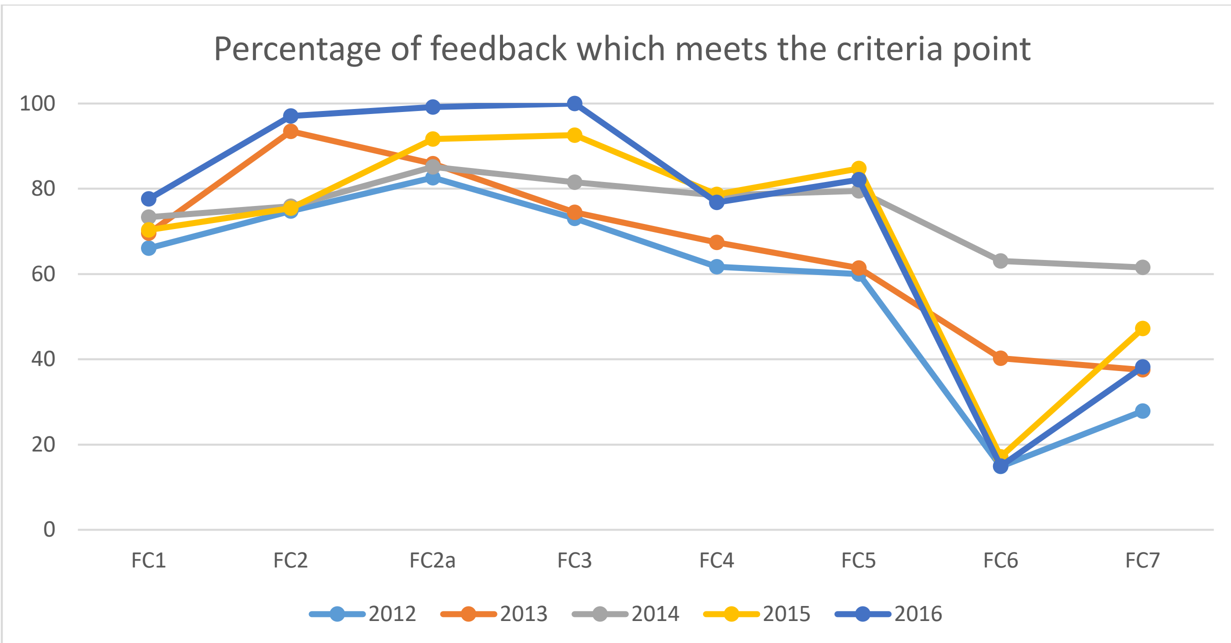


*Figure 14 Percentage of feedback which meets the criteria point*

Pre- and Post-Moderation was a process that was introduced in 2014 (see Chapter 4 for an in-depth discussion). The analysis shows that after the introduction of pre- and post-moderation, the feedback became more structured. Feedback began to more consistently follow good practice as recommended in the literature. This included using feedback to facilitate learning and to feed forward for future assignments therefore helping students to identify what good practice should be. When feedback is used to help students understand what current standard their work is at and how it can progress, it helps to facilitate learning. Feedback also helps students gain a deep understanding of the concepts that they are learning, helping them to operate at the higher level of Bloom's Taxonomy (Ott el at, 2016; Bloom 1956). Section 2.4 discussed how for assessment to be fair and valid, it must be criteria referenced. Section 4.4 also explored that the more specific an assessments specification is the better the perception of the students (Hughes, 2011). This identified practice also allows for feedback to be more specific and targeted, thus keeping assessment and feedback fair and valid and reducing student dissatisfaction.

Pre- and post-moderation were demonstrated to ensure greater consistency of feedback quality with increasing marker numbers (due to the increasing student numbers). This can be seen by the increase of feedback that met criteria 3, 4, 5, 6 and 7 in 2014. FC6 and FC7 once again decline after 2014, however the initial substantial rise is important to note.

Within computer science, students must have a strong grasp on the concepts underpinning the language they are using, as opposed to the language-specific syntax. As computing science students are required to use multiple languages within their course and career it is important that feedback identifies the general, transferable skills that the student can improve upon, in addition to the instance that they are currently using. It is therefore promising that FC1 and FC4 have high values.

The criterion that was met the least was FC6, the feedback given to students must be clear. There were multiple factors which made the feedback 'Unclear'. The first was mistakes in spelling and grammar within the feedback. A second factor to be considered was the display of the feedback. Figure 15 shows an example of errors which can be introduced into the feedback depending on the view. For example, in Figure 15, the ASCII codes ' appear in the NESS system. These ASCII issues were commonly seen within the feedback given to students and were a large factor to making the feedback unclear.

Could have 'addBed' in Room, similar to 'addRoom' in Hotel

Good use of private and getters/setters

Code formatting a bit inconsistent, made readability a little harder

Configuration is well implemented

*Figure 15 Example of Symbol Errors*

Furthermore, in some instances of feedback, the introduction of these codes can increase the difficulty of reading the feedback and in some cases, made feedback illegible. This issue became more frequent during the latter years of the data, as can be seen in Table 24 and Figure 14 when the 'feedback given to students must be clear' section has a "present" rate of 15% in 2016.

Another issue can be seen in Figure 15, which shows a contradictory statement. Minimal feedback is given to the students work, however, this is followed by a "No work Submitted" statement.

> Your style is reasonable. You have used privacy correctly (although not consistently). You have attempted get/set methods (again not consistently). Identation and variable names are reasonable. Commenting is minimal.
>
> A reasonable introduction.
>
> Your tests are reasonable, although could be written much more succiently.
>
> No work submittedÂ

*Figure 16 No work submitted example*

Figure 16 shows an example of an auto-population within the feedback system NESS "write your comments here". Mistakes such as those that can be seen in Figure 15 and Figure 16 show a carelessness on the part of the marker that can lead to students having minimal confidence in the marking and feedback given, thus causing disengagement and dissatisfaction.

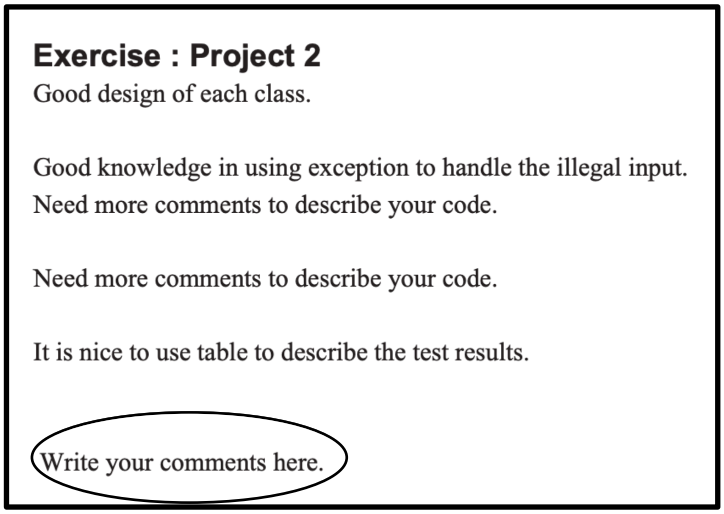


*Figure 17 Feedback auto population*

From the initial analysis using descriptive statistics, we can see that pre and post moderation may have had an impact on the consistency and standards of the feedback given to students as there was a significant change the amount of feedback criteria met when these processes were introduced. .

### *5.4 Correlation of feedback criteria and student grades*

Wilson suggests that higher education can become grade driven for students (Wilson 2006). The grade driven nature of students can often be in the form of an expected grade boundary breakdown within the feedback that they receive, this may occur when a student's grade is close to a grade boundary. The grade driven nature may also be a by-product of the criteria driven aspects of the assessment specifications. Therefore, suggestions could be made that good feedback that identifies which aspect of the assessment criteria the student missed, would positively impact students and help lead to a better understanding of the grade that they received. This section explores the potential impact of feedback on a student's final module mark.

The following statistical tests investigate if the feedback quality has a relationship with the returned module mark for CSC1021 students.

H0 Null Hypothesis: The number of feedback criteria met does not influence final student mark.

H1 Alternate Hypothesis: The more feedback criteria met, the lower the student's final grade.

To assess the impact of feedback on the final grade a correlation between the number of feedback criteria met in students' coursework feedback and the final returned grade for the module CSC1021 is calculated, to do this correlation statistics are used.

Table 25 shows the relationship between the total amount of feedback criteria met and the final student grade using the Pearson Product-Moment Correlation Coefficient (Section 3.3.2, page 47). There was a very small correlation between the two variables r=0.033, n=951, p>0.05. This small correlation is not too surprising. It was often the case when students got a high grade, the feedback met fewer of the criteria. One case showed that when considering a student whose final grade was 92, their feedback met 5 of the 8 criteria. The three criteria not met being FC4 (Must clarify what good performance is), FC5 (Must recommend a manageable workload), and FC7 (When the student gets something wrong, the feedback informs the student why this was wrong), all of which focus on the student improving in the future.

**Correlations**

| | | Total FC met | Grade |
|---|---|---|---|
| Total FC met | Pearson Correlation | 1 | .033 |
| | Sig. (2-tailed) | | .307 |
| | N | 951 | 951 |
| Grade | Pearson Correlation | .033 | 1 |
| | Sig. (2-tailed) | .307 | |
| | N | 951 | 951 |

*Table 25 Feedback criteria and grade correlations*

When investigated individually only one feedback criterion, FC3 (refers to the work itself with given examples), had a correlation with the final student grade. Table 26 shows the small correlation of r=.123, n=951 and p<0.005. This is also to be expected, as when referring exactly to the work, there is something clear for students to work towards. As Ipsative assessment requires a personal aspect to the feedback and often encourages small steps in their progress, it may be a useful tool to help encourage this. This focus on small improvable steps also encourages students into the 'mover' mindset (Robins et al., 2005) as opposed to seeing mistakes as tasks that they cannot do. Once this initial starting point has been made, the students have targets which they can tailor their learning to, thus giving a specific margin for improvement (Hughes, 2014). A positive correlation between FC3 and final module grade returned may be due to feedback aiming to help correct specific misunderstandings.

A further investigation of correlations took place by splitting the dataset up into the years that the student took the module. There was no significant correlation between feedback quality and final grade in the academic years 2013 and 2014. There was however a small correlation with significance in 2012 for the FC4 (Must clarify what good performance is and how the student can reach it) with final student grade (r=.2). There was also a significant negative correlation within 2015 for FC5 (Should recommend a manageable workload) (r =-.144) with student grade. Finally, there was significant negative correlation in 2016 for FC6 (should clarify what good performance is) (r=-.03). From this, it can be concluded that there is no direct link between the number of feedback criteria met and the final module mark of the students. As after the improvement of feedback quality through the use of pre and post moderation, there are no significant correlations with a Pearson product large enough to signify a relationship.

**Correlations**

| | | Grade | Refers to the work itself with given examples |
|---|---|---|---|
| Grade | Pearson Correlation | 1 | .123** |
| | Sig. (2-tailed) | | .000 |
| | N | 951 | 951 |
| Refers to the work itself with given examples | Pearson Correlation | .123** | 1 |
| | Sig. (2-tailed) | .000 | |
| | N | 951 | 951 |

**. Correlation is significant at the 0.01 level (2-tailed).

*Table 26 Grade and Feedback Criteria 3 correlation*

To further explore these findings and investigate any potential impact from the individual criteria, multiple regression was used to investigate any potential link that correlations may have missed.

### *5.5 Multiple Regression*

Multiple regression is a technique that can be used to explore the ability of a set of variables to predict a result. Standard multiple regression is used within this section. Having explored some of the impacts of the relationships between the feedback criteria met and the final grade of the student through correlations, we can hypothesise that there are  no (or a very minimal) direct links between the two. To investigate the relationship further to confirm these findings, multiple regression techniques were carried out using the returned module mark as the dependent variable, and the feedback criteria as the independent variable. The aim was to investigate a potential variance in the marks based upon the feedback criteria.

| Model | | Collinearity Statistics | |
|---|---|---|---|
| | | Tolerance | VIF |
| 1 | (Constant) | | |
| | Must give ways to improve the work, with recommendable transferable skills | .618 | 1.617 |
| | Does not comment on the students themselves but instead comments purely on the work. | .770 | 1.298 |
| | The comments should be non-judgemental | .730 | 1.370 |
| | Refers to the work itself with given examples | .715 | 1.398 |
| | Should clarify what good performance is and how the student can reach it. | .558 | 1.792 |
| | Should recommend a manageable workload | .502 | 1.991 |
| | Feedback given to students must be clear. Feedback should be clear to students, there should be no spelling or grammar mistakes and the written language should be clear to both native and non-native speakers. | .960 | 1.041 |
| | When the student gets something wrong the feedback informs the user why this was wrong | .742 | 1.348 |

*Table 27 Multiple Regression Results*

The first step is to identify if there is an issue with multicollinearity, which is the relationship between two variables. The way to do this is to use the tolerance and VIE columns of the SPSS output, as seen in Table 27. The traditional cut off points to identify multicollinearity is as follows "tolerance < .10" or "VIE>10" (Pallant, 2010, pp158). Table 28 shows that the tolerance and VIE is above .10 and 10 respectively, therefore we can evaluate the model to see if there is any effect on the variance of grades. To do this we look at the Model Summary table presented by SPSS.

**Model Summary** [b]

| Model | R | R Square | Adjusted R Square | Std. Error of the Estimate |
|---|---|---|---|---|
| 1 | .182[a] | .033 | .025 | 16.578 |

*Table 28 Multiple Regression R squared*

Table 28, which shows the SPSS Model Summary gives us the value of $R^2$, which tells us the variance in the grades based upon the model. For this model we can see the value is very small. This can be turned into a percentage (multiply by 100) which gives us 3.3%. These tests confirm the Null hypothesis of the experiment. That while there is a small correlation between the amount of feedback a student receives and their final grade, the variance was too small to conclude a positive or negative relationship. The small relationship is, however, a positive result, all students should have feedback which meets some criteria. It is rare for a student to have a perfect solution or assignment with no area for recommended improvement. Therefore, it should be possible for feedback to identify mistakes, give correctional guidance and inform the student why their solution may not have been applicable in that situation.

### *5.6 Comparing student perceptions with the feedback criteria.*

The assessment experience questionnaire (AEQ) (Gibbs et al. 2003) discussed in chapter 4 investigated student perceptions of assessment and feedback. For the years the AEQ was distributed there is a one-year overlap with the feedback criteria case study. Using the 2016 data, this section will compare the perceptions of students with the contents of the feedback for that year of overlap.

The AEQ identified the perceptions of students regarding assessment and feedback practices at Newcastle University. The key points identified by the results in table 8 Section 4.4 were as follows:

- Statement 22 showed that feedback did not have guidance on where marks were gained or lost from the perspective of the students.
- Statement 23 neutral response suggests that the feedback given to students is sometimes unclear.
- Student interview responses and the response to statement 21 and statement 27 suggest that while feedback helps guide students with mistake corrections, it does so by identifying negatives, rather than explaining how to improve.

Figure 18 shows the percentage of feedback that meets the eight feedback criteria, a full table expansion of these results can be seen in Appendix I.

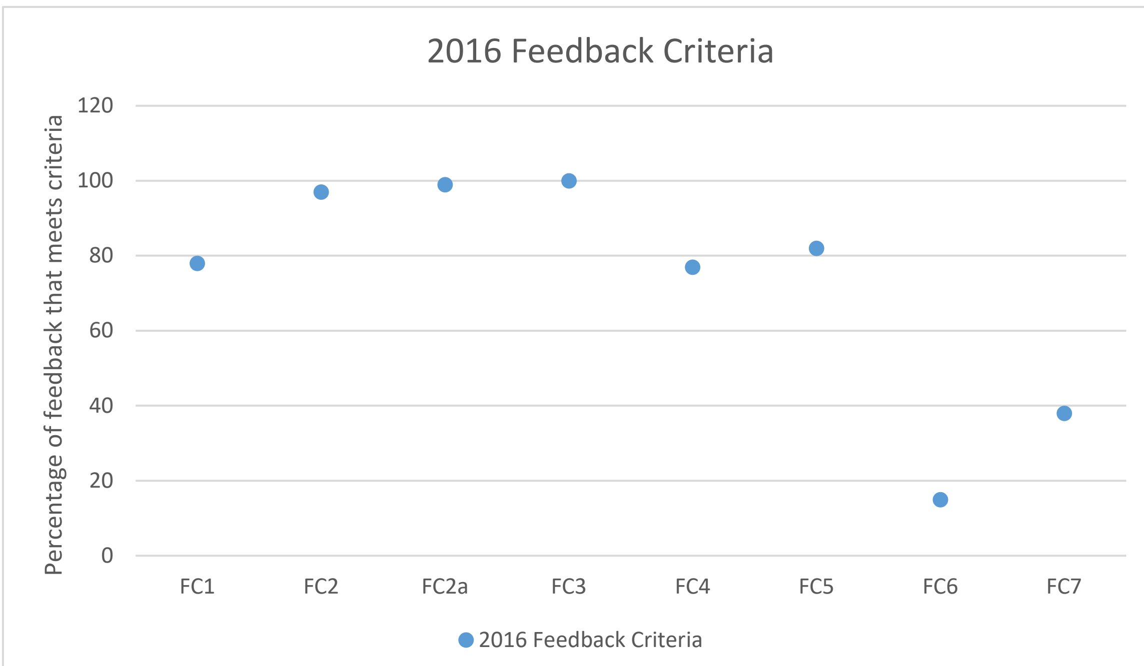


*Figure 18 Percentage of feedback that meets the feedback criteria, 2016 Cohort*

When comparing the reported student perceptions and the percentage of feedback that met the feedback criteria, it can be seen that the feedback does, in fact, have issues with clarity. Figure 19 shows that in 2016 only 15% of the feedback analysed met FC6 (feedback given to students must be clear). It is interesting to note that the system NESS has different display settings dependant on the view that the user has. It was found that the viewpoint students have produces a slightly different output to the one provided to the academic user. Therefore, due to clarity of feedback also taking into consideration the display of the system, these results may vary depending upon the output used, however, this cannot be concluded for certain.

A further aspect to explore is the concept of feedback justifying marks. As students can be mark driven regarding their feedback, statement 22 ("Once I have read the feedback I understand why I got the marks that I did") would be an important factor of student satisfaction (Smith, 2003; Ferrell 2012). However, the results from this statement was a *disagree* response from the students for the 2016 cohort, M = 2.91. While feedback should not be used for mark justification, the student should be able to infer where they have lost/gained marks if the feedback is of sufficient quality, this is often seen in formative assessment where mentions of marks are minimal (QAA, 2018). The feedback criteria analysis shows that the feedback given to students frequently identifies what assessment criteria the students did not complete and what a correct instance of the solution would look like. This can be seen from the responses to FC1: 76%, FC4: 99.6% and FC5: 75%. This, therefore, if not directly, will tell the student where the work was not up to the expected standards to receive the marks and how the student could have done this. Not being able to identify this is an issue regarding a student's assessment literacy. Ipsative assessment can help a student to extrapolate where marks were gained or lost based upon the feedback that the student has been given. This can be achieved by clearly identifying skills that students have not completed within the assignment, and therefore allow students to develop skills beyond that of coding. In chapter 4, staff interviews stated that students often could not see areas of improvements beyond that of coding mistakes. Ipsative assessment can help change student mindsets to focus on the skills that they need to develop as opposed to the marks that they correlate too, moving the focus away from the more competitive nature of traditional summative assessments (Hughes, 2017). Ipsative assessment achieves this by removing the high stakes nature of summative assessment and allows students to focus on their development as opposed to meeting the standards. While an assessment piece will be needed to prove that students have at some point met the external standards, this can often take place in the final stages of the block of learning (such as a dissertation) so that throughout the study students can focus on how their learning is developing. As mentioned in Section 2.6 (page 36) this type of learning happens in doctoral studies, where students spend the majority of their study time learning how to research before finally submitting a thesis and completing a viva.

### *5.7 Inter-rater reliability*

To conclude the feedback analysis, Cohen's Kappa (Section 3.3.4, Page 48) was calculated on a subset of data. A colleague reviewed 10% of the 2012 cohorts feedback using the feedback criteria. The Cohen's value was k=.446. A value of .446 is a moderate agreement between the researcher and the checker, meaning that there is still some personal judgement about what is considered good feedback and what meets the criteria. Therefore, more development of the specificity of the feedback criteria is needed. Appendix J shows an example of a Cohen's Kappa analysis between the researcher and the checker. The Appendix shows both an example of feedback in which the researchers and checkers analysis match, and one where it does not.

### *5.8 Summary*

| Finding | Summary |
|---|---|
| 1 | The double peak in the histogram of student grades (at 30/40% and 70/80%) give evidence to the "double peak" described by Scott (Scott, 2003). |
| 2 | The feedback criteria that was met the least was FC6 (Feedback given to students must be clear.) |
| 3 | The feedback criteria that was met the most was FC3 (Refers to the work itself with given examples). |
| 4 | Feedback informs the student what they did incorrectly and what a correct solution would look like. |
| 5 | Feedback (in general) did not explain why a solution was incorrect. |
| 6 | A contributor to a lack of clarity within the feedback are the errors that are introduced by the feedback system. These errors include, ASCII codes within the feedback, "Write comments here" auto population and "No submission" comments with associated feedback. |
| 7 | There was not a significant correlation between the amount of feedback criteria met and the students final grade. |
| 8 | There was a small correlation (r=.123) between the final grade of a student and if the feedback met FC3 (Refers to the work with given examples). |

*Table 29 Chapter 5 Summary of findings*

Using qualitative statistical tests and quantitative analysis of the developed feedback criteria, historical data were analysed to investigate if the issues perceived by students within chapter four were present within feedback. Table 29 gives a summary of the key points identified within this chapter. The results show that the issues identified in chapter 4 were present within the feedback. While feedback identified what the students did wrong, there were rarely comments on why it was incorrect. Furthermore, the feedback focused on what the student did not do and therefore had an emphasis on negativity. The clarity of feedback was also often lacking, this was influenced by the feedback language as well as the system used to deliver the feedback. Pre- and post-moderation as a procedure improved the consistency of the feedback as well as the quality, however, the issues were still present within the feedback.

# Chapter 6. Computing Ipsative Assessment

### *6.1 Introduction*

This chapter outlines the development and use of the tool, Computing: Ipsative Assessment (C:IA). C:IA aims to investigate the potential benefits of an Ipsative form of assessment, which compares a student's previous performance to their current performance, to help ameliorate the perceived issues identified in Chapters four and five. The technical and non-technical requirements are outlined and the design and implementation are discussed. Finally, the results of a user trial to understand the potential benefits and drawback of a system are discussed.

### *6.2 Rationale for system development*

Ipsative assessment is a process which may help ameliorate student dissatisfaction within higher education. The qualitative and quantitative tests from chapter four and the feedback analysis from chapter five identified issues perceived by students that are also present within the feedback. The issues identified include:

- A lack of assessment literacy within students.
- Disengagement from feedback that did not explore mark allocation and was not programming related.
- Modularisation of higher education courses causing feeding forward to be difficult to understand.
- The feedback given to students was too negative.
- Feedback was not quickly returned to students.

This section identifies how Ipsative assessment can help ameliorate these issues and why a computer-based format of Ipsative assessment is the most appropriate solution.

#### *6.2.1 Ipsative as a solution*

Table 30 identifies the perceived issues of students and how ipsative assessment as a concept may help reduce the impact of these issues. This section then further explores how a computer-based system will enhance the use of Ipsative assessment.

| Findings | Page | How can ipsative assessment help? |
|---|---|---|
| Students have a theoretical understanding of assessment literacy but do not always put these skills into practice, specifically with regards to understanding where their knowledge level is and how to plan for future | 89 | Ipsative assessment has a significant focus on using assessment to help students understand their knowledge level and to help guide them in future regulation of their learning by reflecting on their past experiences and how they have improved |
| Modularisation has caused the links between assessments difficult to identify. | 69 | Ipsative assessment can help students to identify the skills they need to develop from each assignment as opposed to focusing on the criteria for each individual assignment. By identifying the skills needed in each assignment the student can then map these skills to other assessments. |
| It is not always clear where marks and feedback align with the specification. However as shown in Figures 9, 10 and 11 the completion criteria are clearly stated. | 82 | Placing the focus on how the student can improve in the next assignment with specifical goals, while also bringing attention to how the student improved from the precious assessment will give the student clear tasks that are aligned to mark gains. |

| | | |
|---|---|---|
| Well performing students do not get the same level of guidance with regards on how to improve in the future as lower performing students. | 68 | As Ipsative assessment is a personalised experience it gives every student similar levels of feedback. It will focus on how the students can improve the skills they use consistently in addition to the skills they have yet to use. |
| When the majority of feedback is identification of what criteria the student did not meet, this can decrease the student's confidence and motivation | 68 | As Ipsative feedback focus on not just the skills the student needs to learn but also how they have improved since the last assignment, it will also highlight the skills and tasks that students have done well. This will therefore positively re-enforce the growth mindset model and will help with increasing student confidence and motivation. |

*Table 30 Perceived student issues and how Ipsative Assessment may help*

Chapter 4 identified that while students have a strong theoretical assessment literacy, these skills were rarely put into practice. Ipsative assessment develops these skills within students by encouraging the student to self-reflect and self-regulate within their studies. By encouraging the use of self-reflection and self-regulation, students are able to identify where improvements need to be made based on targeted feedback. It does this by providing very specific tasks that the student has to achieve. By continuously using these skills a student will begin to identify how they can work at a higher level with a deep learning mindset (Winstanley, 2017, pp74; Eugenia, 2018; Housand & Reis, 2008, Tai and Adachi, 2019). Developing these skills will also ensure that on future assessments students will be able to identify the purpose of what they are asked to do in addition to the core functionality of the assessment in question.

If a student develops the practical application of their theorical assessment literacy, it will also reduce the disengagement a student has when dealing with work which does not involve programming or direct mark exploration. Developing a students' understanding that assessment and feedback is used to facilitate learning and not just to provide certification, in addition to developing student assessment literacy, may assist with attaching value to work where the purpose is not initially obvious.

A source of dissatisfaction for the students was that feedback was too negative (Section 4.3). It was reported via student interviews that while feedback identified what was wrong, and what not to do next time, it rarely explored what the students did well, or how they could improve the specific work. The negativity of feedback was found to be true through the feedback criteria analysis (page 112). Ipsative assessment and feedback can help to overcome these issues by focusing on the skills and knowledge that the student has, as opposed to the focus of the marks and what students did not achieve. Traditional assessment focuses on the external criteria, feedback for this assessment commonly focuses on what criteria the student did not meet.  This can lead to a “fixed mindset" within students where they believe that they cannot achieve the higher grades and complete the more difficult work (Dweck et al, 1998). The focus on what the student has achieved and what the next steps are will reduce the negativity associated with criteria referenced feedback and encourage a growth mindset within students.

An issue identified within the literature is that in the current structure of higher education, course modularisation has resulted in blocked content which makes feedback that feeds forward difficult to achieve (Hughes, 2017b, pp38). This can also be seen in student interviews where the purpose of assessments beyond programming (which occurs in a large portion of computing science modules) was not clear. An Ipsative assessment approach will help a student understand an assessment in terms of its skills and tasks and help identify how this can be applied to other situations.

### *6.2.2 A computer-based solution.*

Ipsative assessment is a practice which may help reduce the impact of the issues that cause student dissatisfaction, it is however a high workload practice. Ipsative assessment requires that in addition to marking the current assessment and providing feedback, the assessor also needs to understand and refer back to the improvements suggested by the students’ previous submission (Hughes, 2017, p9). This higher workload and the extra workload caused by increasing student numbers may cause the issues leading to student dissatisfaction to be exacerbated (UniversitiesUK, 2017).

A method to reduce the impact on teacher/ assessor workload is that of Computer-Based Assessment (CBA). CBA and computer-based teaching are uses of technology that can reduce the impact of the higher student numbers and the additional work created by Ipsative assessment

(Ferrell, 2012; Moscinska et al. 2008). CBA reduces the workload of teaching staff by requiring the workload to be in specific time frames, as opposed to a consistent higher workload throughout the academic year. CBA requires a larger workload at the beginning of the academic term, to provide the initial material, suggested feedback and possible goals. The tutor is then required to update the CBA at the conclusion of the term based on the experiences had by the student. While the workload and input are not a one-time requirement, by having automated feedback and guidance built into the system it does reduce the workload a staff member has throughout the year. It is important however that the technology does not replace the student-to-teacher interactions completely (Moscinska, 2018).

An additional benefit to using CBA, as seen in E-portfolios, is the ability for a student to clearly see their journey from novice to expert. CBA can be learner centred, it can allow work to be gathered and shown, it can track how a student is progressing, the tasks that they have completed and the work that they still need to do (Deneen, 2018). Using these features of a CBA can help to develop a growth mindset within students.

An issue identified in Section 4.3 and Section 4.4 was the timing of feedback. The process of Ipsative assessment would potentially have a negative impact on the timing of feedback if done by the tutor due to the workload involved. A CBA system however would enable instant feedback to be given to students to highlight areas in which they need to develop. The student would then be able to communicate with their tutor to receive further detailed instruction.

In their 2017 study, Chow and colleagues interviewed 130 participants across south-eastern American universities. Those interviews included staff from multiple roles across the university, including university faculty, administration, support staff, and graduate students. The survey found that the benefits to students using eLearning is the flexibility and convenience that e-learning allows as when using e-learning students can choose when and where they wish to study the course. While there are benefits to E-assessment and E-learning they can be challenging to implement. Chow and colleagues also discovered that there were some concerns that that the lack of face-to-face instruction could decline the quality of the teaching given. However, this is less an issue when E-learning and E-assessment are used alongside traditional forms of teaching (Chow et al., 2017). Moscinska and colleagues (2018) found that when the results are fair, instant and closely linked with the lectures students do engage with the material. These studies are evidence that CBA and learning can be beneficial to students.

Due to the identified benefits of a computer based Ipsative assessment and feedback system along with its potential to ameliorate the issues causing student dissatisfaction, this chapter explores the development and user trial of such a system.

### *6.3 System requirements*

This section identifies the requirements for an online Ipsative assessment based on the findings from chapter 4 and chapter 5.

#### *6.3.1 Functional Requirements*

**FR1 The content of the system should explore the course material as defined in the CSC1021 module.**

As this case study took place at Newcastle University, the programming language that the system focuses on will be Java. However, the system itself will be adaptable to include any subject that the user wants to teach. Moscinska states that when using technology to assist teaching, the material should be as close as possible to that of the original content (Moscinska et al, 2018). Furthermore, by keeping content as similar to the original course material as possible, it should allow for the Ipsative aspects of the system to be tested, keeping the material as a controlled measure.

**FR2**: **User accounts should be provided for students to have a personalised learning experience.**

Ipsative assessment requires a personal component that allows students to view their past progress (Hughes, 2014). Within Ipsative assessment a student is required to investigate past work and assessment points to identify not only how they have improved over time, but also future improvements that they need to make to their knowledge state to complete any missing gaps. The personal component should be supplied via the use of personalised accounts which a student can use to log in and store records of their achievements.

**FR3 Assessment points should be provided for students to develop their understanding of their current knowledge level and required learning gains.**

As this system will be focused on exploring if Ipsative assessment can help ameliorate the issues identified by students, the assessment within the system needs to be fair and valid. This includes ensuring that it is clear what the student is completing in terms of assessment and where they are focusing their attention. It also means that what the students are working on in the system should be authentic and be clear on how it relates to their CSC1021 module (Wakeford, 2003; Hughes 2011). In Chapter 4, Section 4.3.3 (and point 1, Table 29 Chapter 5), staff interviews identified that students focused on the functionality of the code and ignored design aspects. As the programming module that the system will be used alongside has a primary focus of code functionality and syntax, the concept of giving code snippets to students to work from should be used to highlight the importance of coding practices beyond functionality. The implementation of MCQ quizzes will allow for a focus to be on the structure and design of the code, in addition the syntax and functions. By separating these quizzes into topic sections, it will also allow the teacher to identify what topics are not being covered by the students, or which topics students find most difficult. The system will ensure that the students gain practice working at all levels of Bloom's taxonomy, using the extra materials to work within the application and analysis levels, as these can often be avoided by students (Bloom, 1956; Burgess, 2005). As identified in Chapter 5, feedback given to students in the study rarely discussed why something was incorrect (point 5, Table 29, Chapter 5). Therefore, when students take a quiz in this system, they should be able to see if they choose incorrectly and what was wrong about their answer.

**FR4 The system must allow the development of self-referential goals.**

An important part of Ipsative assessment is for the students be able create goals and plans to regulate their learning. The students should be able to view past completed goals to see their improvement to encourage a growth mindset. Goal setting and the implementation of self-regulation will help develop a student's assessment literacy by identifying from the assessment points and results section, which aspects they need to improve. Furthermore, it will assist in developing a student's long-term progression, by focusing not on the assessments within the modules but the skills and knowledge that can be applied in places other than the current assessment. Finally, this will allow students to see the development that they have been making throughout the process, removing the focus from the negative aspects (point 7, Table 20, Chapter 6) of what they have not achieved.

### *6.3.2 Non-functional requirements*

**NFR1 The system needs to be able to work across multiple machines with varying specifications**.

As the aim of the system is to teach novice programmers, the system needs to be able to work across multiple platforms, including the common operating systems such as Windows (Windows, 2020), Linux (Linux, 2017) and macOS (Apple, 2020). Therefore, the technologies used must be able to work across all these platforms. Due to this HTML, CSS, PHP and JavaScript are the languages used in the building of this system.

**NFR2 The system must have an intuitive design.**

When designing technology, it is often considered good practice to use a set of design heuristics. Nielsen defines ten heuristics that should be used as guides that a designer should try to adhere to when making a system (Nielsen, 1994). The heuristics and how they should be implemented are as follows:

*NH1 – "The visibility of system status should be clear. The system should always keep users informed about what is going on, through appropriate feedback within a reasonable time."* (Nielsen, 1994).

> To reduce the amount of loading time needed by the system, content will be loaded in one instance. As the tool will be a website, users will be able to see the consistently see the status of the tool. Furthermore, when completing a quiz, the user should be able to see what answers they have selected within the quiz at all times.

*NH2 - "There should be a match between system and the real world. The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. It should follow real-world conventions, making information appear in a natural and logical order."* (Nielsen, 1994).

> To achieve NH2, the system will follow a similar content structure to CSC1021, which means that student expectations of what the system will do are met. The system should also follow similar terminology to that used in both the lectures and the suggested reading material for the course.

*NH3 - "The system should provide good error prevention. Even better than good error messages are a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action."* (Nielsen, 1994).

> User input and navigation will be kept to a minimum to reduce the possible errors that can be made. However, if a mistake or error is to occur, C:IA will use informative messages throughout the system to inform the user of correct use. For example, an error message is

supplied to the user via the quiz section if they have not completed the correct amount of questions. Error messages for technical failings should be clear and should inform a user to contact the primary researcher.

*NH4 - "Recognition rather than recall. The system should minimise the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for the use of the system should be visible or easily retrievable whenever appropriate."* (Nielsen, 1994).

The design of the system should follow a structure that is familiar to the student. With regards to the code layout shown. To help ensure consistency and familiarity to what students use within the classroom C:IA should use an IDE to display code snippets, such as Eclipse screen captures (Eclipse, 2001).

*NH5 – "Aesthetic and minimalist design. Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility."* (Nielsen, 1994).

C:IA should keep design simple to help students focus on the examples given in the system. Bootstrap (Bootstrap, 2011) will be used to ensure a design that is consistent with good design practice.

*NH6 - "Help users recognise, diagnose, and recover from errors. Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution."* (Nielsen, 1994).

The need for users to be able to diagnose errors in this system is minimal. As this website is designed for novice programmer, it is assumed all potential errors will be familiar to the student, such as page not found errors or database connection issues.

*NH7 – "Help and documentation. Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large."* (Nielsen, 1994).

The students should be given training on how to use the system. Tooltips should be provided to help guide students throughout their use of the system. A primary factor to developing the system in the form of a website was to adhere to familiar standards that the students use throughout their course. Should the student discover a bug or issue that needs to be reported, they should do this through an About page which will contain information on how to contact the developer.

*NH8 - "User control and freedom. Users often choose system functions by mistake and will need a marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. The system should support undo and redo."* (Nielsen, 1994).

To reduce the possibility of error-prone states, all material will be loaded upon the initial user connection. Furthermore, when answering quizzes, students should be able to exit the area and have no impact on their results or score, an exit should not be classed as a fail.

*NH9 – "The system should adhere to consistency and standards. Users should not have to wonder whether different words, situations, or actions mean the same thing, the system should follow platform conventions."* (Nielsen, 1994).

The system should be structured and consistent across the platforms that students can use. There should be no differences based on the technology used by the students.

*NH10 - Flexibility and Efficient use of accelerators. Unseen by the novice users' accelerators may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users and allow users to tailor frequent actions.* (Nielsen, 1994).

This will not apply to this system as students will not have control over the shortcuts or features within C:IA.

**NFR3 The system should be maintainable and scalable for future usage.**

A potential cause of student dissatisfaction is the increase of student numbers making personalised difficult to manage. An aim of developing an online Ipsative system was to manage these large class sizes and to give students quick guidance on their work. Therefore, the system needs to be usable for a large number of students across multiple years.

**NFR4 Data should be stored to allow students to view past results and allow teachers to identify problematic areas for the class.**

As the primary aim of this system is to facilitate an Ipsative form of learning, the data within the system needs to be accessible to the student throughout their enrolment. The student should be able to see more results than their previous iteration (Dweck 1998; Hughes, 2017b, pp32). This will allow the student to view the development that they have had throughout their time studying and encourage a growth mindset.

**NFR5 The system should have a quick response time to ensure attention retention**.

As identified by Nielsen's heuristics it is important that C:IA responds in a timely manner to keep users informed and engaged. In addition to the website development and quiz development, when loading the content of the system, all content should be loaded at the first instance of user login.

### *6.4 System design*

#### *6.4.1 Existing Systems that influenced the design of C:IA*

Teaching novice programmers how to code is not a new task, there are various websites, technologies and books available for students to learn how to code. This section explores parts of the existing systems that were good and advantageous, but also the limitations and why it they may not be suitable to use in this research. Finally, an overview of the concepts inspired by the systems is presented.

To ensure that all systems are reviewed based on the system rationale (page 125), the requirements of the system (page 130) as well as the findings from Chapter 4 and Chapter 5, the following criteria is used to evaluate the systems

1. The system needs to teach students how to code in Java. It should have opportunities for the student to be creative and experiment.
2. The user needs to have personal learning experiences.
3. The system should encourage and assist the student with reflection and regulation of their own learning.
4. Feedback given to the student should inform the student why they were wrong, and what the correct answer should be.
5. Must allow for the implementation of Ipsative forms of guidance, such as adding goals and viewing past achievements.
6. Must be accessible across multiple platforms via the use of web development methods.
7. The system should allow for the storage of data for long term usage.

### *6.4.1.1 Codecademy*

Founded in 2011, Codecademy is an online learning platform (as seen in Figure 19) developed by Zach Simms and Ryan Bubinski. Codecademy has a mission to teach a student to learn to code. Codecademy teaches novice programmers how to code by explaining the functionality of each line of code and dictating what students must write. There is a brief description of what the function does, a coding platform, and an output display. When entering the code correctly, Codecademy, compiles, runs and displays the output of the code (Codecademy, 2011). Therefore, Codecademy meets evaluation criteria one.

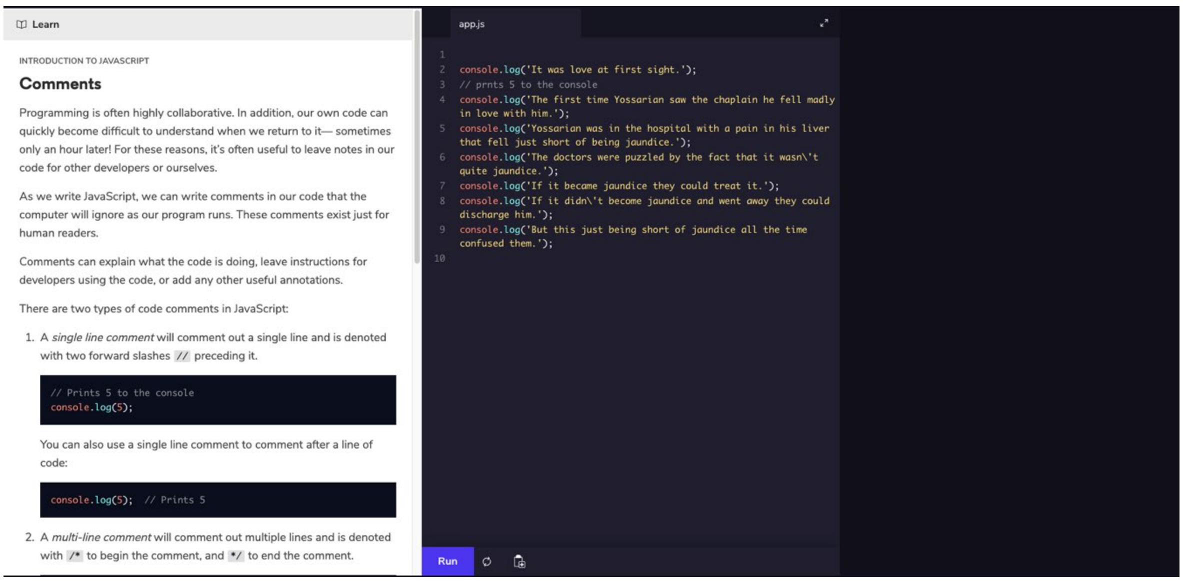


*Figure 19 Codecademy example*

The coding aspect of the website allows the user to practice writing code in a structured way, following a step by step guide on how to program. The system also allows a novice programmer to practice reading code by providing code snippets in an easy to read Intergraded Development Environment (IDE). There are, however, limitations to what Codecademy offers. Feedback is limited due to the type of system Codecademy is. It matches the output of the program with an expected output and determines the correctness, in essence, a unit test approach. This type of coding teaches students little about code design and can often have students patching their code to produce the required output (Whalley, 2006).

While Codecademy follows a good structure for students to learn to code, there are limited methods of tracking the understanding of the material once they have moved onto the next task. Another limitation of Codecademy is the lack of creativity when learning new concepts. Students have to type exactly what the system states. They are taught how to code by dictation, not experimentation. Figure 20 shows an example of a student learning to comment multiple lines at once. In the instance shown, the system is searching for '/*' which comments multiple lines at once. However single line commenting is also a valid (if not good practice) way of coding and is in this case shown as an 'incorrect program'. The format chosen does reduce the impact of students patching their code, however, it removes one of the core features of programming, problem-solving and creativity (Lahtinen, 2005). Codecademy does not meet evaluation criteria four due to this lack of explanation of why their code was incorrect.

Codecademy does meet evaluation criteria two by having a personalised user account which the student can track their progress through the material. It does not however meet evaluation criteria three, as the tool encourages students to go through each step and does not encourage backtracking or further exploration of the topics. Additionally, Codecademy does not meet evaluation criteria five, as it does not provide areas for targeted specific goal setting. It does inform the user of how much of a topic they have completed but it does not guide the student in identifying areas of weakness. Codecademy is web-based platform accessible across all web browsers, therefore meets evaluation criteria six. While Codecademy stores users record of which aspects they have completed, the system stores a yes/no identification on the complete status. It does not store information concerning how many attempts the user took on a subject or identify weaknesses. Therefore, it does not meet evaluation criteria seven.

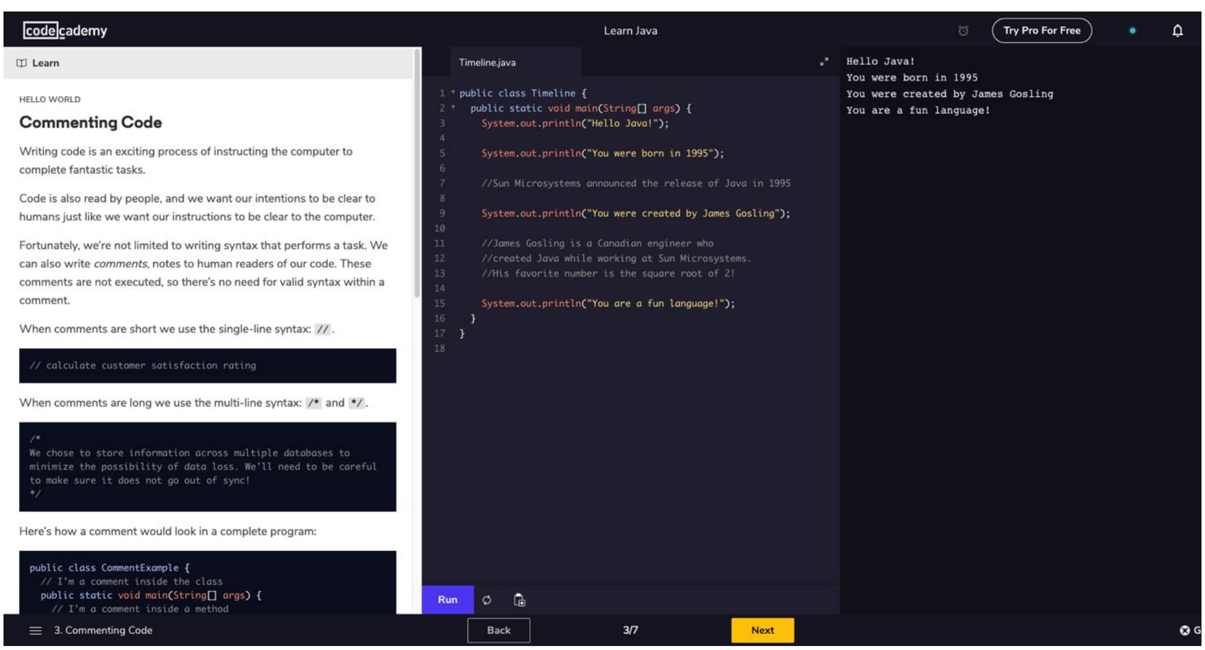


*Figure 20 Codecademy*

For C:IA the idea of taking existing code and asking students to apply the correct snippet is a good learning tool for investigating if the student truly understands the code. While C:IA will have quiz sections as opposed to a live IDE to develop in. The quizzes will follow the same principle as Codecademy. The students will be given a selection of code and be asked to identify the correct code snippets to fit an identified gap. While this runs the risk of the student getting the question correct by guessing the answer, it will still allow them to see the correct result giving some context for their future learning.

#### *6.4.1.2 Khan Academy*

Khan Academy (KhanAcademy, 2008) created in 2008 offer teaching platforms for a variety of subjects, including computer science. The website is structured as a large learning platform, with very small sections explaining one aspect of a topic. Khan Academy does not meet evaluation criteria one as while multiple languages are taught within this tool, Java programming is not. As Java programming is the language used for the module in which C:IA will be deployed, this software is not suitable to use with this research but may have useful features. Figure 21 shows and example of Khan Academy and their small content selections to help students regulate their learning.

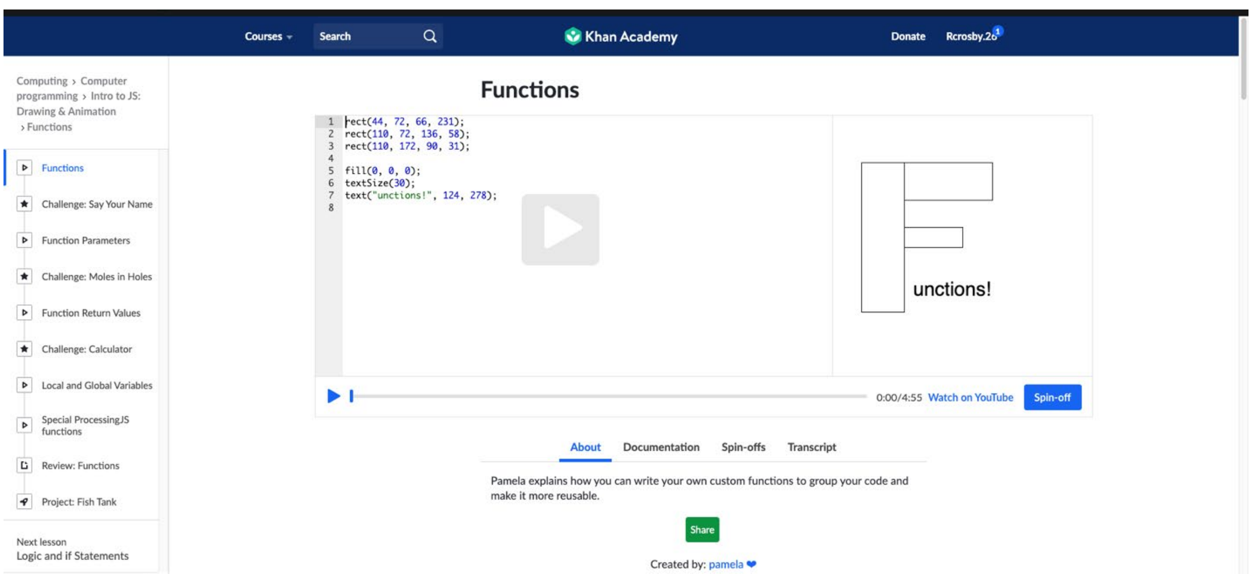


*Figure 21 Khan Academy coding example*

Kahn Academy allows a student to have a personalised account therefore meeting evaluation criteria two. It also allows the student to explore a subject in whichever structure they like while also having an option to ask for clarification. The tool also meets evaluation criteria three by providing a community aspect to the system. It allows the learner and other users to make comments in each section, asking for clarification on different aspects. The community aspect will help identify subjects with which the student is unsure of and prompt the student to explore this further.

Similarly to Codecademy, it provides a development platform with which a student can practice coding and receive feedback. Unlike Codecademy, Khan Academy meets evaluation criteria four by providing informative error messages to the students which explain why their solution will not work. As seen in Figure 22, Khan Academy informs the user that the variable ‘there’ is undefined and gives a possible alternative. Khan Academy does not meet evaluation criteria five, as while there are features available for users to track their usage, there are no features such as goal tracking and specified learning targets.  Khan Academy is web-based platform accessible across all web browsers, therefore meets evaluation criteria six. Khan Academy does not meet evaluation criteria seven as while it stores information regarding a student’s attempts and scores, it only stores the last attempt made by the students, it does not track if the student used the hint feature (KhanAcademy, 2008). While suitable for identifying where the student needs to improve, it does not store more than the previous iteration making it unsuitable for developing an Ipsative and Growth mindset.

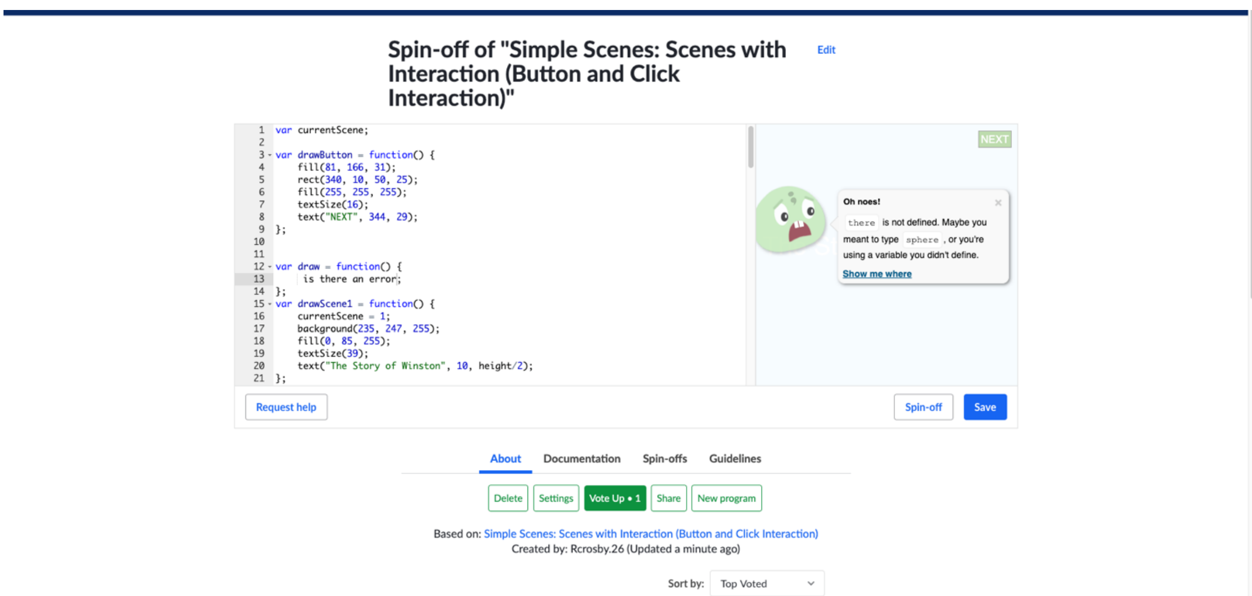


*Figure 22 Khan Academy Error Codes*

#### *6.4.1.3 Numbas*

Numbas (Newcastle University, 2019) is a software application developed by Newcastle University which aims to teach mathematics to undergraduate students. Primarily Numbas is an e-assessment tool, designed to ask questions, or conduct 'exams' during the course.

The system is an open source e-assessment tool where the teacher writes questions about a chosen topic (Perfect, 2015). As the system is an e assessment open source software, evaluation criteria six is met as Numbas is runnable on all web platforms. Due to the text input not having a maximum cap, a user can add more detail to the question and turn it into a tutorial website. Figure 23 shows an example of this for material developed by Becky Allen on Machine learning (Allen, 2020). Numbas does not meet the first evaluation criteria, as the system is used for mathematics as its primary topic. Additionally, Numbas aim is to set 'exams' not to help students understand the material of the course. There are multiple types of questions which can include 'multiple-choice' or free text, each with their own rules. The system itself is not designed for teaching in addition to assessment, although content can be supplied by placing it within the question, however, this option is limited.

Numbas allows a personal account for assessment and allows the students to track their progress in exams meeting evaluation criteria two. Numbas also meets evaluation criteria three and four by allowing the assessor to place as much feedback for a question as needed. The assessor can give step by step instructions on what a sample solution should be. While this is limited in the personalisation it gives it a student, it can assist a student in identifying what components they need to develop and how they can further improve. Numbas also achieves evaluation criteria seven by allowing the student to access previous results through their VLE, Numbas allows students to see all previous attempts.

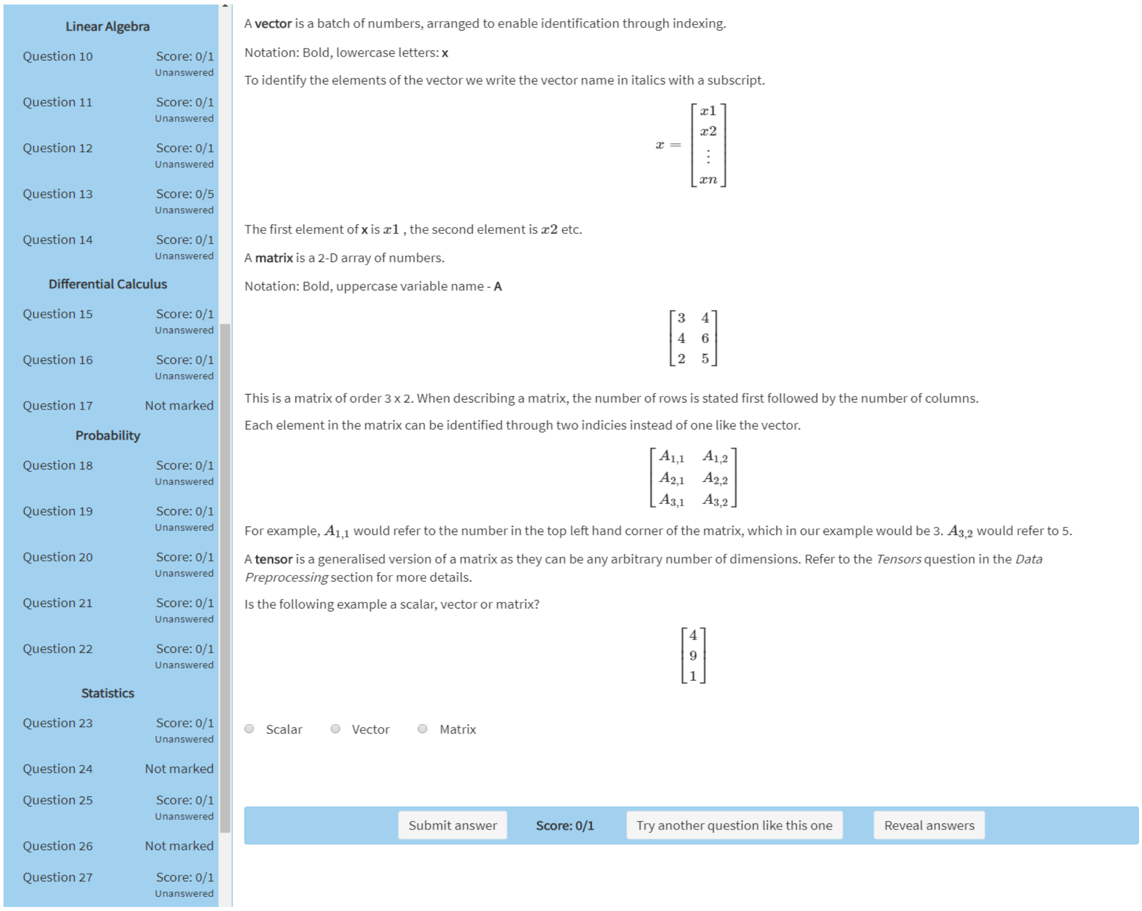


*Figure 23 Numbas example*

The system does have benefits from a developer side, in that it is free to use, but most importantly, it can be integrated into a virtual learning environment (VLE). Once this has been integrated, it allows the developer to track who has used the tool and what questions they answered. However, if Numbas is not released via a VLE, this function is not available. The content can also be edited via CSS and HTML by the developer.

When considering evaluation criteria five, there are no features that can be applied from the Ipsative viewpoint, such as goal setting and target guiding by the system, this must be done externally. The benefit of using this tool is viewing student usage. To obtain this feature C:IA will be developed using databases connected to the Newcastle University homepages, allowing the admin of the system to view all interactions.

| Evaluation Criteria | Codecademy | Khan Academy | Numbas |
|---|---|---|---|
| 1 | Yes | No | No |
| 2 | Yes | Yes | Yes |
| 3 | No | Yes | Yes |
| 4 | No | Yes | Yes |
| 5 | No | No | No |
| 6 | Yes | Yes | Yes |
| 7 | No | No | Yes |

*Table 31 Feature Table*

Table 31 shows a feature comparison based upon the evaluation criteria of each system. As can be seen none of the systems meet all of the evaluation criteria, and no system met evaluation criteria five. Therefore, the system developed must ensure that this is a focus to help guide students through their studies.

### *6.4.2 System usage*

While the system will focus on novice programmers, the full aim of the system is to be useable across multiple subjects.

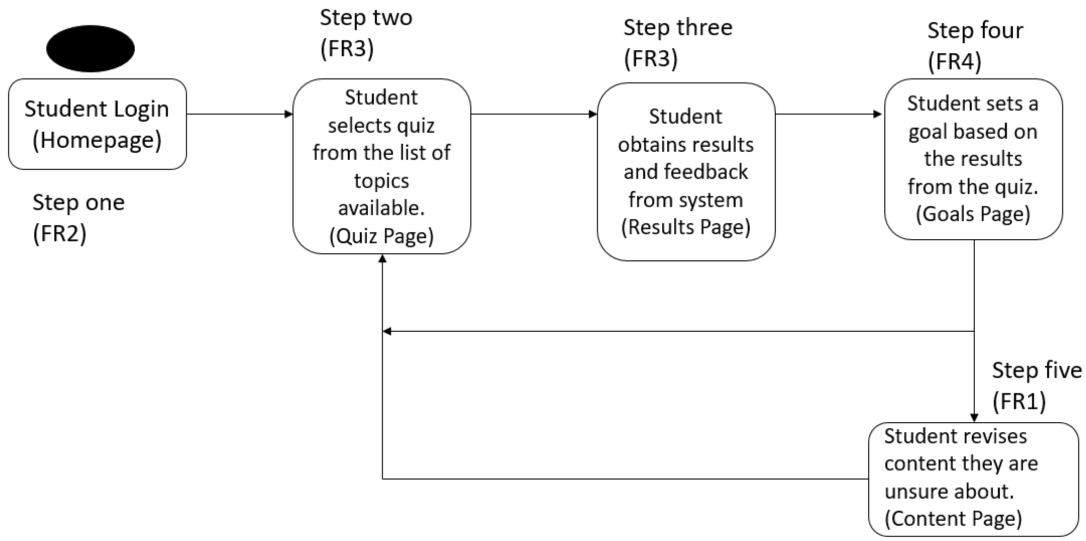


*Figure 24 Expected student use of the system*

There are four aspects to using the system, (i) learning the content, (ii) taking a quiz, (iii) identifying where there are gaps within a student's knowledge and (iv) goal setting. Figure 24 shows the steps taken in the system by the student, as well as which system requirement each step fulfils. Step one in Figure 24, represents the student logging into the system where they land on the homepage. The homepage informs the student of their latest results and what quizzes they have yet to take. If there are subjects that the student has not completed a quiz in C:IA will prompt the student to complete this quiz. This first quiz is an initial diagnostic test for the student, informing the student their current knowledge level (Hughes, 2017) and the beginning of the Ipsative structure. Step two (Figure 24) identifies the student selecting the quiz and topic that they wish to take. Each quiz will ask ten questions in total with a range of difficulty. Once the student has completed the quiz, they will gain some feedback, this occurs in step three. This feedback given involves which questions the student did not get correct. There is also the option for students to view what the correct solution should be. The system however advises the student to not look at the correct answers and instead develop their skills in this area. At step four, the student creates goals based upon the results of the quizzes. This will involve two choices, the first is to create a goal where they need to revise some content, in which the student would progress to step five. If the student is happy with their score on the quiz, they can then choose another topic to engage in, and therefore return to step two repeating the cycle. At each step in the cycle students should be reviewing their goals and keeping them updated.

The main design feature of C:IA is the Ipsative framework, to implement this and the requirements four main concepts were considered:

1. To achieve FR2, a homepage detailing the student's previous results and informing them what subjects they needed to focus on should be created.
2. FR3 will be achieved by the implementation of a quiz page. This will allow the user to assess themselves and, provide a baseline for their learning. The removal of marks towards the degree will allow students to experiment more within the learning. As noted by Black (1998) as well as Black and William (1998) comment only marking removes the focus from the summative aspects of the assessment and allows a student to experiment in their learning and creativity.
3. To fulfil FR4 a results page will be developed. This results page will break down each of the student's quiz scores and allow the student to get feedback such as what the correct answer should be. Additionally, the results page should provide guidance on how their answer differs from the correct one. This will also identify why an answer was incorrect as this was found to be a perceived issue an issue in Section 4.3 page 66 and was noted as lacking in the analysis of historical feedback (page 112).
4. A goals page will allow students to regulate their own work and allow them to set specific, trackable goals, which are an important component to forward-facing Ipsative assessment (Hughes, 2017, pg7). The goals feature will also remove the focus on the negative aspects of feedback as identified within student interviews (page 68) by showing the students where they have improved. This regulation will also help with student's assessment literacy skills. This is achieved when students are able to identify what topics and skills they need to work on and how this can be applied in another situation. Developing the skills from the quizzes and the goals section, should allow students to apply these skills on their feedback from other assessments.

To ensure that NFR3 and NFR4 are met, the system should use the database connections provided by Newcastle University to store user accounts and the associated data. This will ensure that the C:IAs database is maintained and is scalable through the central university IT system. It will also enable data to be stored in a safe and secure way.

Figure 25 shows the design of the content structure within C:IA. The contents page is where the students will learn and review the material. The design will have small deliverable sections, using tabs to navigate through the system. A tab-based system will help achieve FR1, the delivery of content. This will also achieve NFR5 as it will ensure that all material is pre-loaded and keep reaction time to a minimum. Good pedagogical practice will be adhered to by giving small sections of information organised in a way that connects the material. With the aim of the system focusing on use alongside the course, C:IA needs to have a low cognitive cost to its use. Therefore, C:IA needs to have a simple design with few pages.

Minimal selections will assist with NH1, "Visibility of the system status. By keeping all material in one place consistently and having material pre-loaded using tab systems, it will be clear to the students what state the system is in. Nielsen's heuristic 9 "Consistency and standards" state that standards across the system should remain the same and match with real-world expectations (Nielsen, 1994). The system adheres to NH9, by keeping all pages within the system consistent in terms of work feel and behaviour, there should be no confusion on how the system works.

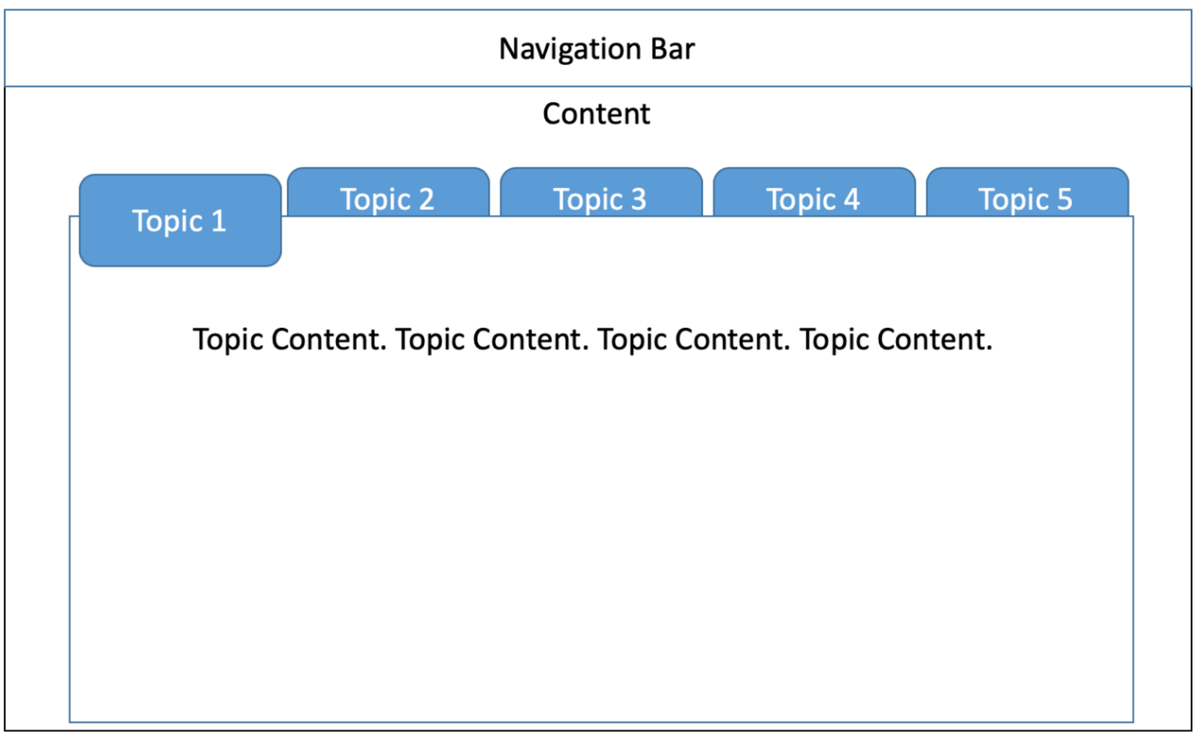


*Figure 25 Content page design*

As C:IA is designed to trial Ipsative assessment, the implementation of an assessment structure was important. Figure 26 shows the design of the quiz pages and the beginnings of the implementation of FR3. These pages will have a structure similar to the content pages, with each topic loaded via separating tags. The system should connect to a database and contain four tables for the quizzes. Table one should contain the bank of questions that are selectable for each topic.

From this bank of stored questions, the system should randomly retrieve 10 and display them, and their possible answers to the student. Table two should contain the potential options and identify which is the correct answer. The third table should contain the student's submissions. Storing the submissions will allow the answers to be marked and stored for future reference. The fourth table will contain the results of a student quiz along with the date and time that they were taken. To ensure "Error prevention" (NH3) (Nielsen, 1994) the quizzes will be created so that the user can select one answer option for each question. It also ensures that the user can easily undo mistakes they make in both answer selection and quiz selection. The ability to cancel quizzes without consequence covers NH8 "User Control and Freedom" (Nielsen, 1994).

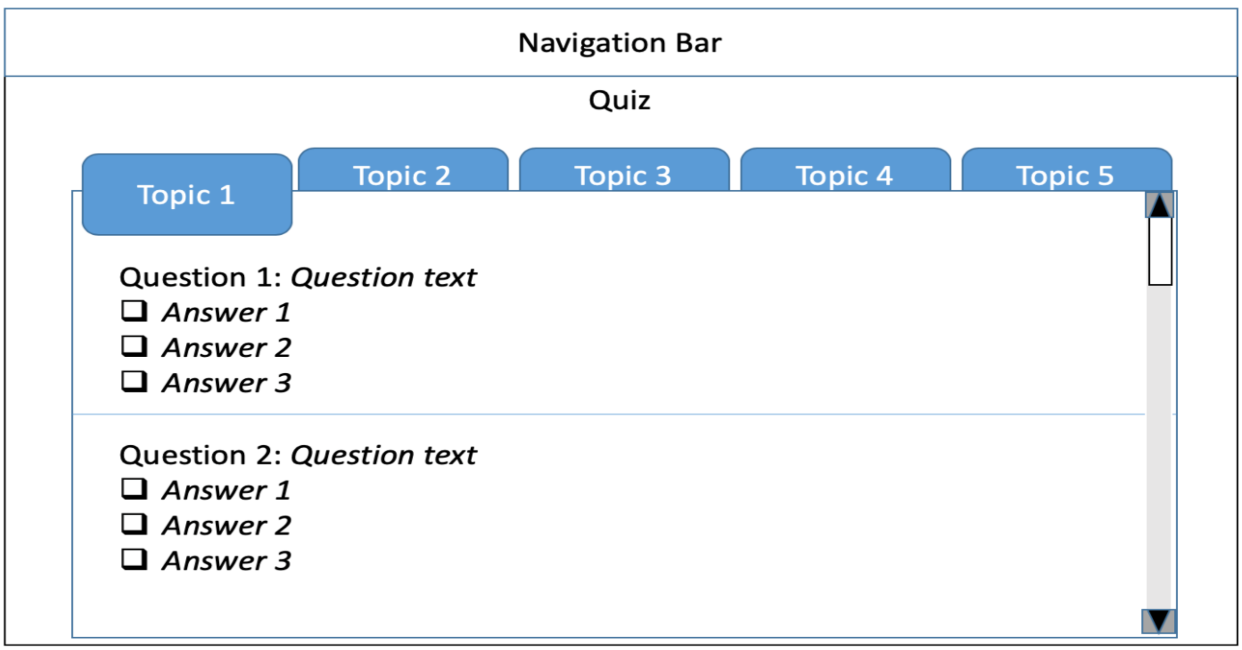


*Figure 26 Quiz page design*

Figure 27 shows the design of the Results page, it is structured in the same style as the content and quiz selection. To achieve this the system should connect to a database and retrieve the results of all quizzes previously completed by the student. Once these results have been collected the results page will give an indication of the scores a student has on each topic, along with what date the student took that quiz.

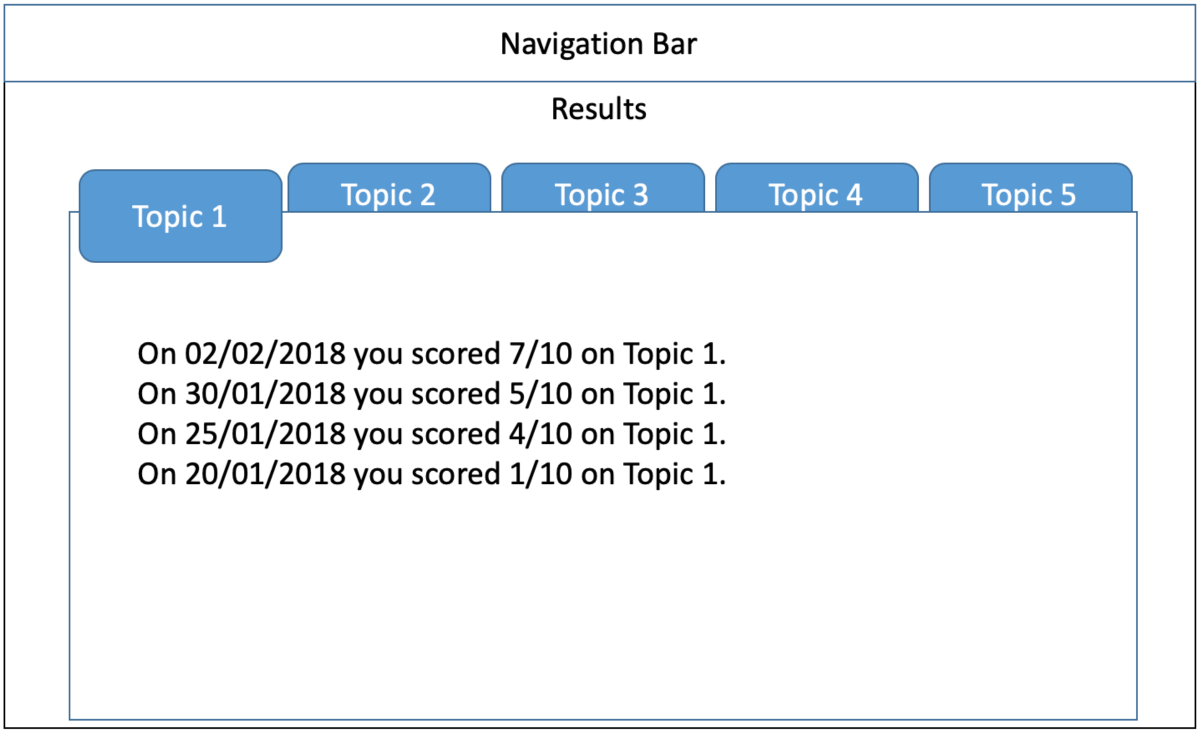


*Figure 27 Results design*

To avoid focusing on the negative aspects of the results, the results page will show each attempt the student completed allowing the them to see their improvement. Rather than giving a percentage or an indication that specific scores are bad, the system should prompt the student to revise a topic they did not do well in. This will help the student regulate their learning by focusing on their areas of weakness (Hughes, 2014; Ott et al 2016).

An aspect inspired by Khan Academy (page 139), will be a goals page (Khan Academy, 2008). Khan Academy has a community aspect where students could ask help from other users of the system. While a great resource for students, Ipsative assessment focuses on an individual growth. While a community page will allow for students to help each other, it may also foster an aspect of competition therefore, rather than a community page, the system will instead use an individual goals page. The goal page will be the primary form of Ipsative methods for the students identified in FR2 as well as FR4. The goals page is where a student can track what areas they need to improve upon, while also tracking their previous activities. To achieve this there should be three functions that the system should have. First a user should be able to store text based inputs into a database. Second there should be boolean condition attached to the text input, on the completed status of the goal. The third function of the goals page should allow the user to fetch all current and completed goals. Figure 28 shows the design of these sections. The aim of this page is to ensure that after each quiz the student adds a new goal. This will help steer the student to focus on not what went wrong but on the future directions that they can take, thus removing the negative aspects of feedback by focusing on ways to improve.

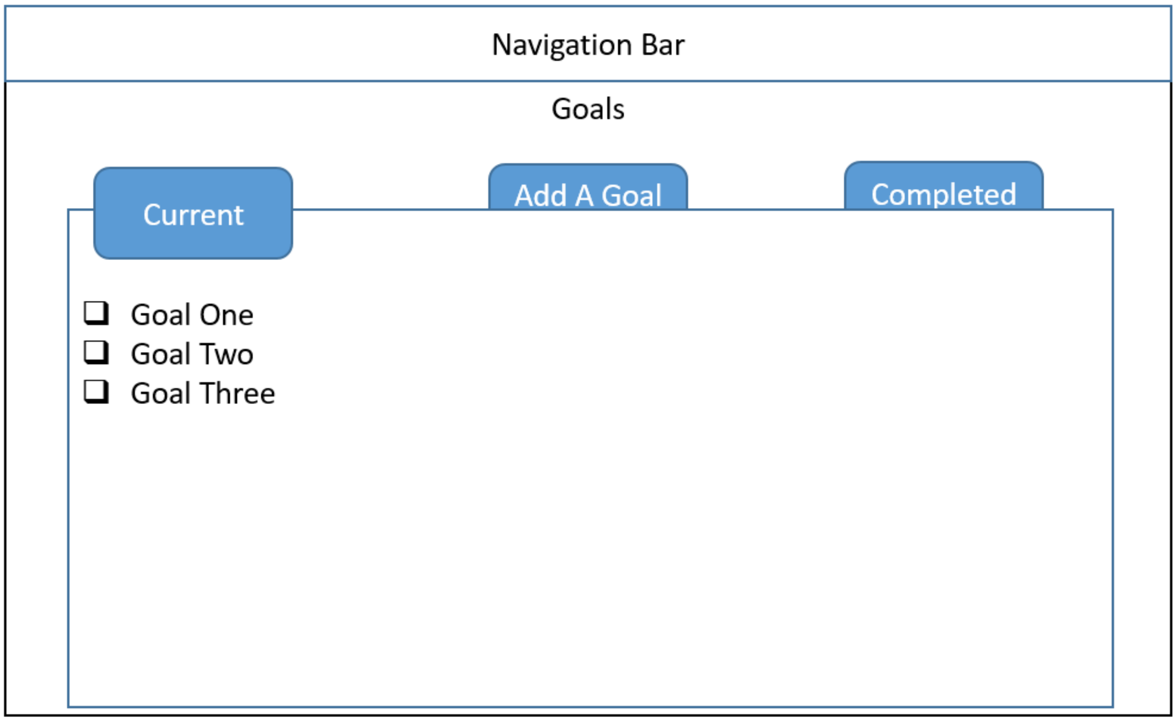


*Figure 28 Goals page design*

The goals page is intended for the students to set themselves learning goals. Ipsative assessment has the potential to help guide students and tailor their learning towards their individual needs. This section aims to help encourage self-reflection and self-regulation to develop a deep mindset (Ott et al., 2016). The development of a deep mindset was found to be lacking in Section 4.3 (pg 69) where Teacher A identified that students take a surface learning approach. The ability for students to see past performances and identify areas in which they struggle will help them regulate and target their learning for future improvements.

Additionally, while students had a good theoretical basis of assessment literacy, the students rarely put this into practice. The use of the goals section to target their learning will ensure that students are identifying what skills are used in each topic and potentially extrapolate them to other assessments and modules.

Figure 29 shows the design of the homepage. The homepage should allow the student to see what their current status is, and what topics they need to learn, fulfilling FR3. This will be achieved by fetching the latest result for each topic stored within the database. It will inform the student which topics they have yet to cover and which topics they will need to revise and study further.

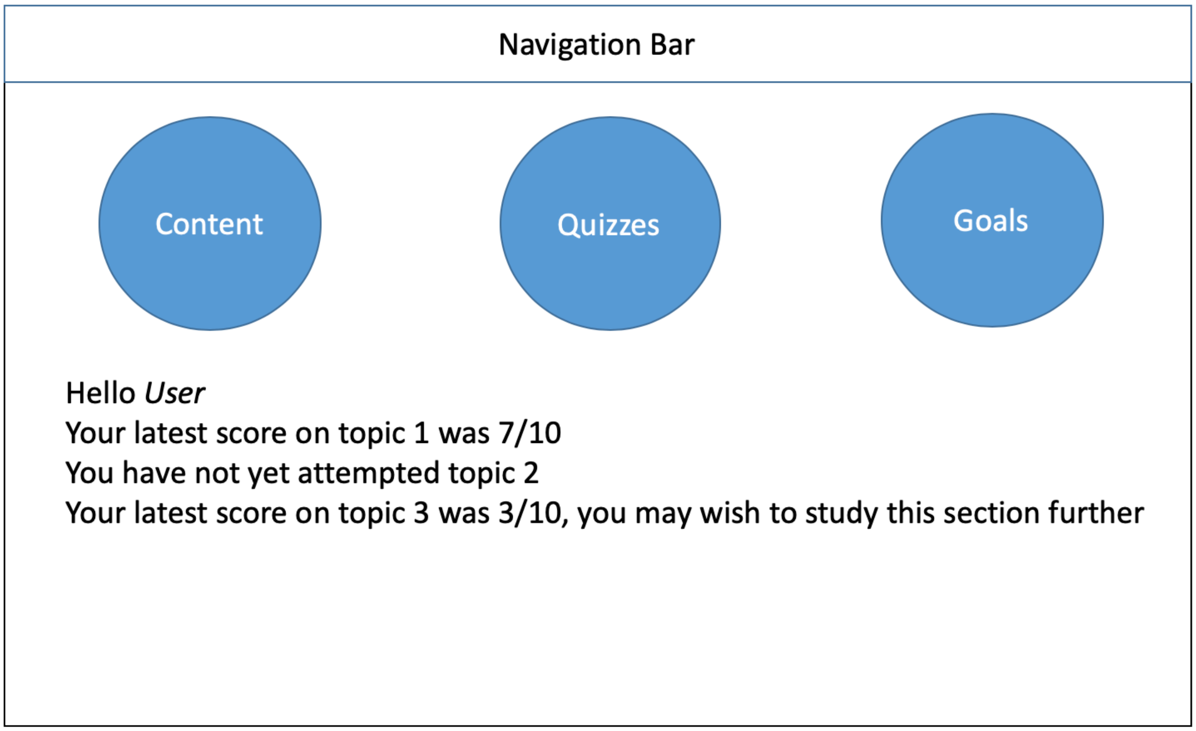


*Figure 29 Homepage Design*

The implementation of NH7 the use of help and documentation is adhered to via the use of emails between the researcher and the student.

These designs each implement a component of Ipsative assessment. The quiz sections included in the assessment component, provide an initial diagnostic test to help guide students to the areas they need to develop (Hughes, 2017, pp3). The results page informs the user of their past attempts and how they have improved in each section. After initial completion of the quiz, it also allows the student to see the correct answers. To incorporate the forward-facing aspects of Ipsative assessment and feedback, the students will have a goals page to develop specific achievable goals for future use (Hughes, 2017, pp7). The trial of C:IA will be used to answer RQ3, how does Ipsative assessment compare to current practice and can it be used to ameliorate issues identified by students?

### *6.5 System Development*

Computing: Ipsative Assessment is a web-based tool designed to investigate if Ipsative assessment could help to ameliorate the following issues as identified in Chapter 4 and Chapter 5:

- Modularisation of content made the links between assessment difficult for students to see.
- Feedback focused on the negative aspects as opposed to the positives.
- Feedback did not inform students why something was incorrect.
- A lack of practical assessment literacy within students.
- Feedback given to the students was too slow.

To ensure that the system met NFR1 C:IA was developed using HTML, CSS and PHP. This ensures that the system could work on any platform that the user has access too. Bootstrap (Bootstrap, 2011) was used to ensure that the tool functioned on a mobile platform. Additionally, MYSQL is used to connect to the databases provided by Newcastle University to achieve NFR3 and NFR5.

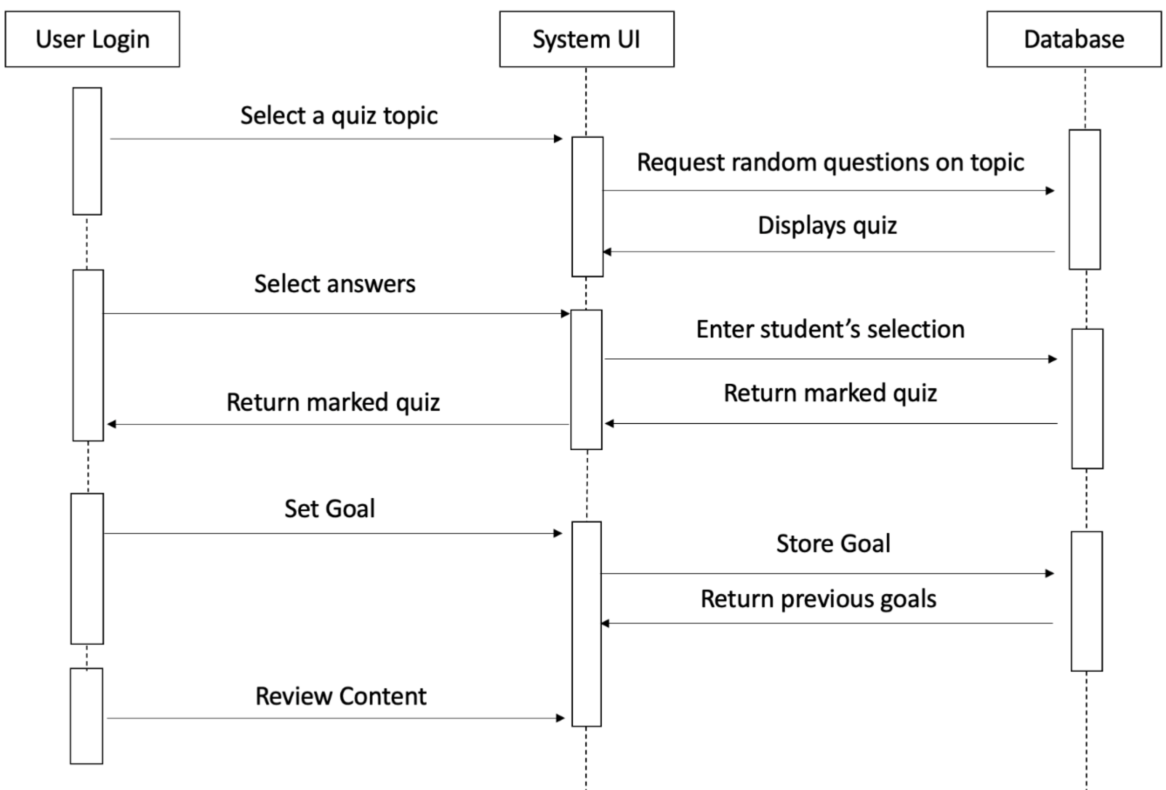


*Figure 30 Sequence Diagram*

Figure 30 shows the interactions that a student and the data have within the system. The primary function of the system centres around students taking quizzes, obtaining their results and planning their learning around this. Therefore, as can be seen in Figure 30, the majority of data flow occurs around these two areas. The first expected action from the student will be to take an initial diagnostic quiz to identify the users starting knowledge (Hughes, 2017). Once this test has been completed, the results are stored within the MySQL database, these results are then retrieved and displayed to the user. From this the students are then expected to set themselves goals dedicated on structuring their learning. These goals are also stored within the MySQL database and a student is able to see the current goals that they are completing and previously completed goals. After identifying what areas the student needs to work on, the content sections should then be used to review the material. Finally, the student should take a further quiz to identify if any improvement has been made.

Figure 31 shows the connections to and from each table within the database. Each user is connected via their user id to the tables containing the submissions of quizzes that they make. Within this table the time and date of when they took the quiz, the questions on the quiz and the chosen answer the student selected is stored. From this, the submissions are also linked via a question ID to the chosen questions. The question table has a one-to-many relationship with the options table. The options table stores the possible selections that a student can make on each question. For each question, there is currently one correct answer and three incorrect answers.

Additionally, the questions table has a many-to-one relationship with the quiz table, as each question can only be within one quiz topic. The questions table has a further relationship with the incorrect answers table, which stores the questions a student did not answer correctly, to be displayed at a later time. Further developing on the structure of these tables is Figure 32, which shows a UML diagram of the quiz classes. Each of these classes have their own roles, from generating quizzes to calculating and storing the answers students make.

Figure 31 and Figure 32 show the relationships between each user, their goals and results. Figure 31 shows the state of the system at the time of development. The relationships within the database have since been updated to reflect the one to many relationships that the results table has with students and quizzes. A developmental error resulted in this relationship not being present when the trial went live, this however had no impact on the use of the system from the student perspective. Therefore Figure 31 shows the state of the system at the time of the user trial, not the corrected ER diagram. Finally, there is a one-to-many relationship between module and quiz. The Module table stores information about the skills tested in each quiz. These details are not shown to the students as an aim of the system is to help further a students' ability to identify what skills they need to develop on their own.

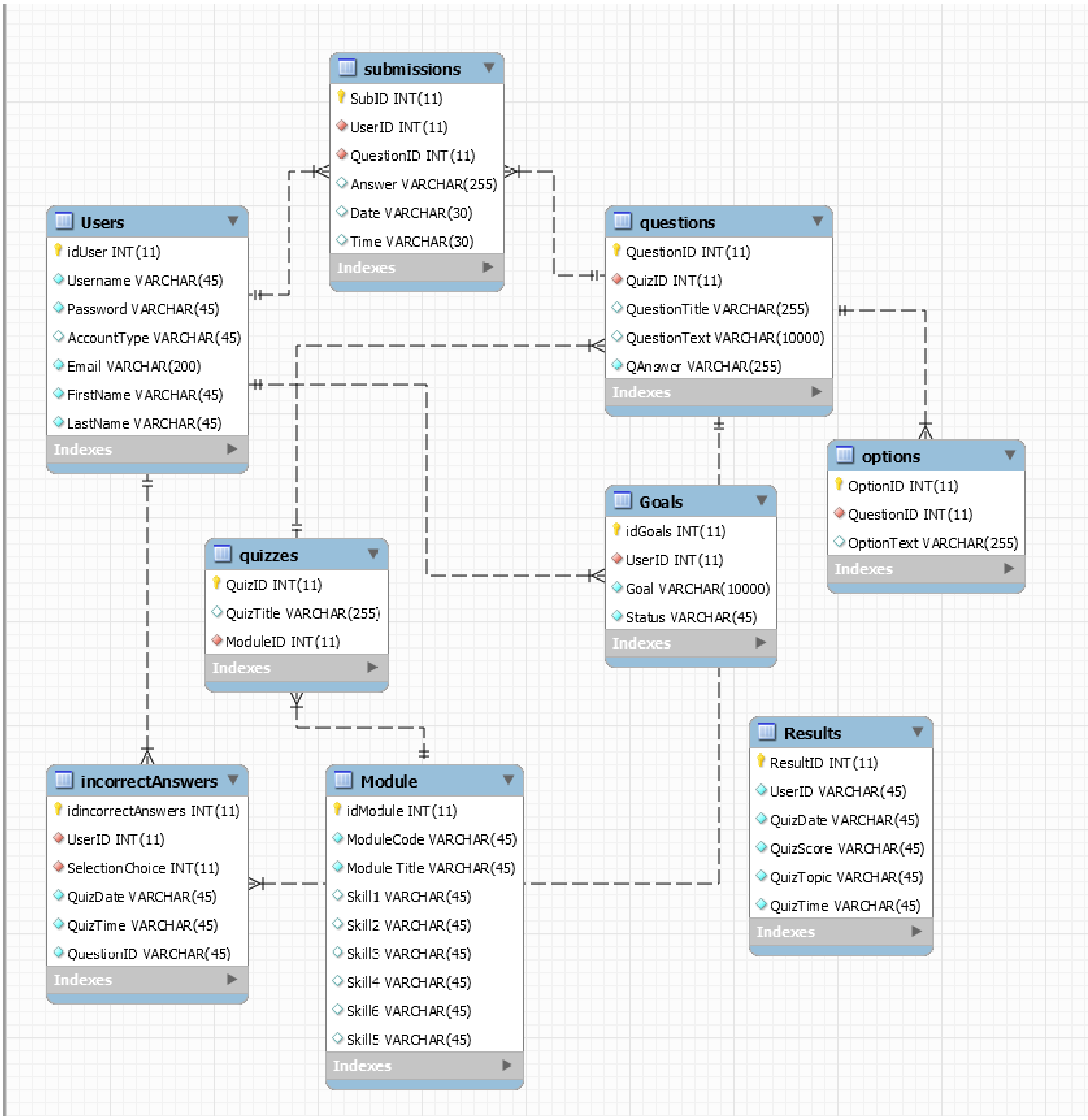


*Figure 31 Initial entity relationship diagram*

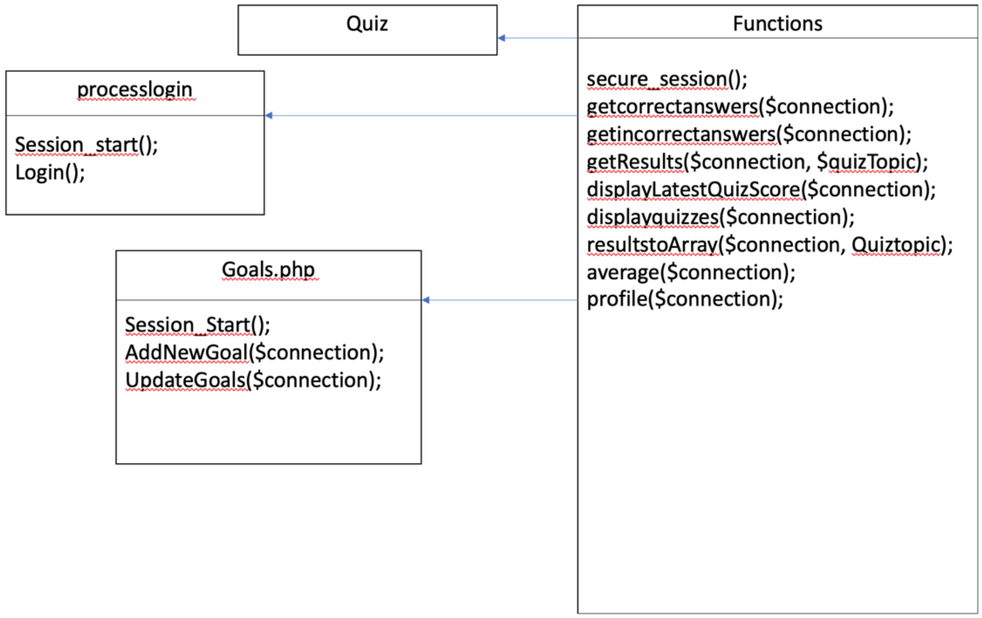


*Figure 32 System UML Diagram*

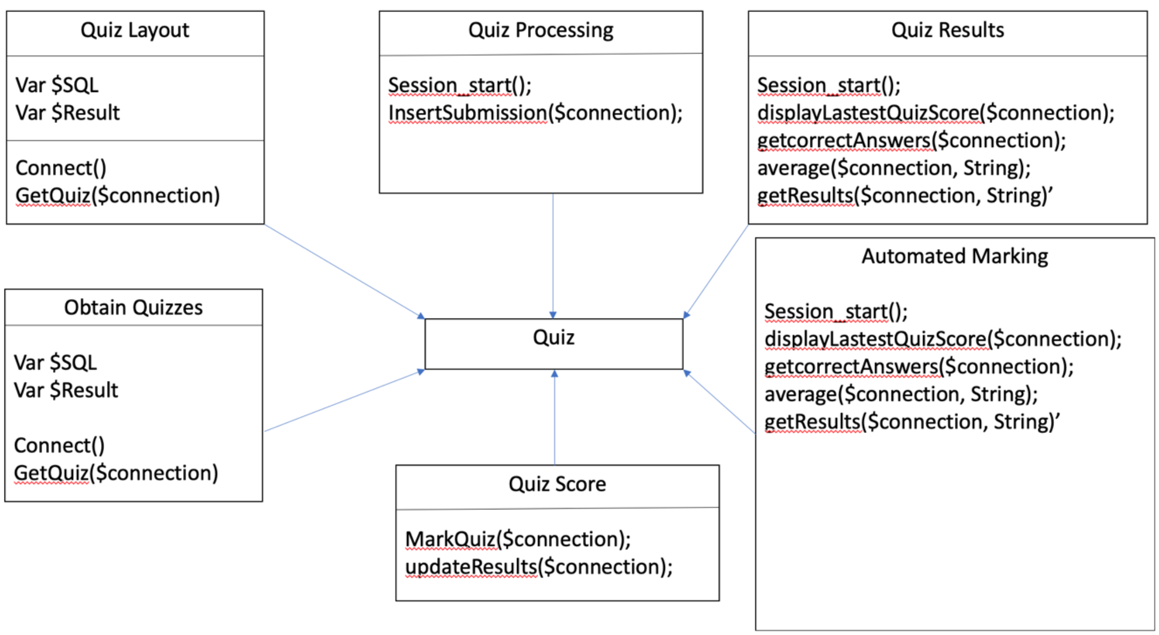


*Figure 33 Quiz UML Diagram*

The first step to ensure that personalisation (FR2) was available to students included the development of login pages, shown in Figure 34.

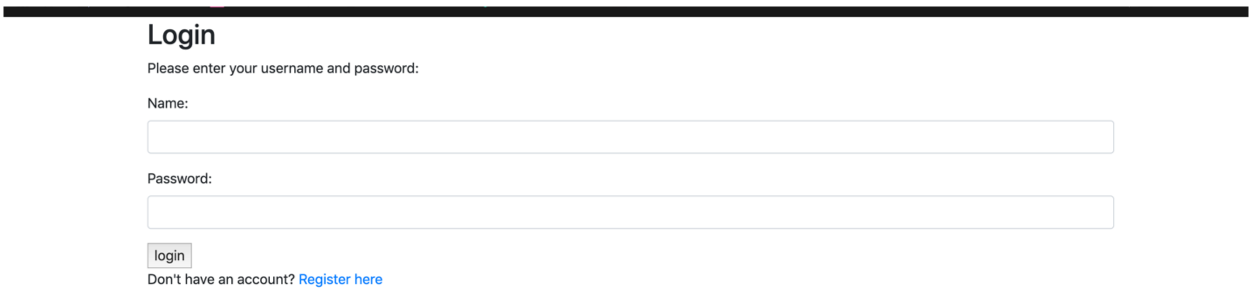


*Figure 34 CIA login page*

To fulfil FR1 (page 130), the content within the system was primarily taken from the lecture notes provided to the undergraduate module CSC1021 Programming 1. An aim of the tool is to have students use it as an additional guide to learn and understand the material throughout the course. Figure 35 shows examples of the content within the system. This figure also shows an adherence to NH4, which allows for recollection over recall, by using a code format reminiscent of the IDE used by students in their practicals.

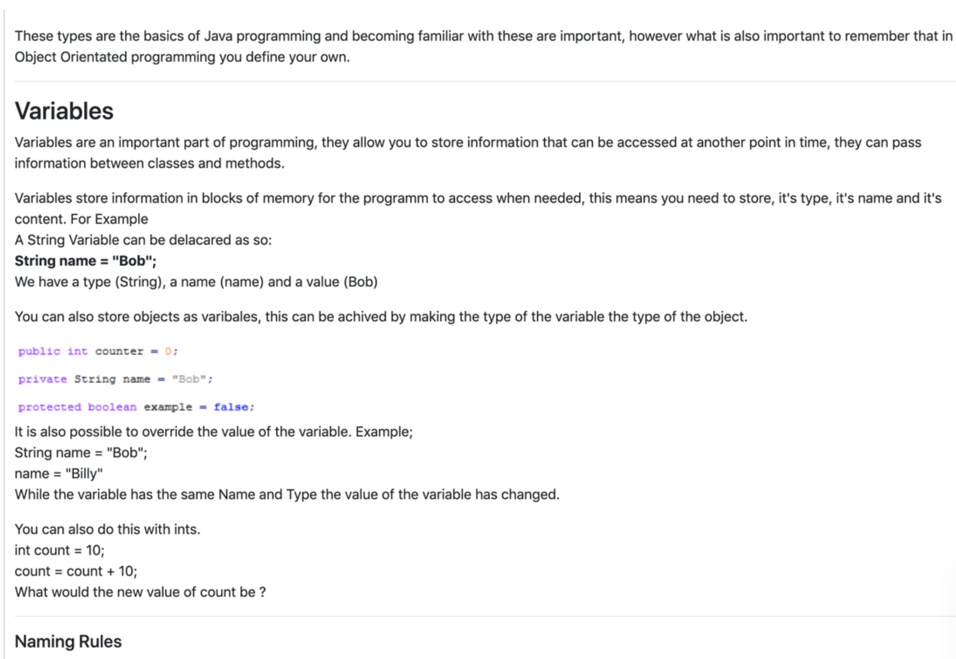


*Figure 35 CIA content page*

As the trial of the system is aimed at volunteers, the content and the results from the system could not contribute to the students' final. As the Ipsative system was the main aim of the investigation, it was decided that the “assessment” points used would be multiple-choice proof of concept questions. Figure 36 shows the assessment in this system which comprised quizzes using questions from an online resource called Sanfoundry (Sanfoundry, 2019). Peer reviewed questions were chosen so that there was quality assurance on the questions used and therefore the experimental aspects of the investigation could be contained to the Ipsative approach.

To implement FR3, multiple-choice quizzes were used to help examine students understanding of concepts such as Control Structures, Variables, String Handling, Methods, Arrays and Classes. The quizzes were broken down into sections based on the content from CSC1021 Programming 1, however they were not part of the summative assessment of the module. The difficulty of the quizzes ranged from fact recall to a requirement for a deeper understanding of the lecture material and questions which would require further research outside of what the system provided. To help ensure that there would be no confusion as to what quiz the student was taking, the quiz selection changed to a menu-based system where the user was able to select from a list of topics. Figure 36 shows an example quiz on the control structures topic.

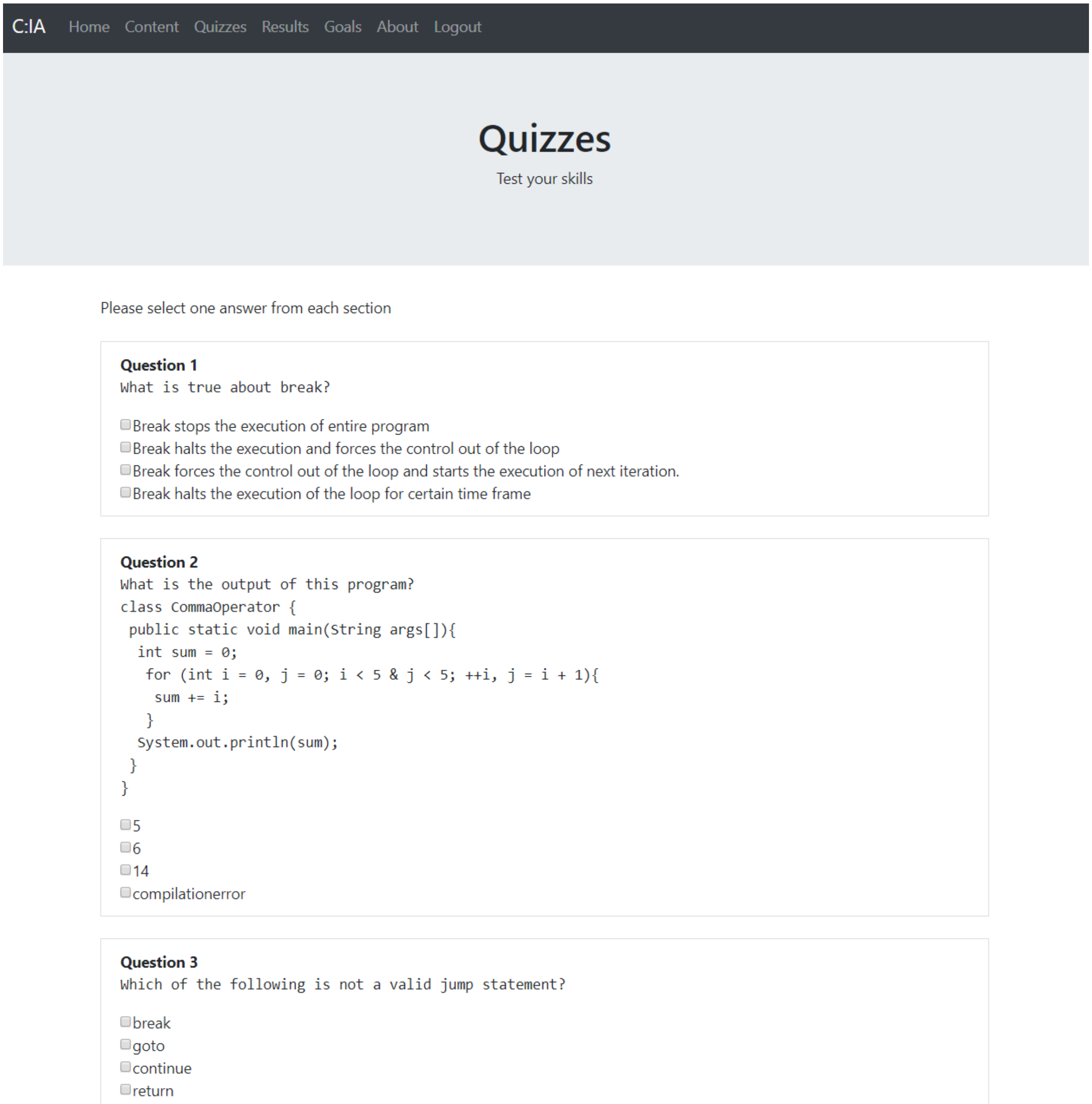


*Figure 36 CIA Control Structure Quizzes*

```
$num=2;//start from 2 as first is listed separate for 'action' attribute
$sql = "SELECT QuestionText, QuestionID FROM questions where QuizID='" . $QuizID ."'ORDER BY RAND() LIMIT 11";

$result = $connection->query($sql);
$row = $result->fetch_assoc();
$counter = 0;

if($result->num_rows > 0) {
    $temp = $row["QuestionID"];
    $tempText = $row["QuestionText"];
    $mcq = false;
   echo "Please select one answer from each section";
   echo "<ul class='list-group'>";
    while($row = $result->fetch_assoc()) {
        $counter++;
        array_push($identification,$row["QuestionID"] );
        echo "<b><li class='list-group-item'> Question ".$counter."</b>";
        echo "<p>" . $row["QuestionText"] . "</p>";
        $SQL = "SELECT options.OptionText FROM options where QuestionID ='".$row["QuestionID"]."'";

        $result2 = $connection->query($SQL);

        //Display the options from the first question.
        while ($option = $result2->fetch_assoc()){
        echo "<input type='checkbox' name='check_list[]' value='".$option["OptionText"]."'>". $option["OptionText"]."<br />";

        }
    echo"</li>";
}
echo "</ul>";
}
```

*Figure 37 Quiz generation code*

Figure 37 shows the code used to generate the quizzes displayed to the students. The SQL selects 10 random questions from the QuestionID table to display to the user. Once these IDs have been selected, a second SQL statements retrieves the text and possible answers for the students to choose from. Once the student has selected the appropriate answers using the form as shown in Figure 36, the selected answers are then marked, and the results are given to the student.

To achieve FR2, and FR4, a goal and results page were implemented. As explored in Section 6.4, to implement an Ipsative process the student needs to see their past progress as well as having the ability to develop specific targeted goals for future improvement (Hughes, 2017, pp7). Figures 38 and 39 show how students can see not only what questions they did not answer correctly, but also the correct answers using the 'correct answers tab'. The student is however advised not to look at the correct solution and instead mark this as a goal to improve upon before trying the quiz again.

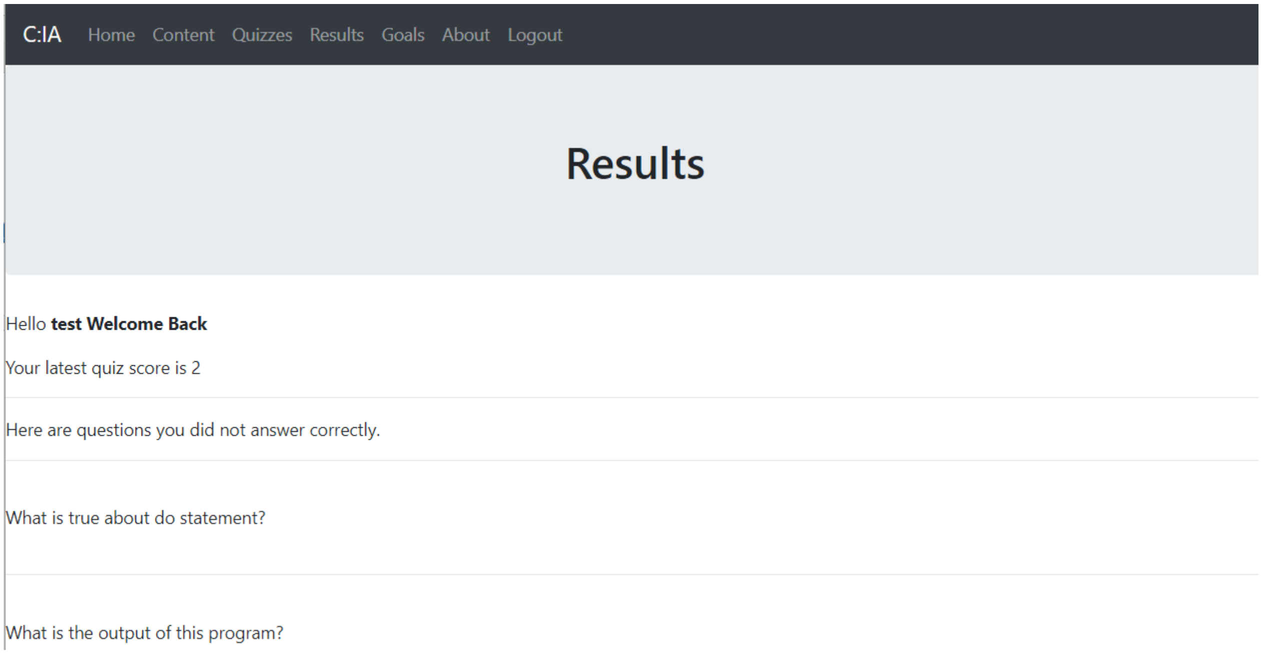


*Figure 38 Results page, incorrect questions*

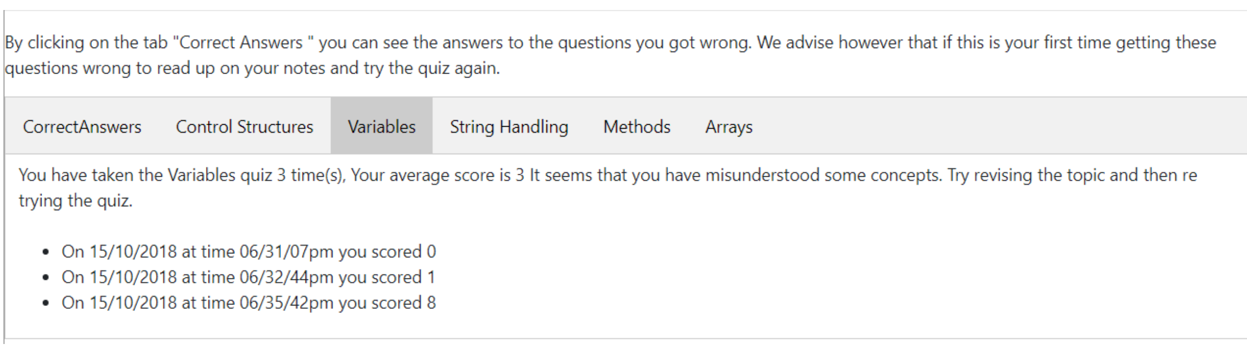


*Figure 39 Results page – Scores*

```php
while ($row = $queryResult->fetch_assoc()){
    $markingQuery = "SELECT QAnswer from questions WHERE QuestionID =".$row['QuestionID']."";
    $markingResult = $connection->query($markingQuery);
    $question = $markingResult->fetch_assoc();

    echo "</br> You selected ".$row['Answer']." The correct answer is ".$question['QAnswer']."";
    if($question['QAnswer'] == $row['Answer'] and !empty($row['Answer'])){
        echo "True";
        $quizScore++;

    }else{
        array_push($questionsWrong,$row['QuestionID']);
        $wrong ="INSERT INTO incorrectAnswers (UserID, SelectionChoice, QuizDate, QuizTime, QuestionID)
        VALUES ('"   .$_SESSION['idUser'].  " ', '"  .$row['QuestionID'].  "' ,'"
        .$_SESSION['QuizDate'] ."' ,'".$_SESSION['QuizTime']."','".$row['QuestionID']."')";
        $connection->query($wrong);
    }
```

*Figure 40 Quiz Marking code*

Figure 40 shows the PHP and SQL created for marking the choices that students have made on the quizzes. The first task completed is to compare the section the student made to the correct answer stored in the database. If this is the correct answer the user is informed that it was correct and their quiz score is increased. If the answer is incorrect, it is stored into the table incorrectAnswers along with the time and date of the quiz. These answers are then retrievable and comparable to the correct answers to help students identify what they need to further develop to increase their understanding.

An important aspect of Ipsative assessment is that of reflection and forward-thinking. Figure 41 shows the implementation of the goals section within C:IA which helps support the student in these skills. It allows students to regulate their learning by setting and tracking the progress of goals.

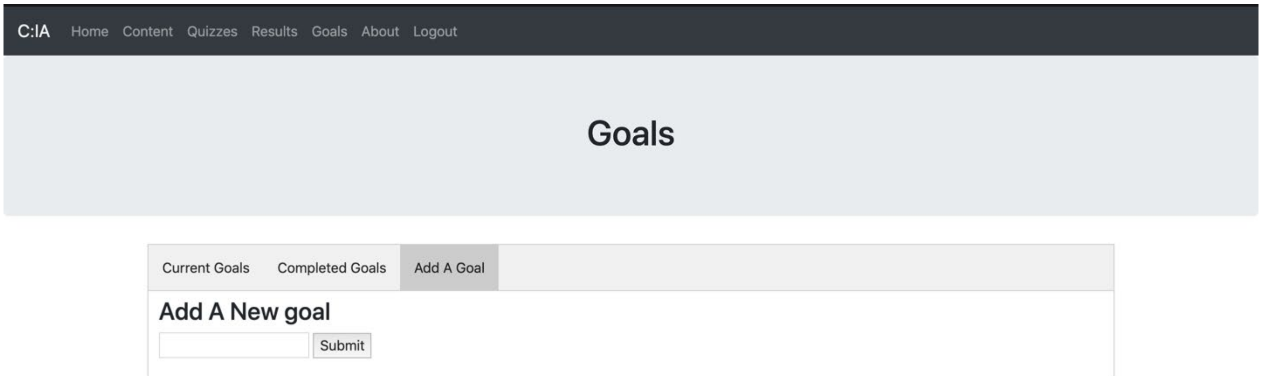


*Figure 41 CIA Goals*

This guidance can also be seen to a lesser extent within the homepage of the system. This informs the students about which topics they have covered, how many times they have taken the quiz and what their latest score is.

```
<?
include_once 'htconnect.php';
session_start();
foreach($_POST['check_list'] as $selected) {
echo "<p>".$selected ."</p>";

$sql = "UPDATE Goals SET Status = 'Complete' WHERE UserID = '".$_SESSION['userId']."' AND idGoals = ".$selected." " ;
echo $sql;
$connection->query($sql);
}
header('Location: ../PhP/Goals.php ');
?>
```

*Figure 42 Update Goals Code*

Figure 42 shows the code used to set existing goals as complete. On the GUI the student will select an already existing goal, which has an id of 'idGoals' as given by the system. The query then changes the goal to have a 'complete' status.  When students review goals in the future those with the 'completed' mark will appear under the completed tab.

Section 6.6 explores an initial investigation with volunteers from the module CSC1021 and the usefulness of the system in using Ipsative assessment.

### *6.6 User Testing*

#### *6.6.1 Student Testing*

The aim of the system trial was to investigate the level of ability students had when beginning to use the system and to find out if the Ipsative features of the system helped both develop these skills and increase their satisfaction. The system was designed to be used in parallel to the module CSC1021.

To understand the initial experiences of the students, the following questions were asked (see Section 3.2.5 for a full justification of these questions),

**Before system usage questionnaire:**

- What is your programming experience?
- What languages have you used?
- What is the most challenging aspect of Computing?
- With regards to programming, what areas do you feel are a strength?
- With regards to programming which areas/topics do you think needs improvement?

**Sample:**

The system was trialled with a group of volunteers from the CSC1021 Programming 1 module. In total, five undergraduate students agreed to take part in the study. Three of these students were female and two of these students were male, additionally one of these students had an accessibility consideration.

**Methodology:**

The volunteers used the system throughout the first semester of the stage one module. After the initial meeting where the primary researcher and the students met to discuss the individual needs and goals, students were to use the system independently. The students were expected to review their progress and update the goals sections allowing for self-development and reflection. After the completion of the module and use of the system, the following questions were asked of the students. (for a full exploration of the purpose and justification of these questions, see Section 3.2.5).

**Post System Questions**

1. Did you use the system? If so how?
2. How confident do you feel with programming?
3. Did the system help you learn the principles of programming?
4. How have you improved since taking the first questionnaire?
5. What aspects of the system did you like?
6. What aspects do you think would improve the system?

#### *6.6.1.1 Pre system exploration*

Information was collected from students to understand their experiences and skills before the initial use of the system. Student one stated that they had one year of programming experience through their studies in an EU country. Student two identified that they had taken programming courses in their GCSEs and A Levels. Student three and four reported that their programming experience was limited to the degree they were studying. From the perspective of this study all students will be classed as programming novices. While it could be argued that student one was no longer a novice, they were a Java novice, which is the primary language used within this course.

Question two asked students what languages they had used previously to the module,

- Student one identified that they had previously used C++ and C#.
- Student two identified that they had previously used Python
- Student three identified that they had used JavaScript, HTML and CSS.
- Student four Identified that they had used JavaScript.

Therefore, while student one and student two were not novice programmers, all students were novice Java programmers.

When answering question 3, "What is the most challenging aspect of Computing?" student one identified that program design was the challenging aspect. Student three identified that object orientation was the most challenging aspect. Student two and Student four however identified subjects other than programming was the most challenging part. Student two identified that mathematics was the most challenging aspect and student four identified that computer architecture was the most challenging.

Following this, students were asked with regards to programming, what did they find a strength? When discussing their strengths students were a lot more specific. The strengths identified were as follows:

- Student one: Implementing a premade design.
- Student two: Creating program designs.
- Student three: Loops and conditional statements.
- Student four: program design.

Finally, when asked to identify which areas the students felt they needed to improve in, they responded as such:

- Student one: Program design.
- Student two: Expectation Handlings.
- Student three: Objects and Classes.
- Student four: Program Design.

Identified by the students, the skills found most accessible were the skills involving syntax and memorisation, the skills of the lower levels of Bloom's taxonomy. The skills that were identified as a challenge were the more difficult but important skills that a programmer needs, such as code comprehension, problem-solving and program design (Bailey and Stefaniak, 2001).

#### *6.6.1.2 Post system exploration*

The first question that was asked of the students in the post system interviews was if the students used the system. The students reported that they did not use the system as originally intended, which was a consistent use alongside the module material. The system was instead used in the week leading up to exams, and the students primarily used the system as a revision tool.

Student One - *"Yes I did, I did use it when you just sent the link and then about a week before the exam as a refresher."*

Student Two – *"Yes, I did use the system. I used the system during the revision process, so sort of after lectures. ….. partially because I was curious to see how the system would work…. but also because in terms of programming, aside from making up your own exercises, there is very little in terms of revision materials."*

Student Three- "*Not very much. Um because I generally just went through the notes because they had similar examples. I used the system every now and again, but half of the time I forgot what the link was to it."*

Student one, two and three gave explanations of why they did not use it very much, whereas student four only reported they did use the system. A lack of engagement was to be expected, as using the system adds to their workload. The lack of engagement within this study gives further evidence to the mark driven nature of students (Wilson 2006). When completely removed from summative assessments as the work in this system was, there was still a link to potential mark gain and with a focus on the summative assessment. The main concerns regarding the system was how it can benefit their assessments as opposed to the skills that students could still develop. It is highly probable that this is due to the voluntary nature of using the system and the fact that it was extra work for the student.

When asked why the system was not used, student three stated that the system was too similar to the module content and had similar examples. The lack of use due to the similarity suggests that the quiz aspects were not considered to be important to the learning experience until the exam period started.  The comments from student three contradict Moscinska's view that the material used in external computer-assisted technologies should be as close to the original resources as possible (Moscinska, 2008). The negative view of the similarities could be due to the way the system was pitched to the students, it was pitched as an addition to the module content, therefore, should have had variation.

#### *6.6.1.3 Usefulness in learning Java*

When asked about the usefulness of the system in helping to understand programming, there was a mixed response within the students. All students reported that the content was more digestible than that of the lectures, in that, the smaller, more directed chunks of information helped summarise topics and give quick access to them. Student two reported that the smaller bite-size chunks of information helped them to learn the theory and combined with the quizzes they were able to put into practice what they learned. All students reported that the quizzes were very difficult in parts, and therefore, the students felt like they could not attempt them. The final note from this section was the comment from student three that while the system helped teach concepts such as arrays and loops, it did not help them put the rest of their skills such as code development into practice. The response to the difficulties of the questions is perhaps further evidence of students using a strategic form of learning and dismissing content that was considering to be too difficult (Ferrell, 2012).

The student interviews also identified that a feature that needed to be added was to take the questions the student got wrong and save them for reflection. While this information was stored it was not accessible to students past their initial attempt at the quiz. Student four reported that rather than looking at the answers, they wrote the question in a notebook to refer to later.

These interview responses show that learning in an Ipsative form can prove beneficial for student learning and can help students develop self-assessment and self-regulation skills.  However, for Ipsative assessment to be beneficial to students, there should be more than one iteration of previous results. This iteration is also advisable as sometimes staying at a consistent level is a challenge for students. This also shows some evidence that students are not taking a surface learning approach to their studies (Hughes, 2017b, pp32).

The self-guided goal feature was reported to be useful in the sense that it allowed the students to track their progress and what they needed to do even when there was time between each system use. Student four specifically commented that - *"I think the goal feature was really good, it allowed me to keep track of my progress because I didn't just do all the topics on one day. So you know coming back two days later I probably forgot what I did before. So, having that goal and knowing what I did before, and I'm going to this, that helped me a lot. I'd say the home screen itself yeah that was good, knowing which I haven't done, and the quizzes of course."*

It is hoped that the goal feature helped reduce the effect of modularisation by helping students identify and track specific target goals in other assessments. The goals aspect also helped develop a student's self-regulatory skills and enabled them to plan their own learning, putting their assessment literacy skills into practice. This can be seen in the post system questions where question four asked the students how they had improved since the initial interview.

Student two – *"I think that the system was, interesting and by that what I mean is, I think it helped reinforce what I knew. Whether I could say it helped me improve, necessarily. However, there was a couple of questions where being unsure of what I was doing, getting them right or getting them wrong, helped me realised by experimenting, thinking if I put that in to get the answer wrong. I think in that regard it could have helped me improve because it could have helped me re-enforce how different concepts worked. I would have said, um so yes to an extent it probably did help me improve I would think."*

What was promising, when asking students to explain what skills they had improved upon since the initial interview, was the specificity that students now answered with. In Section 6.6.1 (page 162) the students each identified skills which they wished to develop. Table 32 shows the initial skills a student identified, the identified improvements the student had made, and the future work the student needs to do.

| Student | Initial weakness | Skills improved | Improvements still to be made |
|---|---|---|---|
| 1 | Program Design | Collections framework.<br>General object-orientated concepts | Understanding of exceptions |
| 2 | Exception handlining | General programming skills | Abstraction and Inheritance |
| 3 | Object Orientation | Applying the programming concepts to the 'big picture.' | Debugging and Abstraction |
| 4 | Implementation of code | Lists<br>Immutable lists<br>Arrays | Understanding of Exceptions |

*Table 32 Student Skill identification*

While there is little evidence that the system helped to strongly re-enforce the concepts of programming, what can be seen is an increase in a student's ability to self-assess and self-regulate. In the initial questions, students identified broad topics with multiple tasks within them, showing a lack of understanding of the individual steps needed to complete the task. In the post questions students have identified not only where they have improved, but specific, actionable areas in which they can further improve, a core aspect of forward-facing Ipsative assessment (Hughes, 2017, pg7). When identifying further improvements, students identified the two final topics with which they had covered, Abstraction and Exceptions. While this shows an understanding of which topics they need to improve upon, only student three identified a practice (debugging) that was not the final most challenging topic on the module.

RQ3 investigates if Ipsative assessment could help ameliorate the issues students face with assessment and feedback. It is promising to see that the features that were highlighted by the students as being most useful were those of an Ipsative form.

The students reported that for topics such as arrays and loops, their confidence improved. However, for the more practice-based topics such as program writing and debugging, they did not feel that there was an improvement in confidence. This also shows the Ipsative form of the system in progress as students can see what skills and areas they have made improvements in but also what the student needs to improve upon in future work.

#### *6.6.1.4 Future Developments of C:IA*

The final question that students were asked was what additional features would be beneficial for C:IA. Each student asked for a native mobile application to be developed, beyond this however each of the students who took part in the interviews had a different response:

- Further links between the goal system and the content, e.g. if a goal is added about learning loop structures, the system should link you to that topic.
- The difficult questions that went beyond the course material were demotivating for one of the students using the system. Therefore, more focus should be placed on keeping the content within the course boundaries. A ranking system for the questions could also be developed.
- A setting could be included in these sections to filter out that which students do not wish to see or questions that are marked as too difficult.
- A more direct link between the topics and the quiz is needed, as currently there is not a direct link.

### *6.6.2 Staff responses to C:IA*

To gather more data regarding Ipsative assessment as a concept as well as the benefits of Computing: Ipsative Assessment, staff at Newcastle University were also interviewed. In total, five lectures took part in semi structured interviews with the following questions (for a full justification see Section 3.4):

1. As a concept, do you think Ipsative assessment would be beneficial to students?
2. Would a system such as C:IA help a student gain an understanding of the concepts of programming?

3. Does C:IA provide a clear region for self-regulation, reflection and growth? Please expand how/ how not?
4. When exploring the quiz sections do you find the questions to be suitable for first year students? Are they too hard? Impossible for the students to do? Or manageable but challenging?
5. What do you think the benefits of such a system would be?
6. What do you think the drawbacks of such a system would be?
7. What features do you think should be added to the system?

***6.6.2.1 Ipsative Assessment as a concept.***

When discussing Ipsative Assessment as a concept the benefits and drawbacks were discussed for both the students and the staff perspective. In the responses there were two themes identified as benefits to students, the first was to increase the independence of the student thus developing their ability to complete independent learning. The second was that the students would be able to see the progress they were making / identify where they were not progressing, and therefore see the value of the tasks that they were completing.

Staff 1 *"I think it would be if the students, especially for independence"*

Staff 2 *"Those who want to sort of progress by doing etc they will benefit because if they, If the feedback relates to what they did last time, they will, they will presumably demonstrate continuous improvement in some way"*

Staff 3 *"Yeah, definitely……. It's a reflection on their personal knowledge instead of uh I suppose benchmark against other students"*

Staff 4 *"They would be able to see the actual value of doing these sorts of things."*

Staff 5 *"I think as a student it would be good if my areas of improvement were highlighted and that future assessment would assess me on things that I am struggling with."*

These responses from staff identify the core benefits of Ipsative assessment. One identified benefit was the potential learning gain made by students and that the structure of Ipsative assessment supports the development of a 'deep learning' approach through the use of reflection (HEFCE, 2016, Hughes, 2017). For example, Staff 4 identified that through using Ipsative Assessment it may help students see the value of the work that they are completing, therefore be more engaged with it. Section 2.3 (page 14) identified that if a student cannot see the purpose of assignments and how they connect, this may cause a disengagement from the work (Ferrell, 2012). Therefore, if Ipsative Assessment can help students see the purpose of the tasks that they are completing, it may help students become more engaged.

When considering the drawbacks of Ipsative assessment for students each staff member identified a different issue.

Staff 1 *"I would assume like employers, would also still prefer that kind of like, here's a mark based on what level you're at."*

Staff 3 *"Maybe they could get a bit more bit caught up and kind of the the marks for it and kind of think, more about getting the answers right, than the actual understanding behind it."*

Staff 4 *"Students may not see the big picture of their overall assessment experience/learning objectives."*

Staff 5 *"I can imagine that providing diminishing returns when you get to larger groups of students."*

Staff 1 identified that Ipsative assessment requires that the student be self-motivated and engage with the material. They also identified that it would be difficult for employers to use, and that it would be much easier to identify an applicant's skills from traditional summative assessment.

Staff 2 discussed that while Ipsative Assessment may help students who are progressing, it could be demotivating for students who do not progress. As identified in Section 6.3 it is important to consider that staying at a consistent level can also be a positive as this can be challenging if the student is struggling with the topic (Hughes, 2017b, pp32).

Staff 3 identified that similarly to summative assessment, students may be more concerned with increasing the marks / quantifying the learning gain as opposed to increasing those marks through the development of their understanding. This was also found when considering student responses in Section 6.6.1.3 (page 164) where students linked the use of C:IA to potential mark gain. Staff 4 identified that there was a concern that students would not see the bigger picture of where the increments of Ipsative assessment fit. Finally Staff 5 identified that there may be a concern regarding fairness of the assessment, as each student will have a different distance to travel upon entering the course and it would be difficult to translate this to effective marks. However in Section 2.6 (page 37), it was theorised that a potential reason for an identified lack of fairness in the NSS results (70% to 77% result, where 80% is considered a positive benchmark) could be due to a lack of acknowledgment of the students learning gain. Therefore, combining Ipsative assessment with summative assessment may be able to increase student satisfaction in this area.

When considering the benefits of Ipsative Assessment for staff, all interview responses had a similar theme. The main response was that it could be used as a diagnostic tool for class knowledge but to also identify who is engaging with the material and developing their independent learning skills. Similarly, there were common themes in the responses to the negatives for staff. Staff 1, 2 and 5 all reported that Ipsative assessment would be difficult to achieve for large class sizes due to the workload that Ipsative Assessment introduces.

Staff 2 *"Whether somebody would actually have the time to go back through 300 students' previous assessments or not is another matter"*

#### *6.6.2.2 Usefulness of C:IA in learning programming.*

The aim of C:IA was to assess novice programmers in their programming skills, and as identified in Section 2.2 it is important that assessment should be *for* learning as much as it should be *of* learning. It is therefore important to understand if C:IA and the Ipsative process could help develop students programming skills. In Section 6.6.1.3 students reported from the use of the system that it was useful in learning the concepts of Java (the language used in their programming module), it is promising that staff also agreed that C:IA would be helpful.

Staff 1 *"I mean, it has to be embedded within a module"*

Staff 2 *"I would say so yes. Yes, I think it's. It's easy to navigate and it can obviously be extended."*

Staff 3 *"Yeah, I think so. I mean you could use it sort of, especially your tool so you could use it sort of like as a diagnostic thing to find out areas that they need a bit more attention on so they might understand some concepts, but they might not understand others."*

Staff 4 *"I would say probably would help with that, certainly. I would say it wouldn't really be a substitute for things like a lectures and that sort of think but certainly be tutorial style things could help with their learning in practicals."*

Staff 5 *"With enough supporting material to go alongside CIA, such as extra reading."*

It was also noted by staff 1, 4 and 5 that for C:IA to be useful it needed to be used with additional external sources, such as being embedded within a module or extra reading. This was not a comment made by students, where they emphasised that the content was more digestible than in lectures. However, it is important to note that these students used the system alongside their programming module, so received the additional benefit mentioned by staff. Additionally, C:IA was not designed to replace all teaching, it was designed as an assessment tool to automate some of the more intensive workload introduced by Ipsative assessment.

#### *6.6.2.3 Reflection, regulation and growth in C:IA.*

As discussed in Section 2.6 a large component of Ipsative Assessment is a student's ability to reflect on and regulate their work, therefore, staff were asked if they thought this functionality was achieved. Staff 1 identified that while C:IA provided a good region for reflection, regulation and growth there was a potential for this to be demotivating to students. This was due to their being no material on how to develop good goals, and from this, students can create bad goals. The example used was a student wanting an average of 10 while already scoring 0 on a quiz, this goal is therefore impossible. Evidence within the literature shows that to engage students, tasks and goals need to be actionable, this can be seen in the development of Blooms Taxonomy to include actionable stages (Coffee, 2008, pg 19) and Whalley's use of actionable test cases (Whalley, 2011, pg33). Due to this it is important that there be some guidance for students on how to create actionable goals that have tangible results.

Staff 2 identified that there were areas for reflection, such as reviewing previous results, but it depended on the student and how they used this system. Similarly, staff 5 expressed concerns that while the goals would be useful it requires engagement from the student. The engagement of students were also identified when exploring the quizzes of the system, where staff one identified that the usefulness (and difficulty) of the quizzes were dependant on the student and how they engaged them.

*Staff 1: "I think they're good. I mean, as I said, like so it depends on how they go about doing them."*

The dependency on student engagement is to be expected as in Section 2.6 it was also identified that for Ipsative assessment (and self-regulation) to be successful it required students to engage and be self-motivated in their learning (Housand & Ries, 2008; Eugenia, 2018; Tai and Adachi, 2019) (pg 39).

#### *6.6.2.4 Difficulty of quizzes.*

In Section 6.6.3, students identified that certain aspects of the quizzes were too difficult and therefore demotivating. Due to this staff were asked to comment on the difficulty level of the quizzes and whether they were too hard for the expected level. The response from all staff members were that the quizzes were manageable but challenging. There were some additional follow ups from specific staff members. For example, staff 1 further emphasized that difficulty of the quizzes and the usefulness of difficult questions depended upon the student's mindset. Staff 2, 3 and 5 identified that some of the questions given in the quizzes would be difficult for students at the beginning of a module, however they would also expect the questions to be manageable for students who were nearing the end of the module. This could present a problem however due to the fact that if students find the work given to be too difficult this could cause motivation issues and result in a disengagement (see Section 2.5.2).

Staff 2 *"I would say manageable but challenging."*

Staff 3 *"I think about the right level. I would see it manageable but challenging."*

Staff *4 "I would say that most of them seemed fine it may have been a bit brief, so perhaps having a bit more context would be useful."*

Staff 5 *"I think its manageable."*

***6.6.2.5 Benefits of C:IA***

The main benefit of C:IA identified by staff was to use the tool as an area for additional practice, as having content within one place (as well as providing additional material to the lectures) would help student develop the skills needed.

Staff 2 identified that the benefit of having such a system would be to give the students a consequence free way to work together and get questions wrong. This would also allow students to experiment with concepts they were not confident with. This was important as staff 5 identified that there needed to be a practical aspect to programming for the student to gain an understanding of the material (Ala-Mutka, 2005; Rum et al, 2016; Bradley 2020). The practical aspects of Computer Science are practiced at Newcastle University through the use of practicals which can be seen in Section 3.2.4 and Appendix C.

Staff 1 *"I as I said I like the system and it's like putting it, it's just putting it in one place type of thing."*

Staff 2 *"It's not actually assessed, so they can work together on problems without getting an assessment irregularity."*

Staff 3 *"For students, yeah, it lets them reflect on kind of their understanding and then it helps them in a structured way to improve their knowledge."*

Staff 4 *"So I think it certainly, in sort of practical setting or questions. It's probably a nice structured way for them to actually learn something to top up their skills or their Knowledge."*

Staff 5 *"Practice. Always practice. When you're programming, especially novice programmers, opportunity to practice is not something that comes straight away to students, some students don't recognise programming as a skill set that you need to get good at over time and that you can only get better, you can't get better by reading about java."*

#### *6.6.2.6 Drawbacks of C:IA*

Staff members were also asked to identify possible drawbacks of using the system, unlike the potential benefits of the system, each staff member identified a different potential drawback to the system. While the responses were different there were similar themes expressed, (i) the system required self-motivation to use and (ii) the impact of student perceptions regarding marks.

Staff 1 *"If there's like 20 of them and ten of them (quiz questions) were easy and it just so happened that every you know one person got the 10 easy questions, then that's probably unfair to someone who did the 10 hard questions."*

Staff 2 *"If somebody was repeatedly getting low scores. It might be a bit demoralizing."*

Staff 3 *"They might just be a bit more focused on kind of the quizzes then I suppose, really digesting the content."*

Staff 4 *"I would be concerned that students might think it's a substitute for actually attending the lectures."*

Staff 5 *"It's another thing to, it's another system, let me be more clear, it's another system that you have to keep on top of."*

The self-motivation of students has been a consistent theme throughout the responses of staff members to the interview questions. This is promising from the perspective of using assessment to facilitate learning.

The responses also identify some of the mark driven nature of students at higher education, as staff identified that getting low marks may be demotivating to students. It was also identified that students may focus more on how to improve the mark as opposed to improving their understanding of the material, therefore taking a surface learning/ strategic learning approach (Struyven et al, 2005) .

Staff 1 identified that Ipsative assessment may not work well with Summative assessment and that there would need to be a way to ensure fairness, in that one student did not get 10 easy questions and one student did not get 10 ‘difficult’ questions. This balance is a crucial aspect to get correct as fairness and validity are a important aspect of assessment (Hughes, 2011)

Staff 4 expressed concerns that students would see the system as a replacement for attending lectures as opposed to an additional resource. Another concern expressed by staff 5 was that C:IA would be another system for students to keep track of on top of the other systems that they use, which may cause a higher workload and therefore a disengagement.

#### *6.6.2.7 Additional Features*

Finally, staff were asked about what additional features they thought would improve C:IA, the identified features were as follows:

- A get help button
- Previous iterations of quiz questions
- Executable code

Staff 2 and 4 identified that a get help feature for when students are struggling to answer a question would be helpful. Both suggested that a call for clarification on the questions would also be helpful.

As also identified by students, a feature to view previous iterations of the quizzes would be beneficial. This is a core component of Ipsative assessment and was not identified to be missing until the user trial of the system, this is further discussed in Section 8.5.

Staff 1 and 5 identified that the design of the system needed to be improved to make navigation easier. This was also identified by student two, who did not find a feature of the system.

Finally, staff 4 suggested that a feature to be added would be to have regions for executable code. However as mentioned in Section 6.4, not including executable code was a design feature aimed at improving a student’s code comprehension and design as opposed to code syntax. Staff 3 reported that they could not think of additional features that they would add.

Staff 1 *"So, for example the content page I thought that was your content. I didn't realize I had to click on Java in order to then get the content. Which now makes sense, because obviously you wouldn't do it just for Java."*

Staff 2 *"Some way of calling, you're getting help that could maybe speed things up."*

Staff 3 *"No, I don't think so."*

Staff 4 *"A way to actually execute the code so people can actually try say write the method and then compiling it."*

### *6.6.3 Interview discussion*

Computing Ipsative Assessment was designed to help answer RQ3, how does ipsative assessment compare to current practice and can it be used to ameliorate issues identified by students?

From the student interview responses, there were three key points to bring attention to.

1) The strongest feature that students found beneficial was the Goals Setting page. As this was the primary form of Ipsative assessment and self-regulation for the students, this was a positive result. The ability for students to see not only their current plans but also what they have achieved throughout their time using this system was a motivation and helped them to develop a deeper understanding and encourages a deep learning mindset (Ott et al., 2016; Hughes 2011; Rum et al. 2016).

2) Students reported that while the system was good at helping with code comprehension and expanding their abilities, it did not help with program writing. This was a good sign as it was the primary goal for teaching within the system to focus on developing comprehension and design skills of the students, not syntax generation. While Lajis states that a student needs to learn the basics of code writing before learning code comprehension, this is not a concern with C:IA as the intention is for it to be supplementary to the module where students learn this (Lajis, 2018). The responses show the development of the key skill of code comprehension (Bailey and Stefaniak, 2001), which is an important step to mastering the ability to code. However, the focus on code comprehension as well as some of the questions being perceived as too difficult was a negative point for the students. Robins identified that when students found work became too difficult, they can become 'stoppers'. When students adopt the 'stopper' attitude or a 'fixed mindset', they begin to target their learning and discount material that they believe to be too difficult to master, locking

themselves out of additional skill sets (Dweck et al., 1998; Robins et al., 2003). The difficulty of the questions is an additional reason that C:IA in its current form was not suitable as a revision tool. The structure of the code comprehension questions, as opposed to basic syntax designs, and the difficult questions in the quizzes were designed to push the student beyond their comfort zones. These functions were not designed to cover the basics as this was assumed to be completed within the module CSC1021. Therefore, using this material which went beyond the scope of the module and the assessment could lead to disengagement and dissatisfaction.

3) Finally, students reported a problem with the material being too similar to the course notes. This was a decision made to keep the course material as similar as possible, as suggested by Moscinska (2018). Students reported that the course notes were easier to use and did not have the issue of the difficult questions when revising.

The staff responses however identified different considerations to that of students, they were as follows;

1) While the concept of Ipsative assessment may be beneficial to students, the actual impact would depend on the students themselves. Staff discussed the concept of "stoppers" and "movers" (Dweck et al., 1998; Robins et al., 2003), it was discussed in both Section 6.6.2.3 and Section 6.6.2.4 that the mindset of students, and their resilience to challenge, is a large factor, this was also shown in the student responses as discussed previously. A further concept discussed however was that by giving the students a consequence free space to work, they may focus less on "beating the mean" (Wilson, 2006) and more on understanding the material.

2) C:IA was identified to be a useful tool if used with other materials as students would need further support to learn the material and to ask for further clarification. Additionally, there was concern that students may approach the system as a replacement to the lectures.

3) Finally, a benefit of C:IA and Ipsative assessment as a whole was to give the students a region to practice their programming skills. This will benefit students as computer science is a practical subject that requires students to have areas attempt the theory that they are learning in lectures, this can be seen by the practical labs sessions often seen in computer subjects (York 2003, Daly 2004). Additionally, Bailey and Stefaniak identify that the top

> skills needed by a computer scientist were very practical based, (Bailey and Stefaniak, 2001, page 31)

This experiment was designed to answer RQ3. While this system trial does not supply a definitive answer, it does show that the Ipsative process can help students to overcome some of the issues identified in chapter 4 and 5. The responses show an increase in the application of practical assessment literacy. The students were able to identify which areas they needed to improve upon and regulate their own learning to achieve this. It shows the development of skills beyond that of the module, and that students explored more aspects of code comprehension than code generation. There was also no identification of negativity within the system or its feedback. The system encouraged students to try and explore areas until they reached the difficulty increase, at which point students disengaged. Ipsative assessment assists with developing a student's ability to identify where their skills can be applied. Students said that the Goals and Homepages sections were beneficial to identify the skills they needed to develop and how these skills can be used elsewhere. This may help reduce the impact of students feeling that feedback does not give much future direction or does not help identify where the student made mistakes, as identified through the AEQ (Gibbs and Simpson, 2003) in Section 4.2.2. Furthermore Staff confirmed the potential usefulness of C:IA and Ipsative assessment to develop the skills of students and to reduce some of the issues identified (such as the mark focused nature of students) however it required a lot of support and resources from staff members.

### *6.7 System Evaluation*

As part of the initial design exploration of C:IA a set of seven system evaluation criteria were created, C:IA will also be evaluated using these criteria. Table 33 shows the evaluation criteria and how they align with the requirements of the system.

C:IA met the first criteria by creating small selections of content for students to learn Java with. It encouraged experimentation by providing space for students to think about how different solutions and code snippets would work together. Student two in the post system questions identified that they experimented with the bits of code they were using within their chosen IDE.

| Evaluation number | Evaluation Criteria | Requirements number |
|---|---|---|
| 1 | The system needs to teach students how to code in Java. It should have opportunities for the student to be creative and experiment. | FR1 |
| 2 | The user needs to have personal learning experiences. | FR2, NFR4 |
| 3 | The system should encourage and assist the student with reflection and regulation of their own learning. | FR3, NFR4 |
| 4 | Feedback given to the student should inform the student why they were wrong, and what the correct answer should be. | FR3 |
| 5 | Must allow for the implementation of Ipsative forms of guidance, such as adding goals and viewing past achievements. | FR4 |
| 6 | Must be accessible across multiple platforms via the use of web development methods. | NFR1, NFR2, NFR5 |
| 7 | The system should allow for the storage of data for long term usage. | FR2, NF3 |

*Table 33 Evaluation criteria and requirements*

Evaluation criteria two and three are met through the systems homepage, which informs the student what areas they need to improve in and how they did in the last quiz they took. This section informs the students what quizzes they have not completed and also what they have not done well and what they should study to improve.

C:IA does not meet evaluation criteria four, as while the system successfully informs the student of what the correct answer should be to a question that they get incorrect, the system does not state why the answers were wrong. Implementing a more specific identification of why a solution was incorrect is discussed in Section 6.7.3.

Evaluation criteria five was met through the implementation of the goals section. The goals section in addition to the results section assists the student in identifying exactly what areas they need to improve in and to plan their learning accordingly. The goals section containing all previously completed goals also reduces the negative aspects of the feedback by showing how they have improved throughout the use of the system.  The system successfully works across all web based platforms and mobile devices meeting evaluation criteria six. Finally, evaluation criteria seven was met by connecting to the database supplied by Newcastle University.

### *6.7.1 Review of system functionality*

The system was developed using HTML, CSS and PHP as well as a bootstrap plugin (Bootstrap, 2011) for the design aspects. The system was easy to navigate and transferred onto a mobile device with ease. All functional requirements were achieved within the development of the system, FR1 was met by implementing material from the CSC1021 module. Personal login details which allowed students to track progress and store quiz results met FR2. The implementation of quizzes allowed for the completion of assessment points, and from this the students were able to set goals based on these results, therefore fulfilling FR3 and FR4.

There were some issues using images on the mobile version, this, however, did not affect the content of the system. Problematic design choices lead to there being more user interaction than needed, for example, clicking an extra link to go from the content to the quiz section. Student B identified that due to their accessibility needs, they missed core features such as showing the correct solution when theirs was incorrect. This suggests that better placement is needed for the items within C:IA and that more accessibility functions be could included e.g. features such as further highlighting of sections and audio transcriptions.

The primary form of teaching content within C:IA was adapted lecture materials from the module CSC1021. This approach was taken due to Moscinska and colleagues (2018) stating that the content of computer based assessment and teaching should remain as similar to the source material as possible. However, in the interviews, students reported that the similarity to the course discouraged further use of the system. A further aspect that students struggled with was the material that went beyond the course, where the students needed a further and deeper understanding of concepts.

```
class multidimention_array
{
    public static void main(String args[])
    {
     int arr[][] = new int[3][];
     arr[0] = new int[1];
     arr[1] = new int[2];
     arr[2] = new int[3];
     int sum = 0;
     for (int i = 0; i < 3; ++i)
     {
          for (int j = 0; j < i + 1; ++j)
          {
             arr[i][j] = j + 1;
             for (int i = 0; i < 3; ++i)
             {
                 for (int j = 0; j < i + 1; ++j)
                 {
                     sum + = arr[i][j];
                 }
             }
          }

        System.out.print(sum);
     }
    }
}
```

*Figure 43 Complex Question Example*

Figure 43 shows an example of a complicated question given to the students. For this question the students were asked to select the output the code would give. While this does not introduce new concepts to the students, it does explore the concepts in a more complex way. In CSC1021 students are introduced to the concepts of arrays, and at the point of using the system will have experience implementing arrays. The level of depth shown within this example would be expected within their next programming module and was beyond their expected competencies.

### *6.7.2 User testing*

The experiment of trialling C:IA had limited success. While low numbers of volunteers were always a risk in a trial of the system, five students initially volunteered with one not completing the post system interviews. There was some information obtained from students regarding the benefits and positives of the tool, however, due to the small sample size, it cannot be extrapolated that the system could be used on large scale. It did, however, give guidance on future improvements, and an initial exploration of the benefits of an Ipsative system. Following the student interviews staff were also interviewed to evaluate the system. In total five staff members were interviewed from the School of Computing; each staff member was experienced in teaching novice programmers.

### *6.7.3 Future development of the system*

C:IA is a prototype tool and more work is needed to explore the full extent of the benefits of an Ipsative form of learning. A second user trial should occur with more participants to obtain a richer data set. Furthermore, due to the nature of the system, it should to be used in such a way that it does not become a revision tool. As due to the design of the system and the Ipsative theory, the system does not work well as a revision tool. This could be achieved by integrating the system into a module for the tool to be consistently used throughout the year. To explore the usage and benefits of an Ipsative approach, a more parallel usage with time to work on goal development is needed. By using C:IA as a revision tool this development time is removed. As a result, the mistakes and feedback become used for short term mistake identification, rather than repeated long term usage to identify misunderstandings (Struyven et al, 2005; Hughes, 2017, pp37). This provides further evidence to the staff interview responses in Section 4.3.3, which state students tend to take a surface learning approach to assessment and focus their learning on the final examination. However, the pitching of the system as extra work should also be reconsidered.

Furthermore additional development of C:IA should involve a native mobile application. The current compatibility with mobile web browsers was not deemed to be accessible enough. The student interview responses suggest that adopting a native mobile application may encourage a more widespread usage.

When considering the quizzes, a difficulty selector should be implemented. A student should be able to assign a difficulty level and filter out questions that they find too difficult. There should also be a closer link between each content section and the quizzes themselves. More accessibility features need to be considered, such as the ability to increase the text size and clearer identification of features within the system. To ensure that the system can identify a reason as to why the selected solution was incorrect, the options table should have an additional column which stores a reason as to why that choice was incorrect.

### *6.8 Summary*

This chapter outlined the development of a prototype system and reported on an initial set of user tests. C:IA was developed to investigate if an Ipsative form of assessment could help ameliorate the issues faced by students within assessment and feedback practices as identified in Chapter 4 and Chapter 5. These issues included: modularisation of content making skill transference difficult, feedback focusing on the negative aspects as opposed to the positives, feedback not identifying why a mistake was wrong, a lack of practical assessment literacy within students, and finally that feedback arrived too slowly. A small trial of the prototype system was run with four first-year novice programmers. This experiment found that the structured skill and goal tracking features, alongside the available view of past results helped students identify and regulate gaps in their knowledge state. When interviewing students, there were no comments on negative aspects of feedback or feedback delays. Staff were also interviewed for their perspective of C:IA and Ipsative assessment and there were similar positive acknowledgments from staff with the caveat that students must engage with the material beyond a surface level for it to be beneficial. However, due to the small number of participants in this trial, further work is needed to confirm these findings on a larger scale.

# Chapter 7. Discussion

## *7.1 Introduction*

This chapter outlines a reflection on the case studies presented in Chapters four, five and six. The results explored in these chapters are discussed in terms of the research questions and the wider literature on assessment and feedback. First chapter seven explores the good practices identified in the literature on assessment and feedback and how this compares to the practices in Newcastle University. Following this, the methods used to collect data on student perceptions are explored, along with the methods used to analyse student feedback. Finally, this chapter explores the use of Ipsative assessment in a trial with novice programmers and its potential to help ameliorate the issues faced by students.

## *7.2 Good assessment and feedback practice.*

Despite being a crucial part of the higher education student experience, students often report via the NSS that assessment and feedback is the area they are most dissatisfied with. Figure one (Chapter 1, page 2) shows the results of the assessment and feedback questions on the NSS. To understand where this dissatisfaction stems from, three research questions were asked:

> RQ1 What is good practice in assessment and feedback in undergraduate computing science in the UK and how does current practice at Newcastle University compare?
>
> RQ2 What are the current perceived assessment and feedback issues experienced by students?
>
> RQ3 How does Ipsative assessment compare to current practice and can it be used to ameliorate issues identified by students?

To answer the first part of RQ1 and identify good practice in assessment and feedback, a literature review was conducted. The literature review showed that there is no specific or direct answer to this question, due to the nature of assessment and feedback being unique to a subject, module and course. There are however some agreed good practices that can be observed in general. These practices include (i) ensuring that the assessment is valid, (ii) designing assessment that is fair, and (iii) having clear and criteria referenced assessment points (Wakeford, 2003; Hughes, 2011). Most importantly, however, is that all assessment should support student learning. By using Blooms Taxonomy and Kolb's learning theory, deep learning and reflection was identified as important aspects to student learning (Bloom 1956; Kolb, 2014). Deep learning and reflection are important as they allow students to identify their weaknesses and enable them to overcome their issues.

For programming specifically (see Section 2.5), assessment needs to be more than syntax regurgitation. Programming assessments should encourage deep learning and provide an opportunity to practice the core skills of a computer science student i.e. problem solving, applying different programming and design approaches when necessary, code testing and finally error correcting (Rum et al., 2016; Lahtinen et al., 2005). Due to this, programming is often a practice based subject where deep learning often occurs in practical lab sessions. This is the case at Newcastle University where emphasis is placed on practical sessions that can be seen in Appendix C (and fully explained in Section 3.2.4).

Chapter two shows that when considering good practice for feedback, what is important is the ability for the students to reflect on their work. The feedback should allow the students to understand how they are currently performing, where they went wrong and where they did not meet the clearly defined assessment criteria. The feedback should also identify what their next targets would be. This should allow the student and teacher to have an open dialogue about how they can improve and reflect upon the work they have done. Finally, the feedback has to be clear to the student, both in terms of how it relates to the assignment, along with the clarity of language used (Hattie et al., 2006; Ott et al, 2016; Torrance and Pryor, 1998; Nicol 2010; Laurillard 2002).

The second part of RQ1, was concerned with how Newcastle Universities practice compared with that of best practice. As noted, for assessment to be fair and applicable to all, there needs to be multiple types of assessment available so that all students have a fair chance (Brown, 2005). This can be seen in the interview responses given by students, where each student reported a different form of assessment had been used, such as presentations, maths and report writing in addition to code writing (page 64).

Additionally, for summative assessment to be fair and valid it must be clear and criteria referenced (QAA, 2018). This practice can be seen when analysing the programming coursework in Chapter 4 (See Appendix E and F for coursework specifications). While each coursework specification changes the level of detail for the criteria needed, it is clear from the specifications that good practice is being followed. To ensure consistency and fairness with marking and feedback Newcastle University implemented a process called pre and post moderation (pg59), this process ensures that feedback given to students is fair. Nichol (2010) also identified that for feedback to be fair it has to trigger inner reflection within the students. This is achieved through the current feedback structure at Newcastle University as the feedback (analysed in Chapter 5) often tells the students what was incorrect and what a correct solution should look like (This will be discussed further in Section 7.3). This will trigger inner reflection on how the student can get from point A (their incorrect solution) to point B (the correct solution). However, what was often missing was the information of how the student could achieve this (Duncan, 2007, Ferrell 2012) i.e. what the student needed to learn on their own. This however is a purpose of higher education (to facilitate student learning) therefore this can also be seen as guiding students with the option of help if needed. Therefore, it can be said that Newcastle University follows the identified best practice from the literature.

### *7.3 Perceived issues of assessment and feedback.*

To answer RQ2, and identify the current perceived assessment and feedback issues, student perceptions were explored. There were two ways in which student perceptions were explored, the first was student focus groups and interviews and the second was through the Assessment Experience Questionnaire.

There were four main issues identified by students from the interview responses:

- There was a lack of practical assessment and feedback literacy among students. An example is when students identified that the reasons for doing assessments were not always clear (page 53). Whereas staff reported that it is explained why these assessments are completed in both the assessment specification and within lectures. This shows that there is a potential lack of understanding within students about how the assignments connect, which as identified by Ferrell (2012) is a result of the modularisation of current higher education courses. There is also the potential that this is due to a mismatch in communication between the staff and the students.

- The purpose of feedback as identified by students was similar to what was explored within the literature, that feedback was used to help progress the students understanding. In practice however, feedback was often used for mark bartering (page 68), as also identified by Matshedisho (2020) (page 9).

- Written feedback contains too many negative points and does not explore what students did well, along with how they could improve. Feedback should identify where the student went wrong as well as what mistakes were made by the student, identified by Black and William (1998). However, it was also noted that the feedback did not tell the student what to do next or what they did well, which was another important aspect of feedback (Ott et al, 2016).

- Within this study there was also an identified lack of communication between staff and students as identified by Wilson (2006). This was found in two situations, the first was when receiving feedback that students did not understand or that they disagreed with, there was no follow up with staff members. If staff members are not aware that students do not understand the feedback, they cannot provide further clarification, and therefore will assume that students are satisfied with the help that they receive. The second instance is seen when students do not understand the purpose of the assessment and do not clarify with the staff members. In this situation staff will once again assume that students are satisfied with the assessments and continue without further clarification.

These issues were important factors to consider when answering RQ2 and they are clear identifiers as to what was causing student dissatisfaction. These factors were important to consider when developing an Ipsative approach to help ameliorate these issues. However, as there were limited responses to these focus group and interview questions the AEQ was used to collect further data.

While the National Student Survey results show that students are dissatisfied with current assessment and feedback practices, it does not tell us what students are specifically dissatisfied with. While the responses to the interviews and focus groups give some insight into this, they were limited in their responses. To further explore the potential reasons behind this dissatisfaction and to expand upon the responses to the interviews and focus group, the AEQ (Gibbs and Simpson, 2003) was used. The AEQ allows us to explore in more depth some of the issues that students have with assessment and feedback.

As previously explored an issue with the interviews and focus groups was a lack of participation, despite multiple recruitment attempts. The AEQ allowed for a large volume of data collection from students in an anonymous way, which emulated the settings of the NSS. The validity of the questionnaire was assured as it had been peer reviewed and used previously. Using the AEQ in a lecture theatre within a core module resulted in obtaining 535 responses. This data gave a good overview of some of the issues that students had regarding assessment and feedback in higher education.

The first conclusion from this data, is that changing the structure of feedback and improving the quality over time via the use of pre- and post-moderation did not impact student perceptions for this study.

The second conclusion was that the less students could use feedback for mark bartering, the less useful it was perceived to be. This aligned with the reported behaviour of students regarding assessment marks from the literature. Wilson and Yorke explore how students use marks as a dependency and a measure of success. Wilson identified that students often used marks as a beating the mean method. Yorke on the other hand termed students' mark reliance as 'learned dependence'. The student works towards the criteria as opposed to creating a solution to the problem at hand relying on the teacher to give them guidance (Wilson, 2006; Yorke 2003). Additionally, Matshedisho showed that when the marks were removed, students still focused on the potential mark gain, and Ferrell showed that even from a tutor's perspective, marks are often the focus (Matshedisho, 2020; Ferrell, 2012). The mark driven nature of students can be seen in multiple parts of this study. In Section 4.3 student responses to interviews showed that a potential reason for their mark driven nature was due to previous experiences of assessments. The reported experience of assessment was primarily exam driven, where feedback was minimal, and marks were the primary focus. This was perhaps enforced by staff members focusing on the mark aspect of feedback, as shown in Section 4.3.3 where Teacher E identified that feedback was used to help students understand where they lost marks. Furthermore, the mean responses to statement 22 (2016/2017:2.91, 2017/2018:3.51  2018/2019:2.90) of the AEQ show that students did not agree that feedback helped them to understand the marks they received on the assignment, therefore showing dissatisfaction with this aspect of their feedback.

The third conclusion from the AEQ was that students felt that receiving more feedback would not be beneficial to them. This was explored through the ANOVA tests (page 96) and the correlation tests (page 98) on statement 17. This aligns with Struyven and colleagues (2005) and Price colleagues (2010) who identified that when students do not find assessment and feedback to be in a suitable form, or to be clear, they will disengage. This disengagement can be seen in students identifying that more feedback would not be beneficial.

It was important to understand if these issues were present within feedback, therefore a historical analysis was completed on feedback given to students.

Using the feedback criteria developed in Chapter two, historical data was analysed to investigate if the issues identified by students were present within the reviewed feedback. Chapter Five shows a full exploration of this data.

The findings show that the issues identified by students in Chapter four were present within the feedback. The main identification from the analysis of historical data was that while feedback (in general) often identified where the student was incorrect and what a correct solution looked like; it did not explain why it was incorrect (FC1). The feedback also did not explain what the student needed to learn to produce a correct output (FC5). This focus that feedback had on the summative aspects of the work and not the formative aspects, show evidence to Ferrell's (2012) identification that formative feedback frameworks are rare and lacking in detail. Further evidence to this is that while pre- and post-moderation improved consistency within the feedback, the lack of guidance on how to improve was still an issue.

There were, however, limitations with this collection method, the first being the subjectivity. Identifying if some of the feedback criteria were met was based on the researcher's judgement, such as whether feedback was judgemental. This subjectivity is understandable due to the nature of feedback, it is in essence a judgement on the marker's behalf (page 30, Torrance and Pryor, 1998). To mitigate this and ensure that there was consistency, inter-rater reliability was completed. This was a statistical test to measure the agreement between two 'raters' on a data set (see page 48 and page 122 for full details). The result of this test was .446 which was a moderate agreement, this means that there is still a portion of judgement within the analysis.

Another issue faced in the analysis of the historical data was the differences in the display for the feedback. Depending on the method of access used to view the data, the ASCII codes introduced were more frequent. Therefore, depending on which method was used the clarity of the feedback could vary. For consistency the researcher chose to use the version with which access was granted as it was unclear what was and what was not present within the other versions. It is suspected that a different view could produce vastly different results if the formatting issues were not present.

### *7.4 Ipsative Assessment.*

RQ3 investigates if Ipsative assessment could ameliorate the issues faced by students. Before investigating the usage of Ipsative assessment the theory of it needed to be explored along with a comparison to traditional and summative formative assessment. Embracing the theory of growth-mindsets, Ipsative assessment compares the student's past performance with their current performance. It reviews the students' skills and abilities and gives personalised instruction on how to improve. The focus is less on external standards and class-wide comparison and more focused on the individual steps a student must make (Hughes, 2011; Hughes, 2014; Hughes, 2017a).

When comparing Ipsative assessment to current good practices, there are many similarities. Ipsative assessment and feedback place a large emphasis on self-regulated learning and feeding forward to future assessments while also reflecting on past work (pg39). When implementing best assessment and feedback practices, it becomes clear that Ipsative assessment is a central part of good practice in facilitating and supporting learning.

An issue arises however, when aiming to give summative marks to an Ipsative form of assessment. Giving a quantifiable grade to a student's learning gain is difficult as not every student has the same distance to travel. Therefore, fairness in marking becomes challenging to uphold. Additionally, quantifying learning gain is difficult within higher education as keeping these processes in line with university regulations and keeping assessment fair is challenging. Section 2.6 (page 37) identified that portfolios are a way in which Ipsative assessment and feedback can be implemented in a summative structure. Portfolios allow students to track improvements that the they have made through their learning journey (Hughes, 2017, pg37). However, the limitation is that the quantifiable grade provided to the student is often based on the final submission, not the growth they have made. Therefore, a viable way for Ipsative assessment to be implemented is by focusing on Ipsative feedback (Hughes, 2017b, pp30). The benefits of Ipsative feedback can be seen in the user trial of C:IA in Chapter 6 (page 165), where student 4 identified that the Goals page allowed them to keep track of their learning, structure what they needed to do and see their progress. This gives some evidence to the benefits of the use of Ipsative feedback in addition to the traditional summative structure of assessment and feedback.

As outlined in chapter six, Computing: Ipsative Assessment (C:IA) was developed to help explore the potential benefits of Ipsative assessment. The full development (page 150) and interview responses (page 162) for the system can be seen in Chapter six. The Goal setting, results page and personalised aspects of Ipsative assessment were well received. While self-regulation has been identified as an important skill for students to have, this is the main area where the system struggled to engage students. Students reported that use of the system was minimal until the end of the semester when it was used for exam revision.

A finding from this research which did not align with the literature was the necessity to keep materials in a computer-based system similar to that of the material of the lectures (Moscinska et al, 2018). Within this study, students using C:IA identified that the content was too similar to that of their lecture notes, and therefore was not useful to them. This suggests that while keeping content similar is important, having some variation to keep students engaged is also important. The differences in content within C:IA were focused within the Quiz section and the Goals section, the disengagement from the students shows that for this study similarity to the content was not beneficial.

Another issue identified was that when the questions became too difficult, the students disengaged and lost confidence (Page 164). This shows further evidence to Robins theory that some students are 'stoppers' (Robins et al, 2003). In Section 6.6.1 (page 164) in the post systems interviews, student 3 identified that they felt they could not attempt the quizzes because they were beyond their difficulty level. However, student 2 showed evidence of being a 'mover' as when faced with a difficult challenge they experimented with the code in an IDE of their choice.

The staff interview responses have similar themes to that of the students, the benefits to using the system (and Ipsative assessment as a whole) would result in an increase in student independence, as well as increased use of deep learning practices (HEFCE, 2016; Hughes, 2017). Additionally the practice area supplied by C:IA is a crucial component for computer science students due to the practical nature of the subject (Ala-Mutka, 2005; Rum et al 2016; Bradley 2020).

There were concerns identified regarding engagement with the material where staff expressed a concern that C:IA (specifically the quizzes) would only be useful if there was engagement from the students (pg 172). This was shown in the responses of students where there was a disengagement due to the difficulty of the quizzes and the additional work that the system gave students. While students identified that the goals feature was the most beneficial, staff identified that there needed to be further developments with this feature. It would be beneficial for the students to have guidance on how to create actionable realistic goals (pg172). This additional development of the goals feature will improve this section for further student use.

The full impacts of having an Ipsative form of assessment cannot be concluded from this work. This work does, however, suggest that the concepts that make up Ipsative assessment and feedback are beneficial to students.

### *7.5 Ipsative as a solution*

RQ3 investigates whether Ipsative assessment can ameliorate the issues identified throughout this research. Ipsative assessment is a powerful technique which could help students reach their full potential. The issues identified by students such as: modularization making links between modules difficult, feedback being too negative and students using a surface learning approach, correlate into a lack of personalisation as well a lack of practical assessment literacy. The case studies completed within this research suggest that Ipsative assessment could help to ameliorate these issues. Ipsative assessment can help develop practical assessment literacy within students by developing self-reflection and self-regulation skills. Furthermore, the skills developed when using an Ipsative approach, such as the ability to identify unhelpful and unbeneficial behaviour may enable students to develop working skills which could be applied in other modules (Hughes, 2017, Chapter 2, pp 38). However, Ipsative assessment and feedback provides challenges to a classroom environment, as a fully realised Ipsative system could be impractical and time-consuming. As discussed in Section 2.6 (page 40) Ipsative assessment is challenging to use within a summative system as fairness and validity of quantifying learning gain is difficult. Therefore, Ipsative within summative assessment in the current higher education structure is challenging and needs further exploration (Hughes, 2017; Zhou and Zang, 2017). However, as a formative technique Ipsative assessment is a useful tool to encourage students to become self-reflective and to help them develop a deep learning approach. Additionally, when interviewed regarding Ipsative assessment, staff expressed concerns over the additional workload Ipsative assessment would cause specifically with the larger class sizes (also explored on page 169; Hughes 2017; UniversitiesUK, 2017). This gives further evidence to the suggestion that if Ipsative Assessment is to be implemented in a classroom, Computer-based assessment needs to be used to assist with this.

### *7.6 Summary*

Chapter seven discusses the findings from the literature review and case studies explored to answer the research questions defined in Chapter one. This chapter explores the three case studies and their findings completed within this research. These studies found that while Summative Ipsative assessment is implausible and difficult to implement fully, Ipsative practices and Ipsative feedback can be used to support student learning.

# Chapter 8. Conclusion and Future Work

## *8.1 Introduction*

This research aimed to investigate if Ipsative assessment could help ameliorate the issues faced by students regarding assessment and feedback within Higher Education. Chapter one defined three research questions

RQ1 What is goof practice in assessment and feedback in undergraduate computer science in the UK and how does current practice at Newcastle University compare?

RQ2 What are the current perceived assessment and feedback issues experienced by students?

RQ3 How does Ipsative assessment compare to current practice and can it be used to ameliorate issues identified by students?

To answer these questions, in chapter two this thesis explored the literature surrounding assessment and feedback within Higher Education, specifically with a focus on computer science. Chapter three explores the methods used to collect data for the three case studies within this research. In Chapter four, the exploration of the perceptions of students were completed via the use of student and staff interviews, as well as the use of the Assessment Experience Questionnaire (Gibbs and Simpson, 2003). The second case study explored in Chapter five was the analysis of historical feedback given to students using the feedback criteria developed within Chapter two. Finally, in Chapter six, the tool Computing: Ipsative Assessment was developed and trialled with novice programmers in the module CSC1021.

## *8.2 Research contributions*

This study expands the understanding of Ipsative assessment and feedback with relation to computer science. It has demonstrated that Ipsative assessment is a valid way to improve student progress, by placing focus not on external criteria, but on the small individual steps that students can take. It supports personal growth and helps to develop a deep learning mindset within students. Additionally, a set of feedback criteria was developed in Chapter two which provides a guide on how to write good feedback at a general level. While each subject may have specific forms of feedback that they have to use, the feedback criteria provides a good guideline on how to create feedback which benefits students for all subjects. This has the additional benefit of collating the identified good practice across literature into one place.

Two data sets were collected to give a further understanding of student perceptions of assessment and feedback. This can be seen in Chapter four where perceptions were collected in two forms, the first was interviews and focus groups and the second was through use of the AEQ survey. This data collection adds to the body of knowledge by introducing an additional data set within a computer science context to the pre-existing ones. This could be expanded upon by looking at computer science students within other higher education institutes. As this was completed with first year students it also gives a perspective on what needs to be changed, or what needs to be communicated differently through the remainder of their studies.

Following this, a second study was completed using the developed feedback criteria to investigate if the perceived issues were present. This case study demonstrated that the perceived issues regarding assessment and feedback from the student perspective are present in feedback given to students. The feedback itself focused on the negative aspects of the work and provided little direction as to why something was incorrect. From this, it could be extrapolated that as the NSS has a national dissatisfaction in assessment and feedback, student perceptions are valid, and work needs to be completed to develop upon these practices. The data set would also be beneficial to do a comparison of feedback given to students in different disciplines with the same set of feedback criteria to identify any differences and how each discipline could learn from each other.

Finally, this work developed and tested a web-based platform to implement Ipsative practices and trialled the system with students. This trailed a practical form of Ipsative assessment in a setting with students, giving a further example of Ipsative Assessment in practice. The goal-focused steps allowed students to practice identifying the skills and topics they needed to develop to obtain a mastery over a subject. However, it was also noticed by the students that the steps that went beyond the scope of the module were considered to be too difficult.

### *8.3 Limitations of the work*

There are several limitations to this study. The first is the exploration of student perceptions with data from one university. This makes the data collected difficult to extrapolate to other universities, due to each university having different prerequisites and different teaching styles.

Furthermore, the participants of the Ipsative assessment trial was a small number meaning that from these initial responses, it is difficult to extrapolate these results. In addition, due to time constraints, the system was only trialled once and therefore, has limited results.

Finally, the statistical modules used within chapter four give an understanding towards the perceptions of assessment and feedback from students. However, the richest data comes from the limited interviews and focus group. The statistical tests lack a personalised element to truly explore the impact on students and is therefore limited in the information it can give.

### *8.4 Conclusion*

This work has identified that Ipsative assessment and feedback as a concept, can develop skills to help ameliorate the issues identified by students and the literature. However, there are limitations and barriers that prevent a fully Ipsative system within higher education. The primary concern for Ipsative assessment is aligning the practice with current summative and grade driven work. Giving quantifiable evidence to a student's learning gain is difficult to achieve, especially when considering the varying level of abilities. The tasks for each individual student to complete can drastically range for an equal amount of learning gain depending upon their initial starting point.

Additionally, Ipsative assessment and feedback is a time and energy-intensive practice. The assessor is required to not only mark the current assessment, but to additionally consult previous feedback and identify how the student has improved. This can be mitigated through the use of technology by supplying the information on a central hub, however, this requires assessors to be consistent in maintaining such a system.

### *8.5 Future work*

There are two primary directions in which the development of this research should continue.  The first being further development of the Computing: Ipsative Assessment tool, accompanied by a trial on a larger scale for a longer time (perhaps embedded in a module).  In future trials further involvement and guidance from a staff member to support the students is required to ensure consistent use of the system. The system itself should also be explored by staff members and assessors to gain a different perspective on the usefulness of such a system.

The statistical tests of chapter four and five give an understanding of the changes in student perceptions and the changes within the given feedback. An ethnographic study to understand the perceptions and trials of students should be completed. This study could be completed through an audit of an Undergraduate module to investigate how assessment and feedback are communicated to the students. An ethnographic study will allow the researcher to see the impact of the of an Ipsative approach first hand and will enable the gathering of a rich data set.

While not in the initial remit of this work, it is relevant to discuss COIVD19 and the impacts that this could have on not only education as a whole but on assessment and feedback itself. With a large portion of classes and education moving online, Ipsative assessment and feedback may become a more viable option. Having large classes and competition with peers is currently not possible. Therefore, with a stronger stimulus to drive learning and for a student to be able to identify improvements and good quality work, Ipsative assessment may be that guide. Through the use of online learning, Ipsative assessment and feedback will give students the ability to identify not only how they have improved but what the next steps within their learning are.

## References

Ala-Mutka, K. M. (2005) 'A survey of automated assessment approaches for programming assignments', Computer Science Education, 15(2), pp. 83–102. doi: 10.1080/08993400500150747.

Alderman L, Wiliams AW, (2005) Ipsative assessment within an integrated assessment structure – a case study in the Graduate Certificate in the Practice of Tertiary Teaching. The University of Newcastle, Newcastle, New South Wales. [Working paper].

Allen B (2020) Available at: https://www.researchgate.net/profile/Becky_Allen3 [Accessed: 18th December 2020].

Apache, (2012) Apache Cordova [Online] Available From: https://cordova.apache.org Last [Accessed: 31st May 2018]

Apple. (2015). macOS Big Sur. [Online] Available From: https://www.apple.com/macos/big-sur/ [Accessed: 10th December 2020].

Arthurs NA, Stenhaug BS, Karayev KS (2019), Grades are not normal: improve exam score models using the logit-normal distribution, EDM Available at https://stanford.edu/~cpiech/bio/papers/gradesAreNotNormal.pdf [Accessed: 7th April 2020].

Bailey, J. L. and Stefaniak, G. (2001) 'Industry perceptions of the knowledge, skills, and abilities needed by computer programmers', Proceedings of the ACM SIGCPR Conference, pp. 93–99. doi: 10.1145/371209.371221.

Belo YB, Moro SM, Martins AM, Ramos PR, Costa JMC, Emerado, (2019), Constructed Response or Multiple Choice Questions for Assessing Declarative programming knowledge? That is the Question!, Journal of Information technology education: Innovations in practice. Volume 18 pp153-170

Black PB, (1998), Formative assessment: raising Standards inside the classroom, School Science Review, 80:291, 39-46

Black PB, Wiliam DW, (1998), Assessment and Classroom Learning, Assessment in Education: Principles, Policy & Practice, 5:1, 7-74, DOI10.10810/0969595980050102

Bloom BS, Engelhart ME, Furst, EJF, Hill, WH, Krathwohl KD, (1956), Taxonomy of educational objectives, David McKay Company, London.

Bloxham, SB, Boyd, PB, (2007), Developing Effective Assessment in Higher Education a practical guide, England: Open University Press.

Bootstrap, (2011) [Online] Available from: https://getbootstrap.com/ [Accessed: 10th December 2020]

Boud DB, (2009), Assessment 2020, Seven propositions for assessment reform in higher education, Australian Learning and Teaching Council. Available from: https://www.uts.edu.au/sites/default/files/Assessment-2020_propositions_final.pdf [Accessed: 18th December 2020]

Boud DB, Molloy EM, (2013), What is the problem with feedback? Feedback in *Higher and Professional Education*, London: Routledge, 1-10

Bradley SB, (2020) Creative Assessment in Programming Diversity and Divergence, *Computing Education Practice*

Brown SB, Knight PK, (2005) *Assessing Learners in Higher Education* Digital Printing Abingdon: ReoutledgeFalmer 13. 1-4

Brown GB, (2001) Assessment Series Assessment guide for lectures. Ltsn generic centre. Learning and Teaching support network.

Brown SB, (2005), Assessment for Learning, *Learning and Teaching in Higher Education*, Issue 1, 81-89

Burgess GAB, (2005) Introduction to Programming: Blooming in America *Journal of Computing Sciences in Colleges* 21(1) p19-28

Callender CC, Ramsden PR, Griggs JG, (2014) Review of the national student survey, Higher education funding council for England.

Calvin Douglas Smith , Kate Worsfold , Lynda Davies , Ron Fisher & Ruth McPhail (2013) Assessment literacy and student learning: the case for explicitly developing students ‘assessment literacy’, Assessment & Evaluation in Higher Education, 38:1, 44-60, DOI: 10.1080/02602938.2011.598636

CHI 2020. (2018). CHI 2020. [Online] Available From: https://chi2020.acm.org/. [Accessed: 18th December 2020]

Chow AC, Croxton RAC, (2017) Designing a Responsive e-Learning Infrastructure: Systemic Change in Higher Education, *American Journal of Distance Education*, 31:1 pp20-42 DOI: 10.1080/08923647.2017.1262733

Codecademy (2019) Codecademy [Online] Available from: https://www.codecademy.com/ [Accessed 04/03/2019]

Coffey, H. (2008) Bloom's Taxonomy,. [Online] Available From: https://www.academia.edu/12127523/Blooms_Taxonomy. Accessed: [18th December 2020]

Cohen LC, Manion LM, Morrison KM, (eds.), (2018), *Research Methods in Education*, eighth edition, London and New York: Routledge.

Cukieman DC, Thompson DMT, Guoly SG, Salimi FS, Karpilovsky MG, (2014). Addressing challenges students face in first year university Computer Science and Engineering Science courses: Overview of a needs assessment workshop.  Volume 4 pp1-5 WCCE'14

Daly CD, Waldron JW, (2004), Assessing the assessment of programming ability, 36(1) pp210-213 SIGCSE'04

Deneen CD, Brown GTBD, Carless, (2017), Students' conceptions of Eportfolio as assessment and technology. *Innovation in Education and Teaching International*. 55(4) pp487-496.

Desmond, B. and Jowitt, A. (2012) 'Stepping into the unknown: Dialogical experiential learning', Journal of Management Development, 31(3), pp. 221–230. doi: 10.1108/02621711211208853.

DN, Macfarlane-Dick, (2006), Formative assessment and self-regulated learning: a model and seven principles of good feedback practise, Studies in Higher Education, 31:2, 199-218

Duncan, N. (2007) '"Feed-Forward": Improving students' use of tutors' comments', Assessment and Evaluation in Higher Education, 32(3), pp. 271–283. doi: 10.1080/02602930600896498.

Dweck, C, Leggett (1998), Asocial-Cognitive Approach to Motivation and Personality, Psychological Review, Vol. 95, No.2, pp 256-273

Eugenia M.Q. Ng (2018) Integrating self-regulation principles with flipped classroom pedagogy for first year university students. *Computers and Education.* 126 pp 65-74

Evans CE, (2013), Making sense of assessment feedback in higher education. Review of Education research vol 83(1). pp. 70-120.

Eclipse Foundation, (2001) Eclipse [Online] Available From https://www.eclipse.org/ Last Accessed [11th December 2020]

Ferrell GF, (2012), A view of the Assessment and Feedback Landscape: baseline analysis of policy and practice from the JISC Assessment & Feedback Programme. JISC advance. Available online: https://www.jisc.ac.uk/rd/projects/assessment-and-feedback Last Accessed: 30/06/2021

Francis, R.A. (2008). An investigation into the receptivity of undergraduate students to assessment empowerment. *Assessment & Evaluation in Higher Education* 33(5) :547–57.

Garner, RG, (1990), When Children and Adults do not use learning strategies: Towards a Theory of Settings, Review of Education Research¸ 60(4), pp. 517-529.

Gibbs GG, Simpson CS, (2003), Measuring the response of students to assessment: The Assessment Experience Questionnaire, 11th Improving Student Learning Symposium. Open University .

Gibbs, GG and Simpson, CS (2005) Conditions Under Which Assessment Supports Students' Learning. Learning and Teaching in Higher Education (1). Pp3-31. ISSN 1742-240X

Gill Turner & Graham Gibbs (2010) Are assessment environments

gendered? An analysis of the learning responses of male and female students to different

assessment environments, Assessment & Evaluation in Higher Education, 35:6, 687-698, DOI:

10.1080/02602930902977723

Hattie JH, Timperly HT, (2007) The Power of Feedback, *Review of Educational Research*, 77(1), pp 81-112

Heels LH, (2020) CSC1034 MOF, [Online] Available at https://www.ncl.ac.uk/undergraduate/modules/csc1023/ [Accessed: 18th December 2020]

Higher education academy, (2012), 10 tools for exploring and evaluating your own feedback practice [Online] Available From: https://s3.eu-west-2.amazonaws.com/assets.creode.advancehe-document-manager/documents/hea/private/resources/10_tools_for_exploring_feedback2_1568036614.pdf Last Accessed. [18th December 2020].

Higher Education Academy, (2015), Transforming Teaching Inspiring Learning, Assessment Audit Tool, [Online] Available from: https://www.heacademy.ac.uk/knowledge-hub/assessment-audit-tool-taught-modules [ Accessed: 12th June 2017]

Higher Education Funding Council for England HEFCE (2016), Review of information about learning and teaching, and the student experience [Online] Available from https://dera.ioe.ac.uk/24436/1/HEFCE2015_24.pdf [Accessed 18th December 2020]

Higher Education Funding Council for England HEFCE, (2017), National Student Survey, http://www.hefce.ac.uk/lt/nss/results/[ Accessed 18th December 2020]

Hughes GH (2011) Towards a personal best: a case for introducing Ipsative Assessment in higher education, Studies in Higher Education, (36:3) 353-367, DOI: 10.1080/03075079.2010.486859

Hughes, GH, Wood, EW, Kitagawa, KK, (2014), Use of self-referential (Ipsative) feedback to motivate and guide distance learners, Open Learning: The Journal of Open, Distance and e-learning, 24(1), pp 31-44.

Hughes GH, Smith HS, Creese BC, (2015), Not seeing the wood for the trees: developing a feedback analysis tool to explore feed-forward in modularised programmes, Assessment & Evaluation in Higher Education, 40(8), 1079-1094

Hughes GH, (2017) 'Introducing Ipsative Assessment and Personal Learning gain: Voices from Practitioners and the Themes of the Collection' in Hughes G. (ed.) Ipsative Assessment and Personal Learning gain pp. 1-24

Hughes GH, (2017b) 'Exploring the relationship Between Ipsative Assessment and Institutional Learning Gain' in Hughes G. (ed.) Ipsative Assessment and Personal Learning gain pp. 25-42

Iannone IP, Simpson SA, (2015), Students' views of oral performance assessment in mathematics: straddling the 'assessment of' and 'assessment for' learning divide, Assessment and Evaluation in higher education, 40(7), 971-987, DOI:10.1080/02602938.2014.961124

Ibabe I, Jauregizar J, (2009), Online self-assessment with feedback and metacognitive knowledge, Springer, Higher Education, 59 pp243-357

Ihantola PI, Ahoniemi Ta, Karavirta KV, Seppälä OS (2010) 'Review of recent systems for automatic assessment of programming assignments', Proceedings of the 10th Koli Calling

International Conference on Computing Education Research, Koli Calling’10, pp. 86–93. doi: 10.1145/1930464.1930480.

Jones, R. and Thomas, L. (2005) ‘The 2003 UK government higher education white paper: A critical assessment of its implications for the access and widening participation agenda’, Journal of Education Policy, 20(5), pp. 615–630. DOI: 10.1080/02680930500222477.

Jones L, Allen B, Dunn P, Brooker L(2017) Demystifying the rubric: a five-step pedagogy to improve student understanding and utilisation of marking criteria, Higher Education Research & Development, 36(1), 129-142, DOI: 10.1080/07294360.2016.1177000

Kandiko CK, Mawerm (2013). Student expectations and perceptions of Higher Education. London: Kings Learning Institute Uk: London.

Khairuddin NK, Hashim KM, (2008), Application of Bloom’s Taxonomy in Software engineering Assessments, WSEAS ACS’ 08 pp 66-69

Khan Academy (2020) https://www.khanacademy.org/about Accessed [10th December 2020]

Matshedisho KR (2020) Straddling rows and columns: Students’ (mis)conceptions of an assessment rubric, Assessment & evaluation in higher education, 45(2), 169-179, DOI:10.1080/02602939.2019.1616671

Kolb, DK, (2014), Experiential Learning, experience as the source of learning and development, 2nd edition, New Jersey

Kuiper RAK, Pesut DJP, (2004), Promoting cognitive and metacognitive reflective reasoning skills in nursing practice: self regulated learning theory, Issues and innovation in nursing education 45(4) pp 381-391

Laerd.com. (2018). Cohen’s kappa in SPSS Statistics - Procedure, output and interpretation of the output using a relevant example | Laerd Statistics. [Online] Available From: https://statistics.laerd.com/spss-tutorials/cohens-kappa-in-spss-statistics.php [Accessed 21 August. 2019].

Lahtinen EL, Mutka KAM, Järvinen HMJ, (2005), A study of the difficulties of novice programmers, ITiCSE’05 pp14-18

Lajis AJ, Baharundin AB, Kadir AK, Ralim NMR, Nasir HMN, Aziz NAA, (2018), A review of Techniques in automatic programming assessment for practical skill test. 10(2-5) pp109-113

Laurillard, D. (2002). Rethinking university teaching: A framework for the effective use of learning technologies. 2nd ed. London: RoutledgeFalmer

Leibold, N. and Schwarz, L. M. (2015) 'The art of giving online feedback', *Journal of Effective Teaching,* 15(1), pp. 34–46. [Online] Available From: http://ezproxy.ace.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=eric&AN=EJ1060438&site=eds-live&scope=site.

Linda B Nilson (2013). Creating self-regulated learners strategies to strengthen students' self-awareness and learning skills. Sterling Stylus Publ. Cop.

Linux.org. (2017). Linux.org. [Online] Available From: https://www.linux.org/.

Lister Rl, Adams EA, Fitzgerald SF, Fone WF, Hamer JH, Lindholm ML, McCartney RM, Moström JEM, Sanders KS, Seppälä, Simon BS, Thomas, (2004) A Multi-National Study of Reading and Tracing Skills in Novice Programmers. ITiCSE 2001. 36(4) pp119-150

Lister RL, Leaney JL, (2003), Introductory Programming, Criterion-referencing, and Bloom. SIGCSE'03, DOI: https://doi.org/10.1145/611892.611954 35(1):143-147

Margulieux LM, Catrambone RC, (2017) Using Learners' Self-Explanations of Sub goals to Guide Initial Problem Solving in App Inventor, *ICER17* pp 21-29

Marin VM, Pereira TP, Sridharan SS Rivero CRR, (2017), Automated Personalised feedback in introductory Java programming MOOCS, IEEE 33rd International Conference on Data Engineering, pp1259-1270

Matshedisho, KRM (2020) Straddling rows and columns: Students' (mis)conceptions of an assessment rubric, Assessment & Evaluation in Higher Education, 45:2, 169-179, DOI: 10.1080/02602938.2019.1616671

Moscinska KM, Rutkowski (2018), Effective computer-assisted assessment: challenges and solutions IEEE Global Engineering Education Conference, pp. 969-978, doi: 10.1109/EDUCON.2018.8363335

Moss SAM, White BW, Lee JL (2017), A systematic review into the psychological causes and correlates of plagiarism. *Ethics and Behaviour.* 28(2) pp. 261-283

Nelson GN, Xie BX, Ko AK, (2017) Comprehension First: Evaluating a Novel Pedagogy and Tutoring system for program tracing in CS1, ICER17 August 18-20 pp2-11

Newcastle University (2019a), Policy and Guidance on Moderation and Scaling Purpose and Objectives. [Online] Available From: https://www.ncl.ac.uk/ltds/assets/documents/qsh-assmt-modscal-pol.pdf Accessed [7 April 2020].

Newcastle University, (2019b) Computing Entry Requirements URL: https://ncl.ac.uk/undergraduate/subjects/computing/ Last Accessed [3rd August 2020]

Newcastle University, (2019c), Policy and Procedures for External Examiners of Taught Programmes. (n.d.). [Online] Available From: https://www.ncl.ac.uk/ltds/assets/documents/qsh-extexam-policy.pdf Accessed [7 April 2020].

Newcastle University, (2019d), Policy on Assessment and Feedback [Online] Available From: https://www.ncl.ac.uk/ltds/assets/documents/qsh-assmt-assessedwork-policy.pdf

Newcastle University (2019e), CSC1021 [Online] Available From: https://www.ncl.ac.uk/undergraduate/modules/csc1021/ Accessed [7 April 2020].

Newcastle University (2019f), Policy on Assessment and Feedback. [Online] Available From: https://www.ncl.ac.uk/ltds/assets/documents/qsh-assmt-assessedwork-policy.pdf Accessed [7 April 2020].

Newcastle University, (2020), NESS, [Online] Available From: ness.ncl.ac.uk Accessed [11th December 2020]

Nesbitt D, (2020) CSC1021 exam papers. MOF Available at https://www.ncl.ac.uk/undergraduate/modules/csc1021/ Accessed [11th December 2020]

Newcastle University (2015) Numbas. [Online] Available From: https://www.numbas.org.uk/ Accessed [11th December 2020]

Nicol DN, (2007) E-assessment by design: using multiple-choice tests to good effect, journal of further and higher education, 31(1), 53-64, DOI: 10.1080/03098770601167922

Nicol DN, (2010), From monologue to dialogue: improving written feedback processes in mass higher education. *Assessment & Evaluation in Higher Education* 35(5) pp 501-517.

Nicol DN, Macfarlane-Dick, (2006), Formative assessment and self-regulated learning: a model and seven principles of good feedback practise, Studies in Higher Education, 31(2), 199-218

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI ’94. Pp152-158

Nilson LB, (2013) *Creating Self-Regulate Learners: Strategies to Strengthen Students’ Self-Awareness and Learning Skills*. First Edition: Virginia: Stylus Publishing

Norman GN, (2010) Likert scales, levels of measurement and the “laws” of statistics. Advances in Health Sciences Education 15(5):625-32 DOI:10.1007/s10459-010-9222-y

Nusconnect.org.uk. (2019). Assessment and Feedback Benchmarking Tool @ NUS Connect. [Online] Available From: https://www.nusconnect.org.uk/resources/assessment-and-feedback-benchmarking-tool. Accessed [7 April 2020].

Office for Students (2019) *NSS 2019 Core questionnaire*. [Online] Available From: https://www.officeforstudents.org.uk/media/ceed008b-1f02-48ad-b566-d9ca7080376f/nss-2019-questionnaire.pdf Accessed [27th November 2019].

Office for Students (2020). Updating the NSS - Office for Students. [Online] Available From: https://www.officeforstudents.org.uk/advice-and-guidance/student-information-and-data/national-student-survey-nss/updating-the-nss/ [Accessed 7 Apr. 2020].

Ott CO, Robins AR, Shepard KS, (2016), Translating Principles of Effective Feedback for Students into the CS1 Context, ACM Transactions on Computing Education, 16(1), 1-27

Pallent JP, (2010), SPSS Survival Manual, Berkshire: Open University Press. 4th edition.

Perfect CP, (2015) A demonstration of Numbas, an assessment system for mathematical disciplines. Available Online http://www.numbas.org.uk/wp-content/uploads/2015/07/caa2015_numbas.pdf Accessed [11th December 2020]

Petrovic JP, Pale PP, Jeren BJ, (2017), Online formative assessments in a digital signal processing course, edu in technology 19(1), pp.209–225

Pintrich PP, Wolters CW, Baxter GB, (2000) Assessing Metacognition and Self regulated learning, *Issues in the measurement of metacognition, 3.*

Price MP, Handley KH, Millar JM, O’Donovan BO, (2010), Feedback: all that effort, but what is the effect?, Assessment & Evaluation in Higher Education, 35(3) pp 277-289.

QAA (2012) Chapter B5: Student Engagement. *Uk Quality Code for Higher Education* [Online] Available From: *https://www.qaa.ac.uk/docs/qaa/quality-code/chapter-b5_-student-engagement.pdf?sfvrsn=cd01f781_8* Accessed [10th December 2020]

Qualifications and Curriculum Authority, (2004), The keys skills qualifications standards and guidance, [Online] Available From: https://dera.ioe.ac.uk/4981/1/8929_qca04_1294.pdf Accessed [7 April 2020].

Richardson, JTER, (2011) Eta squared and partial eta squared as measures of effect size in educational research. *Educational Research Review* 6(2) pp 135-147

Robins AR, Rountree JR, Rountree NR, (2003), Learning and Teaching Programming: A review and discussion. *Computer Science Education* 13(2) pp.137-172

Rum SNMR, Ismail MAI, (2016), Metacognitive support accelerates computer assisted learning for Novice Programmers, *Educational Technology & Society*, 20(3), 170-181

Sanfoundry. (2011). 1000 Java MCQs for Freshers & Experienced | Sanfoundry [Online] Available From: https://www.sanfoundry.com/java-questions-answers-freshers-experienced/ Accessed [7 April 2020].

Sarkar, NI, (2006), Teaching computer networking fundamentals using practical laboratory exercises. *IEEE Transactions on Education*. 49(2) pp285-291

Schanzenbach DWS (2012) Limitations of experiments in education research, *Association for education Finance and Policy* [Online] Available From https://www.sesp.northwestern.edu/docs/publications/2136567975551eeb5641769.pdf Accessed[18th December 2020].

Sclater, NS, Peasgood, AP, Mullan, JM, (2016), Learning Analytics in Higher Education. A review of UK and international practice, full report. JISC [Online] Available from: https://analytics.jiscinvolve.org/wp/2015/06/15/jiscs-learning-analytics-architecture-whos-involved-what-are-the-products-and-when-will-it-be-available/ [Accessed 2nd December 2020].

Scott ST, (2003) Bloom's taxonomy applied to testing in computer science classes, *Australasian Computing Education Conference (*ACE2008), 19(1) pp.267-274

Shadbolt, (2016), Shadbolt Review of Computer Sciences Degree Accreditation and Graduate Employability [Online] Available From:

https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/518575/ind-16-5-shadbolt-review-computer-science-graduate-employability.pdf. Accessed [5th November 2020]

Sridharan BSm Muttakin MD, Mihret MD, (2018) Students' perceptions of peer assessment effectiveness: an explorative study, *Accounting Education*, 27(3) pp. 259-285

Starr, C. W., Manaris, B. and RoxStalvey, A. H. (2008) 'Bloom's taxonomy revisited: Specifying assessable learning objectives in computer science', SIGCSE'08 - Proceedings of the 39th ACM Technical Symposium on Computer Science Education, 40(1) pp. 261–265. doi: 10.1145/1352135.1352227.

Stegeman, M., Barendsen, E. and Smetsers, S. (2016). Designing a rubric for feedback on code quality in programming courses. Proceedings of the 16th Koli Calling International Conference on Computing Education Research - Koli Calling '16. Pp160-164

Stephanou, Gkavras GG, Doulkeridou MD, (2003), The role of Teachers' Self – and Collective Efficacy believes on their job satisfaction and experienced emotions in school, Psychology Scientific research 4(4) pp.268-278

Struyven, K., Dochy, F. and Janssens, S. (2005). Students' perceptions about evaluation and assessment in higher education: a review. Assessment & Evaluation in Higher Education, 30(4), pp.325–341.

Stiggins RS (1991) Assessment Literacy, The Phi Delta Kappen, Vol. 72 pp 534-539 [Online] https://www.jstor.org/stable/20404455?seq=1#metadata_info_tab_contents Last Accessed 12/04/2021

Tai JT, Adachi AC, (2019), The transformative role of self- and peer-assessment in developing critical thinkers. In Bryan CB, Clegg KC, (eds.) Innovative assessment in Higher Education, A handbook for Academic Practitioners. London and New York: Routledge

Taras MT, (2008), Summative and Formative Assessment. Perceptions and Realities. *Active learning in higher education*. 9(2) pp 172-192

The Royal Society (2017) After the reboot: Computing education in Uk Schools. [Online] Available From: https://royalsociety.org/~/media/policy/projects/computing-education/computing-education-report.pdf Accessed [11th December 2020]

Thompson ET, Luxton-ALR, Whalley JW, Hu MH, Robbins PR, (2008), Blooms Taxonomy for CS Assessment Australasian Computing Education Conference 78 pp155-161

Thunė MT, Eckerdal AE (2018) Analysis of Students' learning of computer programming in a computer laboratory context. *European Journal of Engineering Education.* 44(5) pp 769-786

Torrance HT, Pryor JP, (1998), Investigating formative assessment, Teaching, Learning and Assessment in the classroom First Edition, Philadelphia, Open University Press

Trello (2020) [Online] Available From: https://trello.com/about Accessed 10th December 2020

Universities UK, (2018), Patterns and trends in higher education 2018 [Online] Available From: https://www.universitiesuk.ac.uk/facts-and-stats/data-and-analysis/Documents/patterns-andtrends- in-uk-higher-education-2018.pdf [Accessed:16th March 2019]

Wakeford R (2003) Principles of assessment in Fry HF, Ketteridge A handbook for teaching and learning in higher education pp42. Second Edition. London

Washer, PW, (2007) Revisiting Key Skills: A Practical Framework for Higher Education, *Quality in Higher Education,* 13:1, 57-67, DOI: 10.1080/13538320701272755

Webber G, Miller D, (2016) Progressive pedagogies and teacher education: a review of the literature Articles, [online] 51(3), pp.1061–1079. [Online] Available From: https://www.erudit.org/en/journals/mje/1900-v1-n1-mje03045/1039628ar/abstract/ [Accessed 7 April 2020].

Weegar MW, Pacis DP, (2012), A comparison of Two Theories of Learning – Behaviourism and Constructivism as applied to Face-to-Face and Online Learning.

Whalley JW, Clear TC, Robbins PR, Thompson ET, (2011) Salient elements in novice solutions to code writing problems. Proceedings of the thirteenth Australasian computing education conference 114 pp 37-46

Whalley JW, Lister RL, Thompson ET, Clear TC, Robbins PR, Kumar PKAK, Prasad CP, (2006), An Australasian study of reading and comprehension skills in novice programmers using the bloom and solo taxonomies, Australian Computer Society Inc. 52 pp.243-252

Whalley JW, Philpott (2011) A unit testing approach to building novice programmers' skills and confidence, Proceedings of the thirteenth Australasian computing education conference 114 pp.113-118

Wilson MW, Scalise KS, (2006) Assessment to improve learning in higher education: The BEAR Assessment system, University of California, Berkeley, USA 52 pp.665-663

Wilson TW, (2012) A Review of Buisness-Univeristy Collaboration, [Available online] URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file /32383/12-610-wilson-review-business-university-collaboration.pdf [Accessed: 2nd January 2020]

Windows. (2020). Explore Windows 10 OS, Computers, Apps & More | Microsoft. [online] Available at: https://www.microsoft.com/en-gb/windows/.

Winstanley, (2017) 'Supporting Students Learning with Cumulative Coversheets' in Hughes G. (ed.) Ipsative Assessment and Personal Learning gain pp. 65-84

Wong (2010) Motivating students to learn through good and helpful coursework feedback. The higher Education Academy Engineering Subject Centre

Yorke MY, (2003) Formative Assessment in higher education: Moves towards theory and the enhancement of pedagogic practice. Higher Education 45(4) pp 477-501

Yorke MY, Bridges PB, Woolf, HW (2000) Marking distributions and marking practices in UK higher education. Active learning in higher education vol 1(1) pp 7-27

Zhou JZ, Zhang JZ, (2017) 'Using Ipsative Assessment to Enhance First-Year Undergraduate Self-Regulation in Chinese College English Classrooms' in Hughes G. (ed.) Ipsative Assessment and Personal Learning gain pp 43-64

## Appendix A: Module Information

***CSC1021***

Semesters

**Semester 1 Credit Value:** 20

**ECTS Credits:** 10.0

### *Aims*

The module aims to give students an appreciation of the way problems can be solved using a programming language. It provides an intuitive introduction to the concept of programming and covers basic programming constructs including comments, variables, assignment statements, control structures (selection, repetition), expressions, modularisation, input and output.

This module provides a general introduction to computer programming focusing on the concepts of problem solving, basic programming constructs, and program design. It introduces how to design, write and test simple programs.

### *Outline Of Syllabus*

Programming Fundamentals – constructs

- Basic syntax and semantics of a higher-level language
- Variables, types, expressions, and assignment
- Simple I/O
- Conditional and iterative control structures
- Methods and parameter passing
- Structured decomposition

Programming Fundamentals – problem solving

- Problem-solving strategies

- The role of algorithms in the problem-solving process
- Implementation strategies for algorithms

Programming Fundamentals – data structures

- Representation of numeric data
- Range, precision, and rounding errors
- Arrays
- Representation of character data
- Strings and string processing

Programming Languages – virtual machines

- The concept of a virtual machine

## *Teaching Methods*

### Teaching Activities

| Category | Activity | Number | Length | Student Hours | Comment |
|---|---|---|---|---|---|
| Guided Independent Study | Assessment preparation and completion | 1 | 2:00 | 2:00 | End of Semester exam |
| Guided Independent Study | Assessment preparation and completion | 44 | 0:30 | 22:00 | Revision for end of semester exam |

| Category | Activity | Number | Length | Student Hours | Comment |
|---|---|---|---|---|---|
| Guided Independent Study | Assessment preparation and completion | 44 | 1:00 | 44:00 | Lecture follow-up |
| Scheduled Learning And Teaching Activities | Lecture | 44 | 1:00 | 44:00 | Lectures |
| Scheduled Learning And Teaching Activities | Practical | 22 | 1:00 | 22:00 | Practicals |
| Scheduled Learning And | Small group teaching | 10 | 1:00 | 10:00 | Tutorials |

| Category | Activity | Number | Length | Student Hours | Comment |
|---|---|---|---|---|---|
| Teaching Activities | | | | | |
| Guided Independent Study | Project work | 22 | 1:00 | 22:00 | Coursework |
| Guided Independent Study | Independent study | 34 | 1:00 | 34:00 | Background reading |
| **Total** | | | | 200:00 | |

## Teaching Rationale And Relationship

Lectures will be used to introduce the learning material and for demonstrating the key concepts by example. Students are expected to follow-up lectures within a few days by re-reading and annotating lecture notes to aid deep learning.

Tutorials will be used to emphasise the learning material and its application to the solution of problems and exercises set as coursework, during which students will analyse problems as individuals and in teams.

This is a very practical subject, and it is important that the learning materials are supported by hands-on opportunities provided by practical classes. Students are expected to spend time on coursework outside timetabled practical classes.

Students aiming for 1st class marks are expected to widen their knowledge beyond the content of lecture notes through background reading.

Students should set aside sufficient time to revise for the end of semester exam.

## *Assessment Methods*

The format of resits will be determined by the Board of Examiners

### Exams

| Description | Length | Semester | When Set | Percentage | Comment |
|---|---|---|---|---|---|
| PC Examination | 120 | 1 | A | 50 | Online exam in PC cluster - Daysh.Rack preferred. Blackboard OLA. Open book exam. |

### Other Assessment

| Description | Semester | When Set | Percentage | Comment |
|---|---|---|---|---|
| Prob solv exercises | 1 | M | 10 | About 10 weekly exercises (1 hour each) |
| Practical/lab report | 1 | M | 20 | Programming project (up to 20 hours) |
| Practical/lab report | 1 | M | 20 | Programming project (up to 20 hours) |

### Assessment Rationale And Relationship

The examination will consist of a number of "short-answer" questions to test the students' broad knowledge of programming and a section requiring the students to show greater depth of understanding of a number of key concepts introduced. The heavy weighting of coursework (50%) in this module reflects the importance of practical aspects of this module.

Study abroad students considering this module should contact the School to discuss its availability and assessment.

N.B. This module has both “Exam Assessment” and “Other Assessment” (e.g. coursework). If the total mark for either assessment falls below 35%, the maximum mark returned for the module will normally be 35%.

# Appendix B – Interview Transcripts

***Student Transcripts***

***Student A***

<u>What do you think assessment is for? What is the purpose of assessment?</u>

Assessment is a way to track progress. Sometimes I do feel assessment is kind of a test more than maybe feedback slightly. But that may just be my own personal preference because I do want to try and do the best I can so that may be a personal preference. That's the opinion I have got of assessment.

<u>Do you think assessment is similar now to what it was in high school?</u>

I'd say now that there is a lot more feedback to it maybe that is why I see assessment as more than tests because when I was in school it was more just you do this, you don't get feedback here is the result. When I am at uni there's a lot more feedback, for instance on the NESS system. That helps me to understand where I need to improve.

<u>What are your opinions on Ness?</u>

It's structured a bit strange, it's kinda hard to understand if you are just starting to use it. For instance with just the codes at the top with the nav bar, it could be put in a separate section for just subjects, so you could scroll down. Also, I don't like the idea of the side, where you have the exercise bar because there is just too much information at the side. It doesn't suggest which part each should go to because some of them are very similar.

<u>What do you think feedback is for, why do we use feedback?</u>

Just to give you an idea of where you can improve, It's constructive criticism. That's all I have on that.

<u>What kind of feedback have you experienced so far? For example written or peer feedback?</u>

On the ness system with written feedback, that's quite helpful with little mistakes that I have made, such as not including my date for instance on the report, which helps me to understand the way the uni wants us to structure our essays. I've also been given visual help as well such as in tutorials, been shown where I could have improved to get better structure in code. That is quite helpful I had been shown that I had a little background on programming, it helps me improve on my code so I understand it more.

Do you have a preferred method of feedback?

It depends, for instance on essay writing I prefer the typed feedback because it's more understandable for essays. But for coding I prefer the visual because someone is there to help you and if you don't understand something you can at least go back through it.

How do you personally use feedback? Once you have been given feedback what is your next steps?

Usually what I do is take the feedback, I usually write it down or I just try to remember it, but that usually isn't a good idea so I just try to write it down. And then from there when I am doing my next piece of work I take that feedback and try to make sure I stick to the targets which I set myself from the feedback.

What do you think are the issues with the methods of assessment and feedback and what do you think are the good things about it?

I like the programming for instance, giving us the idea of how to improve, but at the same time I'd like a bit more visuals for the control assessment at the end. Just because we get written feedback and some of it is pretty generic and doesn't give detailed feedback. For instance you get your marks back and the negatives can be detailed but sometimes the positives aren't and you cant see where you have really done well which could help just give you a bit more confidence boost. But for writing I like the writing and I like the idea of going on NESS for areas to improve because it's more easier to understand for essays and writing feedback.

What type of assessment have you came across so far? For example, you have already mentioned programming and essays, writing.

I've also done questions for maths as well. I like the feedback from it. But maybe the tutorials it wasn't, maybe it was my fault because I just sat there on my own to do the work. I should have asked for help. I dunno I think it's more just that idea of asking for help, I think the uni needs to get across that barrier. Because a lot of people still feel, even though they express it. Some people feel nervous and anxious and look down upon themselves. Programming, that's okay for feedback, I like the idea of the feedback for that one. What else do I get for feedback?

Software engineering, I know that's a module.

Yeah, that's the essays and also the presentation where I get the feedback through the Ness.

What has influenced the way you view and use feedback?

To be honest that hasn't really been touched upon that much, I do know for instance they do express it and how to use the feedback, but maybe from high school onwards they didn't really comment on it and there when I am doing my next piece of work I take that feedback and try to make sure I stick to the targets which I set myself from the feedback of doing it, and I am still slightly to my own way of doing it, and I'm using software engineering. It's just a bit complicated because the way I do my feedback hasn't really improved.

<u>So is that something you would like help on maybe ?</u>

Yes.

<u>So you mentioned previously when you get feedback you write it down and use that in your next assignment, you use that as a check list. Where did you learn how to do that?</u>

From myself to be honest, I was just doing it at home with IT work. Oh, history essay writing, that's where I got that from so maybe others who have done essay writing wouldn't get that idea from. It was mainly from myself and I have tried to refine it. As I have went along, but this point where I am at hasn't really improved.

<u>Finally do you think it is always clear why you are doing a specific assignment, why it's been set?</u>

Maybe not so much the exercises for programming. I feel like they were just to random. I know what they are trying to do, they try to set questions for the topics they are covering, such as comparisons in comparable. But um sometimes I don't think they consider people, they consider who may need help with other areas such as maths. For instance, recently I was doing one for the binomial efficiency and I didn't know anything about it because I didn't have a mathematical background. And um with that I just felt a bit clueless and about how to put it into a good structure.

At first it wasn't but then as it progressed and me and my friends went through it, there's more parts to being a computer scientist than what people see. For instance, people just believe that being a computer scientist is about programming when it's about being more confident and coming up with ideas and understanding codes that have to be followed such as ethical and behaviours. That's why I enjoy software engineering as well. Although it's a bit, it's not direct to computer science its helpful for when I'm actually in the workplace in the future. I did a work experience at HP and Proctor and Gamble and when I was there I saw one of the employees, he was currently revising and giving a presentation which he was going to give to others, I now understand why he was doing that and I understand the role a lot more now.

Well brilliant. Thank you for coming along for this it was very helpful

***Student B***

What do you think assessment is for? What is the purpose of assessment?

To check that you understand the content of the lectures. And then I have already forgotten the questions. Yeah, to check you have understood stuff from lectures, and that you can use it in kind of a real world setting. But it’s not really the real world is it?

Do you think assessment is similar now to what it was in high school?

No, but that’s because id say I did different kinds of subjects. Like in high school it was; IT and history and geography, so everything I did was kind of an essay. Or something like that, whereas computer science is more practical, the assessment is more practical.

What are your opinions on Ness as a system to help with assessment and feedback?

I quite like Ness, if it was updated. Not like in terms of what it looks like, but it can’t handle people getting to it. Also none of the lectures, because do you know how you are supposed to be able to click on an assignment and it is supposed to tell you what the specification is. Very few put it on. Like <Professor name> is the only person who uses NESS to its full potential. And if they all came out at the same time and far enough in advance, I would like that too.

So what do you like about NESS, you said you liked NESS?

Under coursework there is a summary, and if you go on that without clicking on a specific assignment it brings up everything for a year, well for a semester really. And it gives you the dates, and it should come up with the spec. Its useful for like organising and getting your stuff together. And I also like, on the week of, how it has in the corner ‘this is due’ and it has a little countdown, that’s helpful but kind of scary at the same time.

What do you think feedback is for, why do we use feedback?

I’d say it is to help you on the next assignment. Or to let you know where you are. Because if you just get a mark, that doesn’t tell you anything. Feedback tells you where you have gone wrong, and where you can improve. If the coursework is related.

Do you find the coursework is related a lot of the times?

No, very rarely. Because, well with HCI they were kind of related but it wasn't, well the feedback was for like writing a persona, but I never did that again, so the feedback wasn't that useful. Except that I am doing persona again, in web, no in mobile.

What kind of feedback have you experienced so far? For example written or peer feedback?

Got a lot of written. Not always that great, but you get a lot of it.

Why is it not always that great?

A lot of the time it just says insert comment here. Rather than like, there's that option on ness saying here's your feedback, but a lot of it is normally just insert comment here. Or like 'Good' but if its only good not excellent, how can we make it excellent. We've had a lot of feedback which again comes through ness, or via email, which says, as a class here's what happened.

Do you have a preferred method of feedback?

Written if it is good enough. Like I like personal feedback, personal feedback is more useful. Written, I feel like I would say I would like one on one feedback, but if I was in a situation with a lecturer where they gave me one on one feedback I wouldn't know my reaction. But yeah, personal feedback.

How do you personally use feedback? Once you have been given feedback what is your next steps?

Um I mean I always read it, and then most of the time its forgotten unless I need it again. For like web and mobile they're quite similar, so I've looked back at what the feedback was and was like oh I will do the same thing again or id change it. So that was quite useful.

What do you think are the issues with the methods of assessment and feedback and what do you think are the good things about it?

Um, Good thing, in third year, because it wasn't a thing in first and second year, feedback is very quick and it is marked very quick. So like in second or first year, by the time you got your feedback you kinda forgot what you were doing. But in third year it was quick.

So is that week turn around important for you ?

Um it's a helpful thing, definitely. So yeah its probably and important thing, or a useful thing. Id definitely let people, people would be like, networks is out and I would have completely forgotten what networks was about. Um bad, I don't like some of the assessments. I think that's just me not being good at them. Hm yeah, definitely the speed of feedback. I hate when its rubbish feedback though. When it is just good, what's good and how can I make it better.

<u>So if you got 75% on an assignment, and you got good, you would rather have some explanation of why you didn't get 100%, that 25% what was not good about that?</u>

Yeah I definitely like to know what I did wrong, I like to know what I did right as well, because its good to know what you did wrong. But I also want to know what you did wrong. Yeah that's a great mark and I'd be happy with 75%, but you should always be able to improve so how could I improve?

<u>What type of assessment have you came across so far?</u>

What do you mean by that?

<u>You mentioned in high school you did a lot of essays, now you do a lot of practical coursework. What other types of things have you came across? What do you mean by practical work?</u>

By practical work, I mean like on a computer. Anything with eclipse?

<u>So coding ?</u>

Yeah coding. I've done a lot of presentations. What was the question again ?

<u>What type of assessments have you come across?</u>

Presentations, we've done quite a few of them, team assessments, you don't really do that in high school other than uni. There's been a few essays but then they're not. Going back to another question, my dissertation is 15000 words and it's the first thing, we've never been assessed on writing, then suddenly its 40%. So they could fix that. Yeah those are the only types, presentations, written and then mainly like practical coding.

<u>What has influenced the way you view and use feedback?</u>

Oh what do you mean by that?

<u>Is it a particular person who has influenced that, or is it your colleagues and friends, a video you have seen or is it just something you have done?</u>

Yeah, probably what I have always done, or friends possibly. I usually compare feedback with friends but it is usually comparing how bad the feedback is. Like and comparing who gave your feedback. But yeah probably friends and what I've always done?

Is that a factor in your feedback ? Who gave it?

Only if your checking if someone else got bad feedback and your checking if it is the same person. Because people like, when you go to a lecturer saying this feedback is awful and I want better feedback or I want it remarked. It's better to have the back up of other people.

So is that a process, you as other students provide with the feedback, do you mention to the lecture that you have bad feedback ?

Yeah so our year has a Facebook group and we will always put results our out. And then the comments are always, I got really bad, who gave me that. Or and stuff like that. And then always compare and go to the lecture.

Finally do you think it is always clear why you are doing a specific assignment, why it's been set?

No. Like hardly ever. I can see it in like, sometimes they will explain it to you. And sometimes this is here is your coursework. Even in computing science where all coursework has a real life aspect, you can just look at your coursework and say, what is the point in this? I don't understand when this will be used or how I will use this in the future.

Ok, so would you like something that was like, these are the skills in this coursework., this is what the aim is,.

Yeah probably, don't know if I would read them if I was given them. It feels like it is something I would want but if I was given them I'm not sure I would read it.

That is the end of the interview. Thank you.

***Student C***

What do you think assessment is for? What is the purpose of assessment?

I think it so that both the student and the staff can see where they are at, at a certain topic. It allows the student to see where they are, not necessarily with regards to others. It does allow the staff to see where the class is and where that student is.

Do you think assessment is similar now to what it was in high school?

It's definitely a lot more computer based here. And typed work a lot of the high school stuff was wrote by hand. And auto marking like some of the exams and things like that although I do know people at high school now and for them it has also become a lot more auto marked too. Different to my experience but similar to others.

So what about assessment types, types of assignments and what you do and how it's handled and how you submit it.

Ok Well, with things like ness. We didn't have that it was more the teacher doing what the demonstrators do and signing things off. Normally we wouldn't have the small exercises being officially marked. They would just be an in class thing which didn't count towards a final grade. But the larger projects that's similar to the sort of coursework I would have had in my school.

What are your opinions on Ness?

I think it is massively underutilised and the ability to see a class average would help gauges where they are in relation to others. Possibly if it is a question based system viewing how many marks you got in that question with relation to the average, not only the staff but the students could see a trend in the whole. For example if the results came out with an exam mark the staff could see there was a particular weakness with this question and the students could see that as well., and maybe be able to suggest different ways it could be taught or content added to it to help for next year.

As a student, would you say that seeing the class average is an important part in being able to develop?

I think it is quite important in first year, because the percentage to pass is so different to A Level the perspective of what makes a good mark or an average mark is completely different to what we have experienced previously, so I think seeing how everyone is doing as a whole and how they are doing compared to everyone else would help people gauge themselves better.

What do you think feedback is for, why do we use feedback?

I think feedback is to support assessment. I think without feedback assessment can become slightly redundant, but knowing oh I did well on this I could have improved on that, it allows you to develop your skills and take that feedback on board for future assessment as well as just general usage. I think it is important.

What kind of feedback have you experienced so far? For example written or peer feedback?

Written and Oral feedback. We haven't had any official peer feedback yet, we have done some group work so it would have been nice to have had that officially done, unofficially a few of us were like good job well done. But nothing more formal than that.

Can you expand on what you mean by oral feedback?

So I have had meetings with my tutor and he's said oh you have X marks on this and Y marks on this, that's good. Or that's not good things like that. So just speaking to people sometimes passing them in the corridor. If I know that have marked a piece of work just asking them and having a conversation saying oh they liked this bit or they didn't like that bit. Or you could do this next time.

<u>Do you consider the demonstrations that's taken part in a lot for the undergrad, do you consider that interaction as feedback?</u>

Oh definitely especially when I've gone through demonstrations with my work in front of me saying does this look right, what could I have improved? Is this the most up to date way of doing it. With programming and things like that. I definitely count the interactions with demonstrators as feedback, and probably the most consistent and useful form of feedback, because they're right there with you and they can point to it and do it with the work in front of you.

<u>Do you have a preferred method of feedback?</u>

Probably with the demonstrators, if it's a larger piece of coursework or an essay or something I like to have oral feedback during which I can take notes which I could refer to later. But If it comes to I could only pick one it would definitely be the up to date regular feedback that we get from demonstrators in practicals.

<u>How do you personally use feedback? Once you have been given feedback what is your next steps?</u>

If it is oral, I will write it down, and if I have any questions, I will ask in the meeting then. If it's written I'll definitely read through it, where possible with the work in front of me. With the exams and things, like that because we don't get them back it's not so easy, but with things like programming and the larger coursework. I'll have the feedback up on one side of the screen and my project on the other. I'll go through looking for the bits that the feedback is relative to, and if I am about to start another project I'll probably go back and look at feedback on similar things and think okay I didn't do that last time, I will remember to do that this time.

<u>What do you think are the current issues of assessment and feedback?</u>

With feedback, there has been quite a range of marks for similar pieces given there were some pieces of coursework where some students got said well done for this and some students got, well not penalised officially but they were told they shouldn't have done the same thing that someone else got praised for. So it seems like there is a lot of variance over the marking amongst demonstrators and markers and things like that so I think that is a bit of an issue that needs to be addressed. But I think it is good that we do get feedback, sometimes it's extensive, sometimes it's very short. Um a lot of students complain about that. I guess for some of the similar tasks, for example we have just had programming one to five sign off. Most of us just got feedback saying good, but because we've already had feedback from demonstrators who have sat over our shoulders and watched it, that good was kind of redundant but they added it in anyway I guess. So when it's not too short its handy.

<u>What do you think, what do you think is good about assessment and feedback practices?</u>

I think it's good having those demonstrators then and there but its good that ness does have the option for feedback especially maths, because we got specific feedback for us and general feedback for the class which is kinda heading towards that whole class average idea that I kinda like. On the whole, I definitely think there should be some sort of counter marking for some of the pieces of work where there have been some discrepancies of marking, but I guess that's kinda part of it.

<u>What types of assessment have you come across so far?</u>

We've had short exercises which have roughly week long deadlines, we've had 24 hour exercises like the sprint.  We've had written coursework, group coursework, presentation, we're currently working on a video, so there is a lot of forms of different assessment, some of it independent some of it under exam conditions, some of it in groups. So I think we have covered quite a different range and types of assessment.

<u>Inside of them different types of assessment, you considered, coding, essay writing, have you had experience with anything else?</u>

Um coding, essays, the presentation and the research. Um the, we've done some reports which I class slightly different to essays. We had to do the heuristic analysis and the report on that. The literature review. So we've done quite a few different things, written stuff, practical stuff a few random things like the videos. Stuff like that. So yeah quite a wide range of stuff.

<u>What has influenced the way you view and use feedback?</u>

Probably the biggest influence was when all the way back in high school when we started doing coursework and you were able to go to a member of staff and say what would I get if I handed this in now and they would go, "you would get X grade for these reasons and this is how you would improve it" and that kind of iterative process of taking their feedback implementing it, giving it back to them and getting them bored of marking it and keeping doing that so you can get the best mark that you can so past results influences how I deal with feedback now, because I have seen the improvements that can be made when you implement feedback that your given. When its useful. So kind of previous experience really.

<u>So I think you experienced something similar to that in 1023 with <lecturer name> presentations you submitted one then you got feedback and you did another one which was marked. Did that process help you , do you feel like you did a better, you did better on the second presentation to the first?</u>

Yeah I definitely feel like we did um I think we got 8 out of 10 on the first one and 10 out of 10 on the second so whatever we did it was right. I think as a group we acted well on the feedback, there was a few comments on we should have a slide saying X and saying Y as well as a comment saying what they liked. So we made sure to do the things they liked again and to implement the things we didn't do the first time, so I would say we have definitely taken that sort of thing on board.

<u>Finally is it always clear why you are doing a specific assignment</u>?

Generally yes I feel like with some of the smaller programming exercises its more like a sort of hoop jump saying can you do X, its kind of a simple thing but we want to just check that you can do it anyway and with those things, I kind of, hoop jumping I find a bit annoying I do understand perks of it at the same yes, yes generally I think it is explained pretty well especially in 1023 because I do sometimes feel like the students feel oh this is silly, why am, I doing this. But I personally feel like its explained why we're doing each assessment.

***Student D***

<u>What do you think assessment is for? What is the main purpose of assessment.</u>

Well the main purpose is to test that we're learning the material that we are being taught and that we understand it all.

<u>Do you think assessment is similar now to what it was in high school?</u>

Its quite a bit different because I did maths chemistry and physics A levels, so most of mine were exams. Apart from a few labs., so it's a lot more coursework for me this year, but mostly the same.

<u>What are your opinions on the NESS system ?</u>

It works. It works, quite well for our purposes. I don't have any personal issues with it, or occasionally it is a bit difficult to find some pieces of information. Such as the marks occasionally.

Difficult how?

In terms that well. Not in the marks actually, but stuff like attendance is hidden away.

What do you think feedback is for and why do we use it?

To help us learn from our errors.

What kind of feedback have you experienced so far?

I've had a mix, in the first project I had a mix of what I did right, and although sometimes its not very informative in the case of programming occasionally. Usually it is quite good, covers all the things I did well. And mostly tells me what I did wrong. Although in the most recent programming one I didn't get full marks and their was no explanation of any point as why.

By types I was referring to written feedback, or oral feedback.

Right yes sorry. Um,. Yeah no its mostly been written feedback. I haven't had much verbal feedback, I haven't really spoken to many people about it.

What about demonstrations do you not get verbal feedback then ?

Um, I suppose. No I usually get good explanations of why things are wrong. Although that's more of the fault of HTML I feel. It's usually good, I normally get good feedback from software engineering when I speak to, um I think Matt Marshall, he provides good feedback when I talk to him about my work.

Do you have a preferred method of feedback?

Um, I do actually prefer one on one talking to people about, because then I can just straight up ask people about something I don't understand.

Ok, how do you personally use feedback, once you have been given that feedback what are your next steps?

Um, my next step is to look again at my work and uh first of all try to understand the feedback and see if I do. Usually I do. And then I do my best to keep it in mind when I do my next piece of work

What do you think the issues are with methods of assessment and feedback and what do you think is good about it?

Um I think it will be. What is good about the assessment is it feels like it makes you learn it well. The exams as a method are fairly standard. I feel like exam feedback isn't as good as it could be, we didn’t get spoken through the exams afterwards. I think mostly for coursework, the written feedback is usually good enough.

What type of assessment have you come across so far?

Well we have come across exams, we’ve had, some coursework such as, in software engineer professional, such as an essay, we’ve written code for programming, having submitted that code. We did a um sprint, which was basically an essay but we had a strict time limit doing it.

What has influenced the way you have viewed and used feedback?

Um, Most of the thing that influenced me, my Dad would go through the work with me.

Do you think it is always clear why you are doing a specific assignment?

Usually, I think its clear with all the um, program one is fairly clear why we’re doing it. Are all the software engineering ones? Ethics yes I can see why its needed. Literature review. Yeah id say usually, I’m trying to find any exceptions for that. I agree with that actually. It always seems clear why we are doing it.

That was all. Thank you.

***Student E***

What do you think assessment is for? What is the purpose of assessment?

I think assessment is two purposes one to actually to evaluate us and how we are doing but also to evaluate like how the module leaders are doing so they know if people are failing in one section maybe there has been a problem in the teaching of that one area.

Do you think assessment is similar now to what it was in high school?

Uh its definitely a lot different because our high school did have online exams they were all pen and paper. It’s different but it’s sort of the same in a way.

What are your opinions on Ness?

It’s ok , generally its alright, it does it’s job just sometimes it can be a bit complicated to navigate.

How so ?

You’ve just got like a lot of sub menus. If I want to go to a certain area, I have to spend quite a bit of time navigating to that area.

What do you think feedback is for, why do we use feedback?

Feedback as in like. Oh like to apply ways to hopefully improve in the future.

<u>What kind of feedback have you experienced so far?</u>

Mainly written, I have been talking to demonstrators in the our like practicals and stuff like that. Should I talk about what I have gotten in the feedback now?

<u>You can.</u>

It's just like some of the feedback I have been getting,. It really depends because some of the feedback we have gotten is things like, it's good. For written stuff which obviously isn't like. I cant get anything out of that if it's just, yeah this is alright. I kinda need more stuff about why its good, or where I could actually improve.

<u>So is why was it good as important to you as the where you can improve?</u>

Yeah

<u>Do you have a preferred method of feedback?</u>

I can't say that I do. Written probably because I can obviously go back and look over it, because if someone has written it to me I am probably going to forget what they have said. Yeah written so I can review in the future.

<u>How do you personally use feedback? Once you have been given feedback what is your next steps?</u>

Normally I just read through it, alright this is where I need to improve and just kind of, I don't really use feedback as much, normally because I'm just getting the same things over and over again. About spelling and grammar.

<u>What do you think are the issues with the methods of assessment and feedback and what do you think are the good things about it?</u>

Again as I said before just a bit more information about why this is good or why this could be improve with just a bit more detail

<u>What type of assessment have you came across so far?</u>

In life or in university?

<u>In university.</u>

Um mainly coursework and just the online past exams.

So what types of coursework have you had?

Written, had presentation, had eh, I would call it more practical bits of coursework, where we have had to design something like programming. Web.

What has influenced the way you view and use feedback? Has there been a specific teacher that has influenced the way you used what you were given? Or is it just what you have done?

Basically just a case of what I have done.

Final question, is it always clear why you are doing a specific assignment?

Yes and no, like sometimes in the sense, like in programming its, oh if you do this bit it will teach you this but for instance software engineering some of the practicals or some of the coursework I sort of understand why we have had to do it but its not really clear why.

***Focus Groups***

<u>What do you think the purpose of assessment is?</u>

**Student 1** Try and make sure that you have the capability to progress onto the next stage as well, because if you don't, then there's not a chance of you really succeeding after that so it's to try and make sure you can do what's essentially required and also if not then have you improved

**Student 2**: just to gage a rough idea of where you are. And how you've come to terms with the material.

**Student 3**: Id say that it could also be use by metrics for teachers to see if the way they're teaching has been understood and has been taken in.

<u>So compared to high school what do you think are the main differences (so high school A level college) compared to HE, what do you think are the main differences you have experienced so far?</u>

**Student 2** There's definitely less exams overall, we have like less exams now, and have six modules overall in this while, while in the past like in GCSE we have had like 10 different subjects. Some of them with multiple exams. Things are a lot more independent when it comes to learning and revising for the assessment, because you don't get. Although you are given resources and you are given opportunities to ask your lecturer and tutor I think, whereas when you were in school they were always there, as you had in class with them so you had more one on one help I would say .

**Student 4** Yeah I think its more individual, like work by yourselves of assessment. You cant really get through with just the lecture material for most assessments

**Student 3** The main difference for me, I don't know if it's the same for everyone else but em for my GCSE and A level everything was exam based, I had no coursework based subjects at all, whereas now here there's more coursework

<u>What do you think the purpose of feedback is?</u>

**Student 3** To see where you went wrong and see how you can improve in the future

<u>Do you all agree ?</u>

Yes

So what's your preferred method of getting feedback?

**Student 2** Personally I prefer, for example when we finished our first set of exams we were able to have an individual meeting with our tutor and they would give us feedback, like they would go through our scores and give us feedback whereas for our assignments we get feedback through ness, because it would be impossible for them to speak to every single one. But I personally prefer but I see why they have to do it the way they do.

**Student 4** Sometimes great feedback and it can have parts that I understand or parts that they are saying is wrong its like you know you're not completely correct but its hard to know exactly where you went wrong.

Can you expand on that, on how its difficult to know where they went wrong?

**Student 4** Erm I don't know. Sometimes they just say something quite vague I think. If you don't understand the assessment very well it can be quite difficult to understand what they're saying. I think. Because if you don't understand what they're saying and its quite vague you cant ask a follow up question like so quiz them on it. Because you've just got this paragraph and that's it because that's the only feedback you've got.

**Student 1** Id say the same, having amore interactive way of getting feedback would be preferable.

**Student 3** Just in person is best, even if its not, its usually not feasible.

So you brought up the system we use NESS what are your opinions of ness as a system for feedback delivery?

**Student 3** It does its job, for just giving you textual feedback, erm a lot of the answers are the same for different students suggesting that they just copy and past it maybe. Yeah, its probably not how they do the feedback I'm sure but its how it feels

**Student 2** Its quite generic I think when you get some of the feedback.

**Student 1** I don't think that's really a problem. It does its job

So is that across every assignment you feel like that or for example you have the 10 exercise in stage one programming if I remember correctly. Did you feel like feedback for that was generic? Did you get feedback for that?

**Student 2** don't think so, we just got a mark for that.

**Student 4** They're not sent off to demonstrators the demonstrators just check and then you get a mark instead of actual feedback,.

Do demonstrators talk to you about your feedback in those practical's?

**Student 2** They talk to you saying yeah good, its very well or its improved. Because obviously they mark you off and give you feedback on your sheet, and give you feedback on how yours is different from theirs.

**Student 3** I agree.

Do you understand the point of why you're getting every assignment that you get?

**Student 1** I mean I guess to make sure that we can implement what we have been taught in the lectures, to transfer out, to implement it into actual code.

**Student 4** I don't think, I think eh software engineering professional module has some assignments that have the most complained about. People saying the don't really help us and that just makes them useless really. So, a way  for us to spend our time.

**Student 3** Like we had sprints last semester which was in the middle of all our programming deadlines and it was looking at ethical disasters and I understand the point of a sprint is to make sure you can work under pressure but I just feel like it was a bit, I don't know unrelated.

**Student 2** You had to make a video with your group I didn't feel that was very relevant towards the course really, it was about technical information so it was good from a technical standpoint, but it involved stuff like filming, editing which is quite fun but its not really relevant to the course I feel like.

***Staff Transcripts***

***Teacher A***

<u>As someone who creates assessment what do you think its purpose is?</u>

I've got a glib answer for its assessing students ability. I guess generally speaking there's two different types of assessment. There's exams and there is coursework, and what you assess is different abilities. In an ideal world I would like everything to be coursework because I think its more, its better when knowledge and learning is applied through experience, but obviously exams are more guaranteed that students will work. Even if it's a more false environment, students will expect to remember things for that morning, for one and a half hours, however long it is. But I guess open book exams can combat that a little bit. And then you got em, you got I'm gonna forget the word, you've got formative assessment and summative assessment and then the key to good formative assessment is good feedback.

<u>What do you think is the most challenging aspect of assessment creation?</u>

Assessment creation? For me at least, finding something that all levels of students can have a crack at, have a crack at doesn't sound good, have a good chance of getting decent mark. If they put the work in, which not boring someone who needs challenge the really top end students but not completely out of grasp for the em weaker ones.

<u>What do you think the purpose of feedback is?</u>

Its to help students develop further really. I don't think it should just be you did this wrong. It should be you could do better by doing this, think about these three things going forward, or whatever.

<u>What is your preferred method of giving feedback to your students?</u>

I, for coursework I really like to give demonstration and give feedback there and then. I find the written feedback more of a chore, but I do it as well. For me at least that's its there and then verbal feedback, its not just, there and then feedback there's a chance for interaction back and forth during the feedback.

<u>Do you think students are using feedback in the way you would expect them too?</u>

Um. I never really thought of that. But now that I think of that it is quite an important aspect of it. Erm, how I would expect them too and how I would like them too. Um, I would like to think, I could think of certain examples of where they are. I can think of more examples where I have got no idea to be honest, there's that part of the feedback isn't there particularly for me at least.

<u>In your experience, do you think students understand the point of ever assignment they are given?</u>

On my modules I pretty sure they do. I'm sure other module leaders give just as much effort to explain why.

<u>Is that because you tell them what the point is or are the assignments very self-explanation.?</u>

Assuming they come to the lectures they hear some pretty extensive explanations of why certain particular assignments, well not so much the exam, but certainly the coursework I'm very clear on how it fits in. I suppose in the revision lectures I do cover why the exams are important so yeah,

<u>What do you think are the challenges of assessing programming?</u>

There's some pretty clear challenges around em, I can't think of the posh words, copying (plagiarisms) yes plagiarism, goodness me Friday afternoon. So there's obvious pretty clear issues about plagiarisms because there's only so many ways you can do a particular bit of code. I see, something I found in industry as I worked as a software engineer and in software engineer teams. Every single coder that you worked with, doesn't like anyone else's code. So in terms of assessing code, you have to bear in mind there are lots of different ways of, not structuring the code but presenting the code, coding styles. You have to be careful not to inflict that to train students as long as the code is properly structured and makes sense, and can be re visited in three years time that's the bit which matters really.

What do you think are the most important skills to test during a programming assignment.

Staff 1: A programming assignment should be around, there's choice of algorithm that really important at later stages, its all very well getting the first algorithm and optimising it and making it perfect rather than looking for a faster one. But if you consider different algorithms or opportunities or whatever, that's an important start. Erm the, and that ties into code structure, and as I said in the previous answer it's the clarity for someone to even the same coder to return to it two years later, so not just overwhelmed with comments, variable names, short functions just all these things which make it clear what its doing.

And the final question I had for you, is what would you say the differences are between a programming assessment and other forms of assessment? Such as essay writing/ report writing

Staff 1: Um, again its just, its back to the question you asked earlier to what is assessment for? As long as the student is clear and the module leader is clear what is being assessed so, writing a report an essay is clearly far more reflective than writing code. It’s hard to write reflection into some code. So there is other approaches that is better than getting them too, letting the students understood what to do, and really that’s why I like the feedback to be verbal because it gives you the chance to work in if the students really knew what they were doing. Its kinda of a poster, or presentation or easy it really give you a chance to drill down, I guess it’s the learning outcomes actually

***Teacher B***

As someone who creates assessment what do you think its purpose is?

It’s to, I would say it is to test the students understanding of the concepts. Or some of the concepts you have introduced in the class, and their ability to apply it to a situation that is introduced in the assessment.

What do you think is the most challenging aspect of assessment creation?

Trying to come up with something that’s an appropriate level of difficultly. Because sometimes, things that I have found in the past is that sometimes students have done very well on parts of eh, are we talking coursework exams or both. Sorry do you mean coursework exams or both?

Both.

So sometimes I have written questions that I thought were quite challenging and students have done less well in them. sometimes I have wrote questions which I have found less challenging and people have bizarrely struggled with them. It’s just that, I think the most challenging thing is that you are always running a year behind. So if this years, if last year’s cohort did alright on your exam and this year’s cohort did really well. So this year I might think I can make the exam a bit harder, but if these current stage one cohort are really weak they are paying the price next year. So I think, that the hardest part is getting an appropriate difficult and that you are always guided by the previous year.

What do you think the purpose of feedback is?

The feedback is to tell the students what they did correctly and the things that they did not do correctly on a specific assignment, I would say that both those aspects are equally important

<u>What is your preferred method of giving feedback to your students?</u>

What I normally do is, I write where possible a marking template, giving samples for particularly common. Firstly, for the perfect situation or aspects done well. Then for the common problems with the usual caveats you can add to when needed. And then just applying those and my own comments depending on the work. I will usually just do it chronologically for instance with the coursework, I won't say this is all the good and this is all the bad, I will say for instance this is question one, this is all your feedback for question one, this is the feedback for question two, so I usually do It chronologically within a question. I don't really have a set format for doing the positive first or the negative first, its often, a lot I think depends on the student work. So if its overriding good with a couple of minor things I would probably accentuate the positives. But if the student clearly has a deep-seated misunderstanding of something I would go into the detail of that, because it depends on the message I want the student to take away.

<u>Do you think students are using feedback in the way you would expect them too?</u>

No.

<u>Could you explain that?</u>

Or rather I will quantify that, Not all students. I feel that there are some students that genuinely view feedback as feedback. Reasons, things that they did well, things that they didn't do well and use them. I mean, what I want is, to give an example, if I have a student who has done the database coursework that just finished, I have got a student who has clearly misunderstood the concept of database design. I would want them to take away from the feedback that their table structure, that their approach to designing table structure, so that if they were ever approached, asked by someone to design a database for real, they could take that feedback and apply it in any real world scenario. I think in practice a lot of students only care about the mark and use the feedback to compare with their friends' feedback to see who got what, you know was it similar. It's just like my feedback is different to his and I got the same mark. Or my feedback was different to hers, and we got, we got different marks, but she had the same feedback as me or something like that. I think a lot of students just want to, they're just purely driven by mark and they will only look to see. I mean I had a student tell me once that they got it was something like 80 or 90 percent and they were complaining that most of their feedback was positive and I was like well, your work got a really high mark so you can expect that most of your feedback is positive because most of your work is really good. Depressingly I had, one of my demonstrators told me after dragons den a student complained we don't want feedback about our product we only want a good mark. So I think a lot of students use it as bartering tools to use it to get more marks. Whereas what I want them to do is use it and say well actually, I don't really understand X Y Z or yes I can do this if it's a student. We have more students than, sorry if I am going on a bit. We have more students than ever with anxiety and mental health problems, so if I have got a student with 90% it says there is no question one two and three, were done really well I hope the student thinks yeah I can really do this, so that's what I want, but as I say I think most just try to use them as bartering tools.

<u>In your experience do you think students understand the point of ever assignment they are given ?</u>

Um, not always. I think some students view certain ones such as reflective reports for instance. I have a strong suspicion that some students see that as a waste of time. Um some students view the CV as a waste of time. Which came as quite a shock to me as it's something that they all certainly going to have to use in the next stage of their career. I think there are some students that do and that there are some students that think I don't want to come here so I think that that yes basically there are some students that don't get certain assignments,

<u>What do you think are the challenges of assessing programming?</u>

I think the big challenge is that there is always more than one way of doing things. There is always more than one correct way of doing things, and you're always gonna have a student who's going to go a bit off piece but not necessarily wrong. So I think it is just recognising that and making sure that people look at the, I hate using the word weird, the nonstandard solutions to see if they are just wrong or to see if they are actually a valid way of doing things. So I think it's just the fact that there are so many ways to get to the right answer.

<u>What you think are the most important skills to test during a programming assignment?</u>

I would say that it's the ability to solve the problems as a whole. So it's not necessarily about the language and the syntax, you have this skill set in programming and it's about being able to apply that and also depending on the assignment or the skills as well. For instance the style rules and things like the, you know the whole set if you see what I mean so it's taking the skill set of both programming and the associated non-technical skills and applying it to the problem area that's what I would say.

And the final question I had for you, is what would you say the differences are between a programming assessment and other forms of assessment. Such as essay writing/ report writing?

I think there's a lot more. I mean I thinks it's a lot of students personally struggle with so in a reflective report they are often reflecting about how they did something rather than the quality of the finished product. You know it's like if I build system x , but I found it really stressful. I followed a process model that didn't work for it, but we managed to produce something that worked is very different from the actual production process. So I think it's a lot of the programming assignments are very much about the end product whereas some of the other pieces are not just about um, what was done, but thinking about how it was done and also it's a different kind of research needed. SO for written work its best practice to back up every point you make with something from literature if you are trying to make an argument for doing something a certain way. whereas with programming assignments they just have to do it if you see what I mean. There are some similarities so you never want uncited work just copied in done by someone else just copied straight in and you always there's always a problem solving element to both, but I think programming assignments are very much more about the what not the how or the background.

***Teacher C***

<u>As someone who creates assessment what do you think its purpose is?</u>

Eh, good question. So eh, it depends on what phase of developing the programme you are at. So when I have been involved at developing something from scratch the assessment is eh, a way of enforcing threshold concepts. So by integrating, by designing the assessment as part of the curriculum, you make sure that you have em your criteria or what they must know to pass the course aligns with what the foundations of that subject are. And then basically I would use that as rational for structuring the course. So, making sure that the assessment is valid, that they know stuff, em but so that students can use the assessment as guidance for structuring their learning. Em but yeah, on with that, then there is also threshold concepts which you also reinforce. Largely though I guess it's an indicator of performance, and to what extent a student understand the subject.

<u>What do you think is the most challenging aspect of assessment creation?</u>

Em. See, so the most challenging aspect, most challenging aspect I would say is, giving the students a fair opportunity to distinguish themselves within the remit of the subject

<u>What do you think the purpose of feedback is?</u>

Eh, this has changed a bit. So what, at least my understanding of the purpose of feedback has evolved slightly. I largely thought feedback was, the purpose of feedback was to help students develop their understanding based on the fact the assessment gives you an indication of their current understanding. If you can use that assessment as part of em, so I should have added, so you asked what was the rational or, what was the first question?

<u>As, the developer of assessment what do you think is its purpose?</u>

So I was really talking about summative assessment, formative assessment would be treated totally differently. Formative assessment is part of benchmarking students learning, and so yeah. That's an assumption from my part that I didn't make clear. I was only talking about summative assessment, if we were talking about formative assessment then the rationale is to guide the further learning of the student em, coming back to this question, which was?

<u>What do you think the purpose of feedback is?</u>

So yeah, largely it would be to help the student develop an understanding, but more so in a way that students seem to want more service-based teaching. It's about the, understanding the mark that they have. Em, pedagogically I would say it's about the student appreciating their, their understanding of the subject and parts they have done well and parts they have not done well. But as I say there is definitely an element now of making it satisfying to the student and meeting their expectations.

<u>What is your preferred method of giving feedback to your students?</u>

Em, not exclusively through NESA, and again it depends on the assignment and if it is formative or summative then yeah, you would have to give context for a clear answer. Em I think to make feedback more engaging it should be given through multiple formats and not just given through text-based version a time period after they have completed the assessment. To make feedback most useful em, its beneficial for the student and you have to make it engaging and you have to do it at the right time and that almost always not the time frame that's given for written feedback

<u>Do you think students are using feedback in the way you would expect them too?</u>

No

<u>In your experience do you think students understand the point of ever assignment they are given?</u>

Eh, do they understand the point of every assignment. Not every assignment but they understand the point of most assignments. Whether or not they appreciate the purpose of exercising those skills within the structure of the assignment possibly not. There is almost definitely an under appreciation of the way the concepts they are being taught are embedded within the practice that they are being assessed on. Em but do they understand the point of the assignment in terms of, I mean the purpose of the assessment is subject to the student as well. I mean the purpose for one student may be different for another depending on what they are hoping to get out of it.

<u>What do you think are the challenges of assessing programming?</u>

Ehh, distinguishing the functional and non functional criteria within what is programming. So, their problem solving is a correct solution, and if they have only given a correct solution, they are not necessarily being assessed on their ability to problem solve. But it depends, it’s incredibly subjective programming it’s such a broad concept. You could say that the dissertation is an assessment on programming because it’s based on an artefact and their ability to deconstruct a problem is one thing. Their ability to implement a solution is somewhat another. The way you constrain that emphasises different things.

<u>What do you think are the most important skills to test during a programming assignment?</u>

The same thing holds, I think. If its introduction to programming, then it’s the skills that need to be tests um are functional. Do they understand how to create a program to satisfy a set of given criteria, if its skills that they need to design and safely contrast a problem are much more complex in a way. Certainly more open. Yeah, the skills behind that are much more difficult

<u>What would you say the differences are between a programming assessment and other forms of assessment? Such as essay writing/ report writing</u>

Erm, again, it comes back to what is programming. Is programming being given an interface and implement the methods being given the input parameters and the methods, essential been given everything. Or is programming designing a solution to a broad problem, it completely depends, erm gimmie which one of those it is and ill give you an answer.

<u>So take stage one programming where the object is an introduction to programming, where the focus is the sort of how to program as opposed to best practice.</u>

Even the best practice, learning the best practice doesn't make you a good programmer. You can write flawless code and if its code that doesn't solve the problem , its till just, em and what was the question again?

Yeah of course, and that comes down to the pedagogy of it, if you are assessing based off the knowledge of the key programming terms of it, and essentially that's what it comes down to at a foundational level. You're not gonna be able to test them on like their creative abilities on that programming language, or the ability to evaluate certain things with them, within that very functional assessment. I think within a sense its closer to em, maths assessment or at least foundational maths assessments because at least it's more likely to lead to a correct answer, and if you get to the correct answer in the expected way, then its right, its more discreet then if you were to do something more, or higher up the blooms taxonomy.

***Teacher D***

As someone who creates assessment what do you think its purpose is?

Its purpose is to assess the students, to find out what they know and what they don't know.

What do you think is the most challenging aspect of assessment creation?

Being able to try to assess a large proportion of the material that has been taught without it being excessive, making sure it's, its reasonable time for coursework. Exams can be done within a reasonable frame of time. Getting a mix between the two is useful. They have to examine separate things. Exams are good for testing overall knowledge whereas coursework is can they actually do it.

What do you think is the purpose of feedback?

It's a chance for the student to get a chance to find out how they are doing and if they aren't doing things, aren't doing as well as they could do then how to improve them. Ideally, they should improve a little bit on how their doing from the feedback from the first assignment, they should learn from that. Just doing it is actually learning but getting some feedback is showing how you could do things better. Useful particularly for the sign off exercises, particularly where they have actually explain to a demonstrator and show exactly what it is they're talking about. They're guaranteed to have spent some time with someone who has looked at what they have done closely.

What is your preferred method of giving feedback to your students?

Via ness I guess is the overriding way. So I guess that is my preferred method I guess.

Do you think students are using feedback in the way you would expect them too?

I do not know. In in the past I've given out model solutions in 1021 particularly sometimes in 1022. I have given out model solutions to see how what they have done compares against. Particularly the first bit of coursework in CSC1021. I tend to go through a sample solution so that they can see how its done, not perfectly but well.

In your experience do you think students understand the point of ever assignment they are given ?

Probably not. I would like to think that in programming terms I actually only set two marked projects in first year programming. One in the first module and one in the second. I would hope that they would see where the stuff has been covered in lecture notes is actually put into practice. That's the idea of the projects so that they can put into use the things that they use in the lecture theatre. So I would hope that they can get most of the points but probably not all of them.

What do you think are the challenges of assessing programming?

Erm being able to set something which is not so large that its going to take an enormous amount of time to do, but not totally trivial but getting the right size of task to give them, and also examining or assessing a lot of the thing that we cover during the lecture, it's difficult to cover.

What do you think are the most important skills to test during a programming assignment?

Um it depends on what the assessment is. If its an exam then you're testing them on different things. You're not testing their ability to write programs in an exam you're testing their general knowledge of what students will know and what students wont know. Whereas in coursework you're saying can they actually write programs themselves from a specification which might be quite specific or might be a little more general. The further on as a programmer the more general the specification as they ought to be able to cope with that, and use their own reasoning to decide what is reasonable and what is not.

And the final question I had for you, is what would you say the differences are between a programming assessment and other forms of assessment. Such as essay writing/ report writing

They're assessing different skills, essay writing is more sort of do the research first hand and then putting it quite succinctly, whereas programming is actually building something. It's just producing different artefacts at the end of the exercise. I think that is the main difference. Trying to find out can the students do it. You might also say the same about maths exercises as well. It's a slightly different technique but they're trying to show they can do the question that has been asked. In a general sense that is what all assessments are saying. Here is something to do, can you do it? And if you cant can you find out how to improve and that also you can.

***Teacher E***

As someone who creates assessment what do you think its purpose is?

Ok, two fold, um, one is to assesses, to measure how students are learning. How they understand the material of their course, the individual measuring of their understanding. But it is also to help me as a lecture to understand how well the material is being understood in general. So if students are generally failing its not the students fault.

What do you think is the most challenging aspect of assessment creation?

Most Challenging. Ok, em I guess fairness is an important bit, so making sure that um you are, when you are assessing what the learning outcomes of the module actually are. So um if I am doing an assessment say it's a report, I really want to know how well people have researched into the literature or um that they have understood some part of the some of the other skills, that they have supposed to pick up. erm and you want to make sure that the assessment you are writing is appropriate and you are measuring the right skill.

What do you think the purpose of feedback is?

Ok so, the purpose of feedback, so that a student can understand where they have gone wrong and how they can improve in the future. The main purpose, em being able to understand where they have lost marks, what the mark actually corresponds to and what was expected.

What is your preferred method of giving feedback to your students?

I ok so, there is what I do which is em, a few notes delivered via ness which explain what they have done well, what they haven't done well and combine that with a sample answer that basically gives them what I expect. So, a few lines on the highlights on what they could have done better to get the marks, and compare them to a sample answer. Now my ideal way given more time um, I would maybe give them recording video feedback or something like that give them a bit more detail but the most time effective way seems to be to do it that way.

Do you think students are using feedback in the way you would expect them too?

So it's a little hard to measure um, typically a student comes back and says this is the feedback I got and this is the feedback somebody else got an we got a different mark and um I can't see what's different. That is not what feedback is for. It's not to defend the mark against what somebody else got so there are those cases, I think the majority when students talk about what they expect from feedback and how they are using it, it seems to be what I was expecting it to be used for.

<u>In your experience do you think students understand the point of ever assignment they are given</u> ?

No probably not.

<u>Can you expand on that?</u>

Well I suppose, there is evidence in that from the questions you have coming back from the students, why did I get this mark? I did this fantastically well when didn't I get that mark? And well that may have been it wasn't what we were measuring. Um and if we go back to what the purpose of assessment is, it's got to be formative or its got to be summative whatever, they don't necessary understand that kind of terminology.

<u>So do you think that is the case of them not reading between the lines or do students want: these are the learning objectives, this is the point of the assignment, these are the skills you will have once you have completed the assignment.</u>

We do, do that. Well I do, um in the specifications I will say, these are the objectives, this is why we are doing this, and these are the skills we are trying to develop. Um students don't always know that, students don't always listen if you tell them that during the lecture. I guess, this is all about assessment literacy. It's about them understanding why they are being assessed, to develop and it probably can't be done in a single piece of coursework, its more something that should be done throughout the entire course. You start off in first year which is more about understanding their journey and where they are about to go.

So I also teach Java to the MSC class, so I may think more of that as active programming. There are a lot of problems with VDM is that it isn't really programming although one of the major challenges are that people approach it as if it is. So part of the problem here is getting a student to step back from the concrete ideas they have about solving a problem and think about, you know I am trying to characterise and describer a solution but not the programmatic steps. In teaching it at the Java MSc level um challenges, it's a very diverse group so some of the students who are in it have had years of experience programming. Some of them have never even written a basic hello world, so its keeping everyone, not letting the top end get bored and trying to avoid the other getting worried and scared and confused. We do make use of small tool support in the form of small exercises there to try and support and build up confidence and em I think that is the challenge, having enough of the small confidence building exercises but perhaps some stuff at the other end that interests the others.

What do you think are the most important skills to test during a programming assignment?

Hm. Problem solving, em I think it's those sort of, that level of skills rather than the, I know a nice way to do this. You've got the two things, the how do I come up with a solution and how to I code the solution and how do I make use of what the language provides me with and how do I demonstrate how it works and all of the rest of it. What's most important? I originally said problem solving but you have got to get that basic level of skill right so you can achieve the problem solving, so I guess they go hand in hand really.

<u>And the final question I had for you, is what would you say the differences are between a programming assessment and other forms of assessment? Such as essay writing/ report writing.</u>

There is a greater range for creativity and um there's a lot of other skills that are being assessed with report writing, presentations and the rest of it. In terms of finding the one specific area is originality, you know plagiarism and stuff. Spotting that within a coding environment as against reports and presentations, very different. In some instances its more vague and cases there are not that many different ways of solving problems and writing code. Whereas with report writing, it is very obvious if it has come from the source. Em I'm gonna stick with that as the main difference, there are loads of differences but I am going to stick with that.

***System Interview Transcripts***

***Student 1***

<u>The first question is with regards to the system did you use it?</u>

Yes I did. I did use it when you just sent the link and then about a week before the exam as a refresher.

<u>How confident do you feel with your programming from the start of the semester?</u>

Probably knocked down a few pegs because I think I know what I'm doing but then when I get feedback from my work it doesn't really look like it so I guess I'm feeling a little bit probably worse than I was feeling at the beginning of the semester.

<u>With regards to that feedback does that feedback give you clear instruction on where to go next or what you did wrong?</u>

Um more or less. Some of the feedback kind of seems do this in university whereas in real like you would kinda probably go about it another way, which is sort of counter productive I guess.

<u>Do you think the system helped you with learning the processes of programming?</u>

Yes. Actually I think doing the quizzed helped me with the principles I used less when programming myself so I had to have a refresher on it. So, it did help me go over it on some concepts that I may have forgotten a bit.

<u>Were there any parts of the system that helped especially for that, so the content, or the features of identifying where you went wrong in the quizzes?</u>

I guess that would probably be with the quizzes to see where you went wrong and trying to do them again to see what they actual output should be of the programs.

<u>With the feature of receiving the correct answer to the question you got wrong, how often did you use it?</u>

Um I tried to jot down the question that I got wrong, so that when it came again in another quiz I tried to look and see what answer I chose and tried to figure out why that could have been a wrong answer.

<u>Do you feel like there is anything you could still be improving on with your code?</u>

Yes. There are still a couple of concepts that I still feel wobbly. Qith exceptions I still just do it because I need to, not yet completely I know how, and I know why. I mean I know why but still not how I guess.

<u>We mentioned some specific skills and areas that you wanted to improve on at the start, do you feel like you managed to get some progress with those skills?</u>

Um with the collections part yes, because I definitely believe I said something about lists and maps, with exceptions not so much, as I said I still feel kind of wobbly about that.

<u>Is that something you are going to work on in the future?</u>

Yes

<u>With regards to the system itself what features do you think would benefit the system. What do you think could be added or changed?</u>

Um I don't know its probably just me, but getting the more difficult questions that aren't covered in the curriculum might motivate some to look it up but to me it sort of even seemed a little demotivating because it was sort of like I think I should know this but I don't. So I guess in a way it made me feel a little worse, so it made me feel worse not getting it right.

<u>What about features, instead of content?</u>

Um I guess some part of the goal system could maybe linked with how you use to how you view the content and use the systems because right now it's just add a goal and I guess at some point you will forget that you have set a goal. Ao have a reminder that you have set a goal, try to look into it, try to better yourself at it. I guess that would be a thing I would.

***Student 2***

<u>First question, did you use the system?</u>

Yes I did use the system. I used the system during the revision process, so sort of after lectures. After revision lectures as a means to help revise for programming and the reason I wanted to use it, um was partially because I was curious to see how the system would work um and because I wanted to help out with this experiment, but also because in terms of programming, aside from making up your own exercises there is very little in terms of revision materials. Um so it can be quite difficult to come by different materials that you can use to help revise, because there isn't a lot in terms of, resources. There's' notes you can read, but obviously with 1021 here being an open book exam, it makes it even more difficult to work out what you can revise, because all you have really got is the practicals which you have already completed, so yes it can be, that was why I wanted to use the system.

<u>How confident do you feel with your programming skills?</u>

Is that in general or just after the system?

<u>In general now, keeping in mind the system, we will come back to how the system helped you.</u>

Right, ok. In general I am relatively happy with my programming skills, I'm happier than I was before starting university so. I would say definitely since doing the practical's in the sessions for programming and being able to check off being with the demonstrators I have been through is probably the main area of my confidence in programming, because when I started back in September I wasn't really confident in programming at all and it was something that worried me quite a lot. So that was something that helped me along the way, and I think that was staggered in and um, with regards to the system, it was able to help reassure me that I was able to know what I thought I knew, about programming also.

<u>A similar question, following on from that? Do you think you have improved on the things you identified on the first questionnaire?</u>

I would say um, I think I probably have, but a large portion of that was probably to do with the work that was actually, that was around the time the second coursework stated, so probably I would feel that I increased my confidence over doing that coursework because I had to try a lot of new techniques since that first meeting . Sort of on an impulse that I hadn't expected to need to learn quite so quickly. And I think, so yes I would say my confidence has probably built since then.

<u>Do you think the system helped you improve?</u>

I think that the system was, interesting and by that what I mean is, I think it helped reinforce what I knew, weather I could say it helped me improve, necessarily, however there was a couple of questions necessarily where being unsure of what I was doing, getting them right or getting them wrong, helped me realised by experimenting, thinking if I put that in to get the answer wrong. I think in that regard it could have helped me improve because it could have helped me re enforce how different concepts worked, I would have said, um so yes to an extent it probably did help me improve I would think.

With regards to your programming skills do you think there is anything you think you could still be working on?

I think, probably in terms of programming skills, yes there are, it's quite difficult to work out exactly what they are, given how quickly we sort of move through different topics um, but probably to do with just making sure that I keep revising the fundamental skills, and how to write the program syntactically correct. Um, and it would always be just sort of ensuring that I can understand, sort of, I think as I have come across them it has been things such as abstract classes and interfaces which I know we are just starting to cover again now, but covered at the end of 1021. Something I think I could probably do with improving on still.

With regards to the website itself were there any features you thought were good?

So we have got, what I really quite liked about the system, was your sort of had a topic which was written and it wasn't a large topic so they were quite bitesize in terms of being able to fit them in. So you could read something, and there was a quiz that was associated about it, was that it was ten questions and I thought ten questions was a very helpful number, because it was enough to sort of go through every part of the content that you had just read from that section of the course that you had written up to the system. But it wasn't so much that you had to devote ours to it and what that means is you have the ability to think it's a system that you can actual use, rather than have to devote specific time to look at, which is something to say that for people revising is a very big factor because you don't want to have a system that might be very effective but takes up too much time in one go. Whereas in this case having ten questions was hitting the right balance between not actually testing people ability vs doing it too much.

The final question I have which is a follow on from that what features do you think could be put in to improve the application?

I think so, there were a couple of improvements that I had thought of, one of the immediate ones would be a closer linking to topic to quiz, so to remove some of the confusion and improve the flow of the website a little bit, where when you have completed a question it took you to a specific quiz. So if you were going to click on that quiz you knew you were going to click the right one, um may be a potential area for improvement, because then if you were doing a section it would just be, it would improve the continuity somewhat. And it would help with people who might not be sure exactly, if they're not quite sure on the topics having to then go through a menu to then select what topic they were looking for may not help to improve their confidence, whereas if they were linked directly it may be able to an improvement because the questions will then be just a direct link to what they have read.

<u>The following transcriptions were notes that this student had on the system and wanted to pass on.</u>

I'll run through these and if you would like I would be more than happy to share a copy of these with you also um so. One of the things that I have been looking at was the first thing I noted down, was just generally, I thought it was very nice to have a system that was designed specifically for programming revision because I have said previously there just isn't the resources out there. So um having access to such as system was just nice to have, because I think programming revision was just something you've got to know a lot of the time, and so if you're trying to put aside time to revise it, it can be really tricky to try and decide what should I do, because unlike other subjects reading your notes isn't necessarily just the perfect answer because its open book so if you that its good to know where things are but it doesn't help you with programming or what the output may be it may be. So for that fact to think that was there something I could use was a nice feature, it was nice to have, it was very re assuring. Um, the thing that would be something thought is, topics that you just need to know for the um, when you need to know them, a small issue that I found was there was some small grammatical errors within some of the questions, which made it a little more tricky to decide some of them. It was I had an example image of a program running here from one of the questions that I picked up a slight issue with. It just makes them a little harder, so when it comes to understanding the question just having that flow and the wording specifically, on them just helps to make sure that they're easy to understand um one of the ones that were looking at as well so another small recommendation. When it was looking at some of the larger programs such as one that's here, that is showing a class I believe two classes, um if it was asking for an output for a specific line it may be useful to have some line numbering going on so you can follow them or even some highlighting the line. Just to draw the users attention, particularly from an accessibility stand point I would say just for someone who would have to read through the whole program which you may be instructed to do anyway but it would make it a little easier I think. If it went just given to you and you had to, if you had numbers at least I think that would probably be quite useful

We have so, the one of the other things I picked up on perhaps was, as I mentioned before the UI for taking a quiz isn't quite as entitative as it could be. And what I mean by that is um, for instance it looks almost like a form here, so when it says test your skills I ended up thinking this was the first question and I got a little confused. To what this was, when it was actually a form that I wanted to select one, if these were perhaps buttons so I know they were taking me to different quizzes um because other wise I looked at this and thought it was asking me something which I didn't quite know because it said test at the top, I didn't realise I was just choosing what I wanted to test my skills on. So that confused me just a little bit.

Then there were one or two questions where I wasn't too sure if the answers worked quite right or not, I thought the answers didn't look right to what I thought they should be. Here is an example here was one of the questions that was given, which told me I had the incorrect answer which told me I had the incorrect answer, I tried running the code in eclipse which give me the answer that I had gotten so I wasn't quite sure if that was something I had got wrong, or if there was an issue to how I was picking up answers but it was just a slight discrepancy to what eclipse gave me as an output and what the system gave as what the answer should have been, and I wasn't quite sure why that wasn't exactly. There was also the thing of, yes I did I discover this couldn't be the case, but I did notice you could tick more than one answers on but there was also. Ah yes, so this was actually a bit of feedback that I should have mentioned before when it was a potential improvement was when you finish a quiz it may be quite useful to see what the questions were that you had got wrong I think it did that on some capacity, but if you could have seen what the correct answer was and what your answer was.

If you would be able to look at it and say, I answered this but the correct answer should have been this after words now admittedly, ill agree that's not always the best strategy for some people because they might just do the quiz look at the answers then fill them in incorrectly. Which I think in my case I quite like that as a feature, because I will give the quiz an attempt but then what I want to do is I then want to look at what the correct answer should have been to see if I could work out how to get to that answer in the future and see if I made a mistake so that would be just something I would have liked

<u>There was a section on the quizzes that showed you the correct answers, I'm not entirely sure I told you what you did compared the correct answer, but it did give you the correct answers part if you didn't find that however that is a problem with the design and I need to highlight that more and I need to highlight that in a more applicable way.</u>

Yes because I hadn't perhaps found that because what I picked up on just looking at this here. I just picked up on the fact that it was the um, the just completed quizzes so I think it would be useful for the answer you got wrong screen to label each question clearly. Oh that was another point labelling each question clearly, e.g. labelling question 2 and leaving a space below, so you could tell when one question moved to another would have been perhaps useful and knowing which question it was and, yeah so I had said knowing your answer as well. Ao I don't know if it had actually saved and displayed saved a copy of your answer somewhere but that was something I couldn't find. A copy of what my answers were to know what I had put down originally and um, there were just a couple of tiny formatting errors I discovered on some of the quizzes. And then again the final point that had was the fact that I thought ten questions was a very helpful health number that was quite good in moderation. Um one other thing that I might recommend to improve the system would be perhaps get it so that it could work on mobile or whatever, because it was the type of thing I could imagine someone doing while they were waiting for a bus or something to be able to just test their skills oh how was I on strings and just quickly do a string while their waiting.

Overall um definitely a good system and one that I think has potential. I think there were a few things that I perhaps would have personally liked to have seen worked on but I think it was definitely a system I would use again given the chance.

***Student 3***

<u>The first question I have, did you use the system?</u>

Not very much.

<u>Why did you not use it very much?</u>

Um because I generally just went through the notes because they had similar examples. I used the system every now and again but half of the time I forgot what the link was to it.

<u>How confident do you now feel with programming?</u>

A lot more confidence.

<u>Ok would you like to elaborate on that?</u>

I feel like I generally understand the majority of the concepts now, and can actually apply them, but I still feel I need to improve the debugging part. Because lots of the time I can write the code but at the end there is problems that I don't know how to fix them

You said you didn't use the system much but you used it a bit, do you think when you used the system It helped you understand the principles of programming?

Yeah, I think the content bit was really useful because it summarised and had an example, which was perfect. Especially for quick looking.

How about the other sections such as the quizzes or goal setting?

The quizzed were good, they're quite hard because they certainly tripped me up. But I think they're probably really good, that if I do them a lot id get really good at understanding everything perfectly.

Think back to the first questionnaire we filled out and the things you wanted to improve on. Do you think you have managed to do that?

I cant remember to what I said.

You said you were fine with the syntax, it was applying it to the big picture so writing code.

I feel like I'm a lot better at that yes.

Do you feel like there is anything you could still be working on? You mentioned the debugging

The debugging and using abstraction and interfaces.

With regards to the application and you saying you didn't use it often, what would have encouraged you to use it?

I feel like turning it into a mobile app, because then you could use it the same time as coding or link it to blackboard.

What about outside of the app. Was it a case of there was an extra thing you had to log into and you didn't want to use it, or was it just you focused on course provided material.

Yeah I just generally used the coursework notes.

***Student 4***

Did you use the system?

Yes

How confident do you feel with programming?

More confident then when I started, I feel that the issues I had to begin with erm there not as prominent, like they're not inhibiting me anymore.

<u>Do you think the system helped you with learning them or with learning the principles of programming?</u>

Erm, I think, it helped me learn the principles of programming yea. Erm not really like in terms of programming itself, but it did help with like the topic of arrays and stuff like that. It cleared that up really well. Some of the other ones not so much.

<u>What about the system do you think helped with that?</u>

The content was a bit easier to digest than the lecture notes.

<u>Since you took the first questionnaire do you think you have improved in your programming skills?</u>

Yes I would say so.

<u>Could you expand, what skills do you feel like you have improved on?</u>

I feel like, it was mainly the arrays stuff, you know the different methods you could call arrays and lists and immutable lists and it helped me with that. Knowing which ones to use yeah.

<u>Is there any part of programming you feel like you could still be working on?</u>

Erm, I would say, exceptions.

<u>What features of the website did you like?</u>

I think the goal feature was really good, it allowed me to keep track of my progress, because I didn't just do all the topics on one day, so you know coming back two days later I probably forgot what I did before. So having that goal and knowing what I did before, and I'm going to this, that helped me a lot. I'd say the home screen itself yeah that was good, knowing which I haven't done, and the quizzes of course.

<u>Is there any features of the website that you feel could be improved?</u>

I did find some of the quizzes hard to complete. I know it was kinda supposed to be subsidiary to my, it wasn't supposed to be my entire revision but I did find that, them hard to answer.

<u>Is there any features you would like to see in the website that wasn't there?</u>

No, I think it was quite good.

***Staff System Transcripts.***

***Staff 1***

<u>So can you describe to me what your understanding of Ipsative assessment is in theory?</u>

Ipsative assessment, so it's sort of, assessment is the idea of improving your current level rather than, for example, standardized testing. So, where everyone has to meet at the same level type of thing, so it's more about how you've progressed more than actually the final results. Yeah, so important. The key item of its assessment is to consider the learning game that the student has as opposed to meeting a set of criteria.

<u>So as a concept, do you think it's a sort of assessment would be beneficial to students</u>**?**

I think it would be if the students, especially for independence so and at University level, because the idea is that the students would then be aware of what their current level is, and then my hope is that you know they would if they focus more on the learning you know. And the fact that they are progressing then rather than the marks even though they are important. But at least at first year, it would be nice for them to actually gain that confidence. You know that kind of thing if they focused on what, they've how they've progressed, rather than the actual marks.

<u>What about from the staff perspective? Are there any benefits to the staff perspective?</u>

As I said in terms of independence, it would be you know so that the students, so from transitioning from school to University, you know. So, we always find it difficult for students to kind of learn how to be independent and how to go about their own research and so on and so forth. So that would be a benefit. However, I do see some negatives because the other problem we've also got is that we need to come up with a mark for them. So basically, you would be, instead of coming up with one coursework spec would have like 300 coursework specs, so that would be a difficulty there, which might be the next question, sorry.

<u>And what about the negatives for the students? Can you think of any negatives on that part?</u>

So the students always, so I said like so, they're used to more of a kind of like summative and formative, more summative rather than formative kind of assessment. So they are so used to comparing themselves with others. You know what I mean? So it would be. It would be a different mindset that they're not used to so and they. Regardless, you know it's how the world works to a certain extent, and if they're so, if they're focused more on the competition and you know getting the job and the marks and so on and so forth. Plus, I wouldn't assume. I would assume like employers, would  also still prefer that kind of like. Here's a mark based on what level you're at. You know what I mean? So just because it's easier rather than it is harder to mark that kind of like that. That growth then it is, you know, saying right here's a test. You will need a 20% of it. Therefore, you still need to grow, type of thing you're not ready for our business or something.

Do you think a system such as C:IA helps you gain an understanding of the concepts of programming.

So in terms of, so, I haven't looked at the content obviously, but in terms of the quizzes and stuff, I think it would be useful. I like the idea obviously that a student sets their own goals and there they you know they're able to sort that out, but it just depends. I mean, it has to be embedded within a module. It can't not be, because I find that typically students, so they will do some, you know, because they've got their summative assessment, they need to put more time and effort into doing the summative. So you know, so something like this, where if it was not part of their summative assessment, they wouldn't necessarily put the focus on, at least, at least not at first. So, I, they will do either one of two things, they'll do it from the beginning, and then they'll stop doing it. Or they might do the opposite, which is it will be too late. You know that they're not doing so well. They'll try it. And then because they might not see the short term benefit, they might drop it again. So that's it's one of these things that you do have to keep on, trying over and over again and kind of using, you know, and I think that that would be difficult for some of them.

Do you think CIA provides a clear region for self-reflection regulation and growth?

I think so. I mean, it's good for reflection, so I mean the idea that you've got these goals and setting them looking at them. I mean, it's just whether or not I mean there is such a thing as like unrealistic goals, so I mean something, so one of the goals I did was is need to get a better average in their quiz. Now you know a better average by one is a good is showing progress, right? But if a student says I need an average of 10 and yet they've already had zero at some point, it's very unlikely they'll ever get 10 unless they were to, to delete that so they need to understand, you know that there is, You got smart goals and realistic goals and it just depends on it's it's not something that you do, it's it's something you learn overtime. You know if you set yourself unrealistic goals it will be more likely to unmotivate you then actually motivate you, which is a shame, so maybe it might be worth showing some examples of goals or even maybe some materials related to that on how to set it all.

Might have been good to add to the system actually so that you know so, smart, realistic how to get you motivated, that type of thing.

<u>So the next question focuses on the quizzes in the quiz section. When exploring quizzes, do you find the question super for first year students? Are they, Are they doable? Are they too hard or the impossible? Then managing but challenging.</u>

I think they're good. I mean, as I said like so it depends on how they go about doing them. So if I were to do this as even as a student, what I would have probably done is tried the quiz. You know first without trying like running any of them, that type of thing. And then what I might do then is then try running them type of thing afterwards and learn from that. So the you know there was a couple of questions though, where as I said I queried whether or not it was actually the answer matching the question so that that obviously does not help. So there might be what you maybe could have added to the system. Was is rating the confidence of a question you know so you know 'cause if it says that this is wrong, yeah, then you know. But it's unfair to mark if it is wrong. If it actually, if the question itself is wrong, so actually maybe what you could have done is say, I think there was a system actually, one of the third years have done this year. They have this thing where actually you could. You could rate the question you know you could say actually this is wrong and then that would be removed. So that also means of course that your quizzes might be then become nine questions rather than ten, which skews your average. And so on and so forth. But you should be able to report the question is wrong. And then that would go somewhere where it would be fixed, and so on and so forth. And that shouldn't affect their statistics. So I mean they could do that with all ten questions, but they're not going to learn anything from that, are they? So, but there still needs to be a way where they could report the question, they think is incorrect, and if we find that it is correct, you know, so from a staff point of view, then we can say, well, actually no, this is correct, you know, and bring it back out again.

<u>So we touched on these next two questions regarding Ipsative as a concept, but what about using a system of the system itself?</u>

<u>What do you think the benefits of a system would be?</u>

I as I said I like the system and it's like putting it, It's just putting it in one place type of thing so as if it was embedded within a module, It would be useful so that they can actually kind of like make you know. It's as I said, they need to see the benefit of it, otherwise they will not try it, it could be useful for maybe kind of like exam revision type of stuff. But the thing is is we don't do exams much anymore. Funnily enough so, but in terms of so, it's type of stuff we now do is coursework based, so I wonder how useful they would find this now I, I'm uncertain. We obviously provide quizzes though in you know on Canvas so, so and having different quizzes every time. It is useful, you know so and if they are related to the topic or the weeks materials, then and they're all different and you can review them. If I remember correctly as well, I've never tried that. Actually, you can look at the incorrect question, can't you? And see what you know from your results. Oh no.

<u>You can if you submit the quiz. Currently. One of the issues of the system is looking at previous attempts so</u>

That's what I was thinking. It would be nice if you could roll back and see like what what I done. So like it says you're you know on this time at this. So at this date and at this time you scored nine right? Which is great. But what was that one that I got wrong? You know we've got students who are like that right there like, so which one did I get wrong and why? You know, so it was so yes at the time, but sometimes we're busy. You know, we want to take the quiz now. We wanna, you know, going well, we've gotta have dinner, come back and then reflect on it. But at that point maybe you've been logged out, so you know. So being able to look at previous attempts so that you can review on that is definitely useful. I would say.

<u>That was a feature that need that was missed on the initial design was picked up after the first user test. Going back wasn't. Embedded properly, which is something that a large part of Ipsative assessment is quite ironic. What do you think the drawbacks of such a system would be?</u>

As I said, so I mean like so if these quizzes so OK, so there is one argument so if we did put this as part of the summative OK, and you know we took their average score OK, so and that's what we would take as their their mark. I can imagine that a student would say that, so because they're all doing different quizzes as well and each question has a different level of difficulty. Then a student might not think it's fair if you know what I mean. So if we were to take that, as you know, so this is this is the score that you got, you know, so that so it's it's. It's kind of blending the Ipsative with summative. It doesn't work, well that's what I'm saying there. So that would be one of the drawbacks. But in terms of like a formative assessment, I don't see anything wrong with this kind of system. So, but to blend Ipsative and summative? Is a difficult thing to do as I said, because it would just I mean this this person is taken. You know the control control, try and speak, control structures Quiz 10 times. Their average score is four. Yeah, so I mean, we could for example say right? OK, we'll take the top five in the average of those. Maybe that would be a fair away, but even then, you know I do feel like a student would say, well, they didn't. They did different questions from me, you know. And if there's 100 of them. Then that's fine, but if there's like 20 of them and ten of them were easy and it just so happened that every you know one person got the 10 easy questions, then that's probably unfair to someone who did the 10 hard questions. Yeah, something like that.

<u>So the final question is what features do you think should be added to the system?</u>

I think I've mentioned quite a few, so we've gotten along so, but I, I mean the presentation needs work, so but that's not what you're asking in terms of features. So yeah, the previous attempts one would be useful. So, one thing we obviously identified was as the home page needed to be removed. Would have to say with me it would be easier if you could make it more obvious. Sometimes with your buttons. So, for example the content page I thought that was your content. I didn't realize I had to click on Java in order to then get the content. Which now makes sense, because obviously you wouldn't do it just for Java. You would do it for other concepts and topics such as Python, math, etc. Didn't know, didn't realize that though at first and quizzes, yeah, so I've already mentioned that in terms of websites. If you could also not just the radio button. So if you did I can't remember what the tag is now. There's a tab that you can use where you can give a label to the radio button and that would mean you could click on the text as well.

<u>Is the tag label?</u>

Yeah, it's label yes, it's Label, so the yeah. For some reason it's not showing any correct answers, which I find odd. So I mean, if you scored nine. Is so is the correct answers of the last Test that you took. Yes. So if the last Test I got was zero then it will not show any correct answers, right? OK, which is slightly odd. I wonder whether it might be worth kind of showing. Once again, you know these, you know these are the correct ones for, you know for this attempt and so on and so forth.

<u>So that might be something it should do. It should display what questions you got incorrect then what. It is a list of the questions you got incorrect and what their answer should be at the bottom. That's the way the system should work.</u>

But that's when you submit the quiz, isn't it? So I'm on the, the results tab, and when I click on correct answers, I see nothing. So I'm assuming, as I said, that's because the last test I took I got zero, so therefore I had no correct answers, which is unfortunately. So as I stated, that would, would not make me happy, but no matter. That's fine. We've all gotta learn, but yeah, so that kind of thing. Obviously some of the quizzes you had some quizzes for some content, and some for others, but that's just content. That's not really that important. The goals. I mean, the goals, as I said, actually, what could have helped would be some content about how to set a good goal. I do think that would be very useful to the students you know. So the idea of making a realistic goal. 'cause I said so by setting an unrealistic goal they would unintentionally demotivate themselves. You know, so you know the current goals are not too bad. Actually, take the introduction programming quiz. That's great that that's that's a simple one that can and is easily, you know, done revise, control structure specific, specifically, the do loop questions. Yeah, OK, that's fine. There's nothing. I mean some of the completed goals are ridiculous. Generics and testing are a bit not specific enough, but yeah, I I get the idea. The small kind of thing would be when you add a new goal and you say submit. It could have automatically taken you to the current goals tab an about. Yeah, there's nothing there and then log out. So I mean like yeah, that's fine. So so it's it's. I mean, I like, I've, I've always liked the idea of Ipsative assessment, in formative kind of exercises. I always have done and I feel like students do learn a lot more from that process. But the problem I still think. That you have with these systems is it just doesn't belt well with summative, and as I said this, this would great be great for revision purposes for as an exam, ironically enough, but because we've moved away from exams, it's actually probably less useful in that case, but it would be great for, so we you know, As I said, we use canvas quizzes. Now it's the same ten questions. By having these separate ones. Actually that would be useful. And having this goal system as well embedded would be brilliant. Yeah.

<u>Do you have any further comments you wish to add that haven't been covered by the questions?</u>

No, I think I've covered everything now, sorry.

***Staff 2***

**Based on Ipsative assessment as a concept would be beneficial to students?**

Probably, but quite, with the student numbers we have could be quite practically difficult to do, but I I would imagine that some students would benefit from it, so certainly.

<u>And how do you think students would benefit</u>

Those who want to sort of progress by doing etc on they will benefit because if they, if the feedback relates to what they did last time, they will, they will presumably demonstrate continuous improvement in some way, shape, or form, or know if they have exhibited some kind of continuous improvement. I mean, some students just don't, by and large don't seem to care, so probably wouldn't bother looking back at it, but I think for those students and that most of them who do care, I think it would be a I think that's how they would benefit, yes.

<u>How about from the staff perspective? Do you think it would be beneficial to staff in any way?</u>

I said, I think it's one of those things like small group teaching. You would ideally love to do it. If you, if you could. You know, uhm. Whether somebody would actually have the time to go back through 300 students previous assessments or not is is another matter. I think it would give some indication as to who was making an effort, because you could tell if, if there were sort of little things, but important things like variable names that you mentioned or comments, those kind of things. Then I think it would be, certainly good indicator who's taking the course seriously, And the like. I think it would probably work better with more with some modules another. So, for a lot of the modules that I teach, the second assessment looks at something completely different to the first. So by which I'm meaning that like the information storage and retrieval semester, two exam covers semester two material, semester one coursework covers semester one material. But I think it would work better for things like programming where there's lots of sort of practice things that you evolve as you go along. So I don't think it would work necessarily, work well for every module, but I think there's some topics it would work really well for, particularly ones where you've got sort of weekly practical exercises that build on each other as well.

<u>So developing off that answer, what about the, so you mentioned that for example, semester two and semester one are completely different from each other? Are there any transferable skills that this could help with? So essay writing, research, structuring, the way the answer questions?</u>

Possibly, but I think the odds of, so I think being able to go back over and find those things for certainly for the big class would be quite daunting for a staff member particularly if they've got about, and also for potentially for PhD student marker as well, who may also be approaching this. Particularly if you've got, you know if you've got three and a bit weeks to Mark 300 scripts, it's like you've got X minutes and to do it and you just may not have the have the time and I think it could be quite daunting for, particularly for a new marker as well, but for a small group I think it would work very well.

<u>So regarding CIA itself, which is the system. Do you think this system like this would help a student gain an understanding of the concepts of programming?</u>

I would say so yes. Yes, I think it's. It's easy to navigate and it can obviously be extended. upon by admin to add in more things as the course gets more complicated even be done as a sort of non modular thing. So for instance, programming two could add things into it and leave in the concepts and programming one. For example, if you've got common key topics that run for a long duration course.

Do you think CA provides a clear region for self reflection, self regulation and student growth?

I'm not sure about it doesn't seem, the reflection I'm not quite sure, other than that you can look back at your results, student growth. Definitely can see what you already did and you can do the same things like improving. It depends If the student is reflective nature, they should be able to use that to self reflect.

When you were exploring the quiz. Did you find the questions to be suitable for novice students? Where they? Did you find any to be too hard to be impossible or manageable but challenging?

I would say manageable but challenging. And because the students can have as many girls as they like,  it doesn't matter if they find it, If they find it challenging at first if, if they've been studying for like 12 weeks and then they found it challenging, I might be a bit more alarmed. I think it. I think it'll push the students, but I don't think that's a bad thing.

What do you think of the benefits of such a system would be to the student to the class, to the teacher of, to the students in the class?

Well for the students in the class, for starters, it's not actually assessed, so they can work together on problems without getting an assessment irregularity. As long they don't take that approach on into their coursework. I think for the student. It's not something that they have to give any obvious answer, so if if they just sort of completely bombed the first couple of goals. Who's gonna know, kind of thing? So it's got that sort of person, It's got that kind of personal thing to it. For the teacher to stop the blathering on can you think of any more example questions etc etc because they were already out there. Obviously I know the teachers got to think of the example questions, but they had something done. So I mean for for the teacher they can just give some other ways of testing so. Hopefully they'll be better prepared for doing exam questions 'cause hopefully they'll be better at critical thinking.

OK, and on the flip side, what do you think the drawbacks of such a system would be?

If somebody was repeatedly getting low scores. It might be a bit demoralizing. Uhm, any sort of error in it like? I mean, I obviously clicked the wrong button, but I thought I'd answered option 3 to one which it said I've got long. Any kind of bugs in it would, could cause issues with that. No real different to any other sort of similar system, I would say. I would say, as I said, probably could be a bit disheartening for they student who kept on and on doing like very very, very fast but maybe an anonymous call for help button I don't mean something like I'm standing on the edge of the cliff, kind of that mental health kind of thing. But like, like, an option where it could send in a question, and to somewhere that the lecture could then reply to might be a might be a handy thing that how you would sort out the anonymity. I don't know or what people how people could misuse that. Having seen lectures being on the receiving end of student racism before, I might actually think a bit carefully about that, because that was all done. Hidden behind an anonymous curtain.

<u>So the final question is you mentioned something there, but in case there's anything else you wanted to add, what do you, What features do you think could be added to the system to improve it?</u>

So possibly. Some way of calling, you're getting help that could maybe speed things up, so maybe auto generate a message somewhere where the subject is stuck on question, on Quiz X question Y kind of thing. I wouldn't say too much. I'm not one of these people who would want leader boards and compare with your friends and things because I think if you can't do it knowing that everybody else in the class can and you can't, it's  probably not going to be very motivating. So I would say that is definitely not an additional feature. Uhm? So, from a student perspective, I see I could from a staff person only call for help from a student perspective, staff perspective apart from the sort of admin functions you'd expect, like adding and adding quizzes and questions and things, possibly something, I dunno part of me thought it would be quite nice to sort of have a tracker to see how people are getting on with the material, but that might discourage students from doing it if they think they're gonna show up as being stupid. You see, I mean sometimes people don't think we want to know who's struggling, so we can help them. They think, you know it's not because so we can laugh at them and say they’re thick its so we can help them, but not all students seem to get that. I mean, I had a student once who failed everything I said that said look your highest mark at 32. You're programming, you know, they were running these extra programming clinics. Did you know about them? Yes. Did you go to any of them? No. Well, would you have benefited from going to them? Yes. Well, why didn't you? I didn't want everyone to think I was stupid. And now, thinking well, you know. It's not very sensible not going ,so because of that misinterpretation. Maybe what will be better would be like a sort of aggregated anonymous results where the teacher can have a look at it, and if it's. I mean, if the teacher looked at it and saw everybody was getting one out of ten for control structures, it could be like let's do some extra material on some control on control structures. Or here's a drop in session on control structures or a an extra in person session in the lecture room or something. So maybe for staff sort of possibly anonymous. I don't know ways of seeing roughly who's engaging and what's and what's happening with it. I mean if only then you could, if only for the fact you would see if people aren't using it.

***Staff 3***

Can you describe to me your what your understanding of Ipsative assessment is?

So I think that it is sort of. Is it learning from that learning from, using it as an assessment tool from their previous learning so it's not against like other students. I suppose it's from like what they've done before like their previous learning.

Yeah, so the core idea of assessment is that the student can look back on their previous work, see what they didn't do too well and tell their learning from that point they can move forward by saying he didn't do well in control structures, so we need to focus on that and assess your understanding of control structures next. It's about that personalization and self reflection and regulation of the work.

So as a concept, what do you think of Ipsative assessment, would be beneficial to the students?

Yeah, definitely 'cause I think it's a it's it's so it's. It's a reflection on their personal knowledge instead of uh I suppose benchmark against other students. So they can kind of have an idea of where? The idea that specific areas that they need to improve And it's it's based. It's it's a bit more personalized kind of method of assessment, I think.

Do you think there are any drawbacks to the students for using this system?

Uhm? No, not many drawbacks, I mean sometimes it is good good to have a look at what

what level like your peers are at, ‘cause sometime that can motivate I suppose. If you think that other people are doing kind of better than yeah but That's not necessarily, yeah, a drawback.

Do you think a system such as computing Ipsative assessment would help students gain an understanding of the concepts of programming?

Yeah I think so. I mean you could use it sort of, especially your tool so you could use it sort of like as a diagnostic thing to find out areas that they need a bit more attention on so they might understand some concepts, but they might not understand others. So you could. Yeah, get them to do the quiz and then the know specific areas where they need to improve other than the content there to help them build that knowledge.

Do you think computing Ipsative Assessment provides a clear region for self reflection, self regulation and growth?

Yeah, I think so, and I think I really like that, they can set their own goals. I think that's good and they the current goals that they have so they know kind of where they're at. And then they can see the kind of the the results so we can see if that made an improvement so they can see if kind of the question that they gotten, kind of more points so that they have progressed or at they could maybe reflect on why they haven't met the current goal.

So when exploring the quiz sections of the system, do you find them to be suitable for first year students? Are they too hard or the impossible? Manageable but challenging?

I think about the right level. I would see it manageable but challenging 'cause if they're not, they're not really simple. You know what I mean? They're not like, They would be required, Then they have to have a think about they require to have some understanding, so I think that the light level I think the they're not too hard, but they're not too easy at the same time. So kind of a starter.

So we've talked about the benefits of an ipsative system of Ipsative assessment as a whole, what do you think the benefits of such a system of this system in particular would be to staff and students?

So you could use it as, so as staff, you could use it as I could a diagnostic tool I suppose, so you could say try, try the quizzes to see where you're at. Try the quizzes so if you. See if you under understand, see how many you get like to see evidence that you're not that sure on so you could use it like that. And for student, yeah, it lets them reflect on kind of their understanding and then it helps them in a structured way to improve their knowledge 'cause they can create their own goals so they could, they could see areas they need to improve on.

And similarly, what do you think the drawbacks of such a system would be?

Uhm? I suppose you could get, I don't know actually. Maybe they could get a bit more bit caught up and kind of the the marks for it and kind of think, more about getting the answers right, than the actual understanding behind it. So like just kind of, be more, they could, they might just be a bit more focused on kind of the quizzes then I suppose, really digesting the content, but I don't think that that's like a big a big drawback.

<u>And are there any features you think should be added to the system?</u>

No, I don't think so. I think it's got everything 'cause it's got the the content. And then there's, yeah there is quizzes form them I really like the section for the results I think that's really good and the goals. So no, I think it's got. It's got everything.

<u>That was the last structured question I had, but are there any other comments or input you would like to make on the system?</u>

Erm no, I just think it's nicely set out. It's easy to navigate your way around, its easy to use. And I think the contents nicely set out. Yeah, it's just. It's it's nice and easy to use, easy to navigate. And the content the contents good in it, the good level.

***Staff 4***

<u>So the first thing I have to ask is, are you aware of what Ipsative assessment is?</u>

I've heard of it. Haven't really practice it too much? I've seen principle, but I have read

around the subject area.

<u>Can you describe to me what you think the theory of Ipsative assessment is?</u>

I think its main theory is that it's rather than sort of seeing grades or assessment has been done in isolation. It's about building up a sort of portfolio of tests and and that sort of thing and work and gradually improving it based on on on the previous grades and experiences. So basically started off with say perhaps low marks. Then building up on that, so hopefully next time round people can have higher marks and also reflect on what they did well and not in the previous tests for the current tests or for their current assessment.

<u>You covered most of the fact you covered it quite well. The Ipsative theory and the important key points are sort of the reflection based on past marks and planning towards the future and more</u>

focusing on the learning gain from the student perspective as opposed to the end point. So saying that as a concept, do you think ipsative assessment would be beneficial to students?

I think it would be out. I would see it as part of the overall assessment certainly. I can see this as being part of the overall process I think we still need to have tests and coursework, and that's certainly see this being a sort of ongoing process in. The background marks as well. They would be able to see the actual value of doing these sorts of things. I suppose as well, looking at this style of system and I would imagine that it's probably less. It'll be probably, less daunting to use then having to do formal examining as well.

What would you say are the drawbacks of Ipsative Assessment for students ?

Students may not see the big picture of their overall assessment experience/learning objectives for a module, and for some I can imagine having multiple assessment may be frustrating, especially if it "forces" them to reflect on things they didn't do so well in the past, some students prefer to "move on".

OK, so you kind of touched on this there, but do you think the system such as CIA would help students understanding of the concepts of programming in this form of learning?

So I think certainly in terms of assessment and and assessing their understanding of programming, I would say probably would help with that, certainly. I would say it wouldn't really be a substitute for things like a lectures and that sort of think but certainly be tutorial style things could help with their learning in practicals.

OK, do you think that CIA provides a clear region for self reflection regulation and growth?

So looking, so one thing I thought when I was getting their feedback on quizzes. It might have been easier having some sort of colour scheme. Accordingly, to make it clear what they got right. I like the idea that that doesn't really have too many statistics. People get hooked on that sort of stuff, so certainly think think perhaps having the feedback a bit clearer to things that they got right. Highlighting those and then perhaps being slightly more clearer with things they didn't get would be quite useful.

How did you find the goal section? Did you think that was helpful in this sort of regulation?

Yeah, so for contents section. That style one yes, I thought it was quite useful and I liked the use of, so I like the screenshots of the code samples. The suggestion I would have for if you will move forward with this system for more final into implementation is to perhaps have those blocks as actual code tag sections. So you can actually do the syntax highlighting with actual selectable text, because that might, that might encourage people to try out code snippets and that sort of thing as well.

<u>So what I was referring to, there is specifically the goal section where students can type in their own goals</u>

OK, sorry, I do apologize. I thought that was a nice idea. Sorry yeah, so have the current Goals is nice. What is wondering though, is it might be nice to have a similar thing you saw, current state. How many are outstanding? How many unfinished and of course finished with.

<u>So was it easy for you to find the goals that you that had been previously completed with the system.</u>

Yes it was absolutely fine, yeah? I can see it here.

<u>So one thing I didn't ask before, which was meant to is we talked about Ipsative assessment from the student side and the workload and the benefits. What about from the staff side? Do you see any benefits as a staff member to using something like Ipsative assessment?</u>

Well, I think probably yes, because that there is quite a lot of information in the system which makes it easier to not having to mark also, Its always good to try and reduce ones marking load. Also as well it can be handy because if you have a staff view you can make sure everyone is progressing well. Also you can see if students are doing it, that kind of stuff. Marks are dropping down and that might, They might be helpful.

<u>What about the limitations or drawbacks? From the staff perspective, do you have any comments on those?</u>

So I think really it's, well, I think there's two aspects, I suppose, and that's the making course content can actually might be because this is online probably wouldn't be the same as actually seeing them face to face in a Seminar. That's sort of stuff. Unless this was done and the practical of course. Also there wouldn't be much in the way of extra help and support and helping them.

<u>That covered the staff perceptions of the system and how the benefits and drawbacks.</u>

<u>What about Ipsative assessment as a concept? If it wasn't in the system and it was something you had to implement as a staff member?</u>

So I think I can certainly see it in useful in terms of especially students start off with, the first set of modules, especially when they need quite long, And lots of support, support when they're starting from the blank slate. Having a way for them to easily sort of done that they're working knowledge from scratch and also have a way to reference would be useful. I think from my perspective though for say for more advanced modules in their third year as a problem in the masters year with that sort of stuff. I think you probably have to be a bit more careful with it with this style. The pace of it might be too small for them specially with the timeframe more limited might be more appropriate for that.

<u>So you said you had to play around with the quizzes and you did some questions on that, when you were exploring the questions. Did you find them to be suitable for students, or were they too hard for the impossible to do or manageable but challenging?</u>

I would say that most of them seemed fine it may have been a bit brief, so perhaps having a bit more context would be useful. Or perhaps even a sidebar or pop up, get out more guidance may be a nice to add as well.

<u>OK, uhm. So yeah, this is kind of already being covered, but just to be a bit more direct about it, what do you think the benefits of using this system in a class would be?</u>

So I think it certainly in sort of practical setting or questions. It's probably nice structured way for them to actually learn something to top up their skills or their knowledge. Certainly someone with you set aside the practical would be to do the stuff and have them to do it in their own Time as well. Obviously it wouldn't be a substitute for attending lectures, So I think I could support tool I would say.

<u>I, that sort of answers the next question of what do you think the drawbacks of using such a system in a class would be?</u>

OK, right? So as I said, I think it's fine for verifying people’s working knowledge and that sort of thing. I would be concerned that students might think it's a substitute for actually attending the lectures, get away with doing this, I'm not sure actually. Its is often the way I suppose, especially with online and tutorials, and that sometimes prefer those and kind of skip often, the lectures themselves.

<u>OK, and the final question I have for you is what features do you think should be added to</u>

<u>the system in future iterations?</u>

So. I think I said for the tests perhaps having some extra clarification for if they get stuck, perhaps a side link to  Some extra help might be an idea to have some videos in as well as part of the stream as part of the voice. Even as well with programming, A bit like what you can do with them to be free schools having sort online, probably out of scope and more advance but have a way to actually execute the code so people can actually try say write the method and then compiling it and then see it actually working. At the same time as they doing the tutorial as well. I suppose suppose part of that it could actually be part of the assessment as well. So if they wrote something which follows the spec, question compiles and works like and then not be formed a part of their overall grade as well for the test.

<u>OK, and are there any other comments you would like to make that hasn't haven't been covered by questions that I asked today?</u>

It's a it's an interesting system. I think it's something which would be of use to students and staff and like.

***Staff 5***

<u>Do you think Ipsative assessment as a concept would be beneficial to students?</u>

Yes to students, I think it would be beneficial.

<u>In what way?</u>

I think as a student it would be good if my areas of improvement were highlighted and that future assessment would assess me on things that I am struggling with. Rather than re assessing things that I have in the past done quite well on.

<u>I noticed that you put a heavy emphasis on that it would be beneficial for the students, how would you find it from the staff perspective?</u>

I find that, my understanding of it while primitive and naïve is that if you have larger working groups or cohorts of classes that need to be assessed its more difficult to provide an Ipsative experience, for a larger group. Whereas I imagine its not as difficult with maybe a class of twenty or thirty.

<u>So in an ideal world, where you have an ideal world where you have a class of twenty or thirty, are their benefits to the ipsative process to the staff member?</u>

I think so because as a staff member when you have a class of about 20 or 30 you have more of a connection to students and their abilities, and a cheesy as it sounds their journey through the education. Therefore, I will be able to tailor assessment more unique to things they are struggling with. I can imagine that providing diminishing returns when you get to larger groups of students.

<u>Considering the system C:IA which you had access too before this meeting and you had a play around with for a system that tried to automate that and encourages the students to set up their own goals and direct their learning. Would that have a benefit in a classroom or a classroom with the 300 people?</u>

I think if you could get it working for both it would be a fantastic tool to use, erm, my concern would be making sure that every student is getting an equal assessment opportunity. I think irrelevant to the class of 30 or 300 if the system made assessment fair and it was purely Ipsative it would be great regardless of the numbers, but I could see it being easy to implement on a small class.

<u>In the sense of a formative tool if not a summative tool does it have benefits in that sense?</u>

Absolutely I think any tool that supports formative learning of programming languages especially java is useful. Erm as long as the content is up to date, relevant and correct.

<u>Would C:IA help develop the concept of programming?</u>

With enough supporting material to go alongside CIA, such as extra reading, and extra examples and case studies of using a programming concept then CIA could be useful for learning programming concepts.

<u>Did you find while you were looking through C:IA, did it provide clear regions for self reflection, regulation and growth. Did it encourage the student to consider this?</u>

My part of C:IA, I only looked at one particular question set so I didn't get a heavy sense of reflection from it. At face value I took it at a quiz format, probably because I only looked at the quizzes. If you want me to have a look at it in more detail then feel free.

<u>Did you have a look at the goals section?</u>

No I only looked at the quiz section.

<u>Could you have a look ?</u>

Yep do you want me to look at it now?

Yes Please

For the benefit of the recording I only looked at the quiz section, I am now looking at goals.

And results if you could.

I've got no goals highlighted, if I was to click at results. The results section has my last quizzes results on as well as the option to select the correct answers at the bottom. But it did highlight the questions in particular that I was struggling with.

So one thing that may not be clear and maybe a design issue is if you go back to the goals page and if you click on the tab that says add a goal, complete a goal, that should show you if a student can add goals and see what they're doing. And most importantly see their previous goals too.

Right, yep. I can see that, would you like me to add a goal.

Yes if you would like to add a goal.

Erm, revise control structure.

I think you said you didn't get the do loop quite correct**.**

That right. Typically the do loop questions. That's submitted and in my current goals that has been highlighted there. Revise control loop structures, specifically the do loop questions.

So based on having viewed that if we go back to the question, does C:IA provide a good region for self reflection, regulation and growth ? What do you think of that?

I think, self regulation is kind of like taking control of your own goals in a way, I think you have the ability to. My worry would be that students need to be very, erm, not militant, what's the word I am looking for? Organised to acknowledge their goals as well as jot down what the current ones are and also mark them off as complete as they go. Erm, as for reflective yeah sure having a nice list of all the goals that completed which the system does have, would be nice to see. It would be nice to see it maybe categorised a bit more. Or if it was in a, or if I was able to format them by time or completed goals, or I don't know, if I was able to add labels or tabs to them perhaps. Say look I completed this particular thing related to Java or this particular concept related to control loops etc. Erm so yea, if I was going to give this to a student, I would be worried that they wouldn't want to keep track of their goals unless they are reminded too after a certain time frame. So perhaps, I don't know if this is relevant but if there is the ability to ping the students to a notification or an email to say you haven’t added a goal in X amount of time, would you like to do so? Here are suggested topics that you have struggled on in the past.

Going back to the quiz, when exploring the quiz sections, do you find the questions to be suitable for first year students? How do you find them in the sense of, are they too easy, are they too hard? Are They unreachable for the students to do, or it is manageable but a challenge for the students to do?

I think its manageable. I mean I attempted the quiz before and ended up doing no revision and was able to get a moderate score as somebody who actually develops for Java I was quite ashamed to not get 100% but erm, you know, I did it on a whim quite quickly. I think its, it depends if you gave it to the students at the beginning of the year, at the beginning of their experience with Java they would struggle obviously as they haven’t seen Java before. But they would have seen another programming language before, so they might be able to get some of the questions right, but they might not be able to understand some of the syntax or the semantics of the language. If you gave it to them at the end of their learning of Java, at the introductory to Java in their first year of stage one, then they should be able to get eh the majority of the questions correct, urm, yeah.

What do you think the benefits of using a system like this would be?

Practice. Always practice. When you’re programming, especially novice programmers, opportunity to practice is not something that comes straight away to students, some students don’t recognise programming as a skill set that you need to get good at over time and that you can only get better, you can’t get better by reading about java. You need to practice it, you need to see the error messages and such. And having a system that highlights particular core areas, like control structures, variables, string handling etc, erm would allow me to practice all of those questions, erm, my only concern is that, would be, there is a limited set of questions here, obviously Java goes much wider breadth than what is available on the system at the moment. It would need more questions to make it worth while. Even categorising them in terms of difficulty may be good erm I didn’t test this before but are the questions static or do the come from a dynamic range of questions.

There is a database that stores the questions and each question has a attribute which says whether its what sections its in and it just pulls a random 10 questions in a random order. However since it is a prototype there is a limited amount of questions, so in this case students would be able to do the quizzes constantly and see the correct answers which is why I encourage them in the quiz itself to not actually look at the correct answers the first time around.

Makes sense, yeah.

So opposite to that, what do you think the drawbacks would be?

Its another thing to, its another system, let me be more clear, its another system that you have to keep on top of. It's a stressful time learning to program, especially the frustration of not getting it straight away, of not getting what's going on, or being confused by error messages or noticing that your friends are understanding it and getting it quicker than you that can apply a lot of stress to a student, both academically and also maybe, like self motivation or thinking that they're not good enough to be able to do it. I think that having another system where its metric based and they're saying, oh I've only got maybe two out of ten on these questions correctly. The would be that the students would see themselves, would see that they're not actually improving at all and it may be a negative experience for them. Which to a learner a new learning in a programming language, that can be quite a road block I would say. But if it was highlighted as a system that was intended to help you track your progress over time and provide questions that are challenging but obviously re enforce that its ok to get things wrong that's the whole point of learning then, I can't see the system being bad or a drawback. Erm to long didn't read, basically its another system. Erm that students may struggle to keep on top of while also learning how to program.

The final question I have is what features do you think should be added to the system going forward?

If you bare with for a couple minuets for me to have a proper look around the system for me to see what else there is. So you have a section of the content for the module specific stuff, you've got the quizzes to pull in based on a score. Results. Ability to track goals. Something about data protection. Erm ok. Mabey perhaps per student. Not user, because you should have an academic or module leader profile or something that is able to set questions. But from a students perspective if they had like a profile page that allowed them to see results of quizzes over a time period.

There is a homepage, if you click on C:IA it should take you to a homepage which displays their latest result.

So similar concept but over a time period perhaps, so you are able to see how you have progressed over a certain time. And that's, if one thing , if reflection is a heavy part of your module being able to look back and see how you have progressed quite well, as well as the feedback because lets say point a point b and point c, if you are able to access that information at the end you would be able to reflect on it quite well and as a learner I think reflection is very important because you are able to see how you have actually progressed and it gives you the motivation and confidence to keep on going. Erm

Ok, so this may not be clear either, but if you go onto the results page, and if you click on say control structures, that should bring up a list of everything you have done and all the different quiz topics, the only thing it doesn't do is save the incorrect questions from each one, which was a feature identified by students.

Yeah, I think pulling that information into one that it being just on results it being attached to your profile page. More visual less text, having a time graph perhaps, allowing you to select specific points then it brings up the score received at the correct time. That's perhaps the second or third time that you've said maybe that isn't clear from design. So saying that I would re think the design and where these things are being pulled from. Personally I believe the less clicks to see this information the better. That's a personal belief I'm not a design expert, erm but it makes sense to me to have it in a profile or past answers.

**Ok, thank you. That was all the questions I had.**

CSC1021 - Practical 8 (Methods)

# <Lecturer Name>

## Aims

- To introduce methods
- To gain familiarity with parameters and return values

### Introduction

In this practical, we are going to start by calculating the areas and volumes of a square and a cube, and finish by writing a simple power or exponentiation method.

This practical is quite mathematical, but its purpose is not to test your maths. If you are unfamiliar with this piece of maths, feel free to look up an explanation on the web, or ask one of the demonstrators.

# Task 1

Create a new package called "Practical8" inside your Eclipse workspace.

```
/**
 *
 * Purpose: Calculating Powers
 * @author <name>
 *
 */
public class Power{
  /*Complete*/

  public static void main(String[] args){
    Power p = new Power();

    int length = 6;
```

```
      p.square( length );
      p.cube( length );
      p.hypercube( length );
   }

}
```

Taken from: Power/Power.java

As it stands, this code will not run.

# Task 2

Write methods that calculate the square, cube and hypercube as shown in the comments. For example, you could define `square` as follows:

```
public void square( int length ){
     System.out.println( "The square of " + length + " is:"
                    + (length*length) );
}
```

When you have finished, the Java should run and execute correctly, and display three print statements. If you are not sure that you have this answer fully correct, please ask a demonstrator at this point.

| | |
|---|---|
| **Note** | By hypercube, I mean the fourth power. It is not really the correct name, but either it or possibly "tesseract" is about the closest that there is. |
| **Warning** | Make a copy of this code as you will be changing it for the next task. This is the last warning — from now on, you will need to remember to do this for yourself. |

# Task 3

The program that you have now works, but it is not an example of good design. We will see later why this approach is very limiting. Instead of placing `println` statements inside the methods, let’s move them to the main method.

Consider the following code:

```
/**
```

```
 *
 * Purpose: Calculating Powers
 * @author <nane>
 *
 */
public class Power{
  /*Complete*/

  public static void main(String[] args){
    Power p = new Power();

    int length = 6;

    System.out.println( "The square of " + length +
                   " is: " + p.square( length ) );

    System.out.println( "The cube of " + length +
                   " is: " + p.cube( length ) );

    System.out.println( "The hypercube of " + length +
                   " is: " + p.hypercube( length ) );

  }
}
```

Taken from: Power2/Power.java

Rewrite your methods so these execute correctly. You will need to use a `return` statement and change a few other things to match.

# Task 4

This is now a bit long-winded. You probably have expressions like this:

```
length*length
length*length*length
```

Firstly, `length` is a bit long for a variable name, particular as `square` and `cube` methods can meaningful operate over things other than lengths. So, first, change `length` to `x`. Eclipse will do this all for you, if you ask it nicely.

Secondly, you probably have the same statements at several places in your code. Let's replace these with calls to the appropriate method. So, in your cube method, make the following change.

```
// replace
x*x*x

// with
x * square(x)
```

The hypercube methods should also be changed to include a call to the cube method.

# Task 5

This is still quite long winded. It would be nice to change the square, cube, and hypercube methods into a single method. We are going to achieve this in several steps. First, introduce a method called power as follows:

```
// Define a power method, with two parameters x and n

// if n == 2 call the square method with x as parameter

// if n == 3 call the cube method

// if n == 4 call the hypercube method
```

Add to the power method an appropriate condition for n==1. If you are not sure what to do in this circumstance, please ask a demonstrator. The complete method should implement these comments:

```
// Define a power method, with two parameters x and n

// if n == 1 the appropriate condition

// if n == 2 call the square method with x as parameter

// if n == 3 call the cube method

// if n == 4 call the hypercube method
```

# Task 6

We now need to use the power method. Edit the calls in the main section of the code to use power rather than square, cube and hypercube. To do this, you will need to add an appropriate **exponent** as the second parameter to each call.

## Task 7

Currently the cube method calls the square method, and the hypercube method calls the cube method. We have just replaced these calls in the main method with calls to the power method. Replace the calls in cube and hypercube with calls to the power method also.

## Task 8

We still have a number of calls to square, cube and hypercube within the power method. But these methods are now not doing much. So, take the method body of, for example square and put in place of the call to square in the power method. Do likewise for cube and hypercube. You should now have no calls to square, cube or hypercube so can safely remove these.

**Note** This is now an example of a **recursive** call. These look very strange when you are not used to them. The trick is to not worry about it too much. You are just carrying out the same substitution that you did earlier on the main part of the program.

## Task 9

Your code should now have only a single method called power, with four if statements. A close look at the if statements for n>1 show they are identical in form; try and replace these statements with a single line of code.

You should now have a method that works for any integer value of n. Add another println statement calculating the fifth power to demonstrate.

## Optional Task 10

**Note** This task is optional, and you do not need to complete it to get sign-off.

The code as it stands is missing quite a few edge cases. Can you modify is so that it works for any integer value?

## Conclusion

This concludes the second practical. Make sure you have your worked signed off.

Last updated 2016-11-03 16:21:10 GMT

# Appendix D – Feedback Sheet

**Feedback Sheet, CSC1021 Project 1**

The highlighted answers below give you *general* feedback on issues concerning your project.

**Student Name: . . . . . . . . . . . . . .  . . . . . . . . . . . . . . . . . . . . . . . .**

**The Program**

| | | | |
|---|---|---|---|
| Is the tax calculation correct? | Yes | Partially | No |
| Is the Tax calculation well coded? | Yes | Partially | No |
| Is the structure of the tax calculation clear? | Yes | Partially | No |
| Are exception conditions catered for (e.g. tax on £2,000.50)? | Yes | Partially | No |
| Is the Bar Chart drawn correctly? | Yes | Partially | No |
| Is the tax table well laid out? | Yes | Partially | No |
| Is the overall solution well structured? | Yes | Partially | No |
| Is the overall solution well coded? | Yes | Partially | No |
| Is the code well commented? | Yes | Partially | No |
| Is there appropriate use of constants? | Yes | Partially | No |
| Is the overall solution correct? | Yes | Partially | No |
| Have the style rules been followed? | Yes | Partially | No |
| Is the testing adequate? | Yes | Partially | No |
| Is the required output present? | Yes | Partially | No |

**Additional Comments**

**Marks**

| | |
|---|---|
| **Program (out of 16)** | |
| **Documentation (out of 4)** | |
| **Total (out of 20)** | |

**Marked by:** . . . . . . . . . . . . . .  . . . . . . . . . . . . . . . . . . . . . . . . . . **Date:**

**Appendix E Coursework Specification**

## CSC1021: Assignment 2

# <Name>

## Aims

This assignment will exercise:

- The use of the basic language features of Java
- Designing and building a solution to a specification

## Background

As part of a larger project, building an online ordering system for a Pizza company, you need to design and implement a system for calculating the costs and some other features about a pizza. The system will allow you to build a pizza from a base and some toppings.

## Task

Your task is to create a prototype which allows cost calculations for pizzas.

You will need at least three classes.

`Pizza`

> This should carry all the essential information about a pizza, including a single base and some toppings.

`PizzaBase`

> This class should model a pizza base.

`PizzaTopping`

> This class should model a pizza topping.

You will need to instantiate multiple objects of each of these classes. You should have at least two `PizzaBase` objects (one thick, one thin), and at least four `PizzaTopping` objects. If you are not an expert on pizza, you can find toppings by searching for a pizza menu on the web. You will need to store this in an appropriate data structure for use within your program.

Pay attention to the design principles that you have been taught during the module. For instance, I would expect that each ingredient will have a `Cost` property, which will be accessible by `get/set` methods, and be stored in a `private` member variable.

For later stages of this coursework, you will build a simple user interface. This is a prototype system, so the user interface does **NOT** need to be complex; a simple command-line interface using `Scanner` and `println` statements is entirely sufficient. However, as your classes may be used in several different situations, none of `Pizza`, `PizzaBase` nor `PizzaTopping` should require any user interaction; use of `println` or `Scanner` within these three classes will be considered an error, and will be penalised.

There are a number of features that you should attempt to achieve; please complete these in the order given.

## Build a Pizza

The initial version of your system should allow enable you to create objects for a pizza, one base and several toppings. Each of the ingredients should have a cost associated with it. You should be able to combine these to create a pizza which should have one and only one base, but many toppings. The finished pizza should calculate it's cost. You should write a simple class called `PizzaBuild` which contains a `main` method which demonstrates this working.

## Create a user interface

Create a new class called `PizzaChoice`. This should have a `main` method, and class should provide a simple user interface. This should provide you with a choose of one base, and a number of toppings. It should compose the objects that you created in the previous task. Once complete, you should print a short report showing all the ingredients, the price for each and the total cost. The report should be formatted for easy reading. You may create additional classes, beyond PizzaChoice should you find this useful.

## Named Pizza

Sometimes, there can be too many choices in life. Write a new class PizzaMenu, with a main method, which allows the user to choose from a menu of predefined pizza's. For instance, a Margherita pizza consists of tomato, mozzarella and basil, on a thin base. You should provide the user with a choice of at least three pizzas. As before, you should print a short report showing all the ingredients. You may create additional classes beyond PizzaMenu should you find this useful.

## Vegetarian Options

Modify your PizzaToping, PizzaBase and Pizza classes so that the contain a property which describes whether they are vegetarian or not. A pizza should be vegetarian if and only if all of its ingredients are. Extend the reports produced by the last tasks so that they clearly state whether the pizza is suitable for vegetarians.

## SUBMISSION NOTES:

You should compile a report, which contains an introduction of at least 1 side A4 (no more than 2) providing an overview of your program's design. This should include:

- A description of which tasks your code achieves.
- A description of how you have used inheritance within the program.
- Output from the program(s) for a defined set of input.

Finally you must include testing documentation. At least 5 test cases are expected and for each you must provide:

- Purpose of test (concise description)
- Coverage (which classes/methods are being exercised?)

- Test values
- Results (both expected and actual)
- Debug information (changes you made as a result of this test, if any).

Submission:

- You should submit an electronic version of report as a Word document.
- In addition you must submit an electronic version of your Java source code using NESS.
- Submission should be in an appropriate form. Code should **NOT** be copied into Word or Notepad, but should be left as source files.

## MARKING SCHEME:

Java Code 70%

Correctness 40%

Style and readability 30%

Report 30%

Introduction/design 10%

Testing 20%

The marks are split evenly amoung the four tasks. Code style should follow Java coding conventions.

Last updated 2013-11-19 15:09:12 GMT

***CSC1021: Assignment 2***

# <Name>

## Aims

The assignment will exercise:

- The use of the basic language features of Java

- Designing and building a solution to a specification

## Background

As part of a larger project, building an online hotel management system, you need to design and implement a system for storing and calculating some key information about a hotel.

## Task

Your task is to create a prototype which allows capturing knowledge about the current state of a hotel.

You will need at least three classes:

Hotel

This should store all the essential information about a hotel, including a name and some rooms.

Room

This should store the number of beds in a room.

Bed

This should store the size of a bed (i.e. single or double).

You will need to instantiate multiple objects of these classes. Pay attention to the design principles that you have been taught during the module. For instance, every `Hotel` should have a `Name` property, which will be accessible by `get/set` methods and be stored in a `private` member variable.

For later stages of this coursework, you will build a simple user interface. This is a prototype system, so the user interface does **NOT** need to be complex; a simple command-line interface using `Scanner` and `println` statements is entirely sufficient. However, as your classes may be used in several different situations, none of `Hotel`, `Room` nor `Bed` should require any user interaction; use of `println` or `Scanner` within these three classes will be considered an error, and will be penalised.

There are a number of features that you should attempt to achieve; please complete these in the order given.

## HotelReport

Create a class called HotelReport which, when given a Hotel object will produce a short, textual report describing the name of the hotel, the number of rooms and, for each room, lists the number and size of the beds.

## HotelTest

The initial version of your system should allow you to create objects describing one hotel, with several rooms, each with one or more beds. The completed Hotel object should be able to calculate its maximum occupancy. You should write a simple class called HotelTest which contains a main method which demonstrates this working.

## HotelConfigure

Create a simple class called HotelConfigure. This should have a main method which provides a simple user interface, which allows you to choose a number of rooms, and for each room choose the number and size of beds.
After the user has completed this task, the HotelConfigure class should print display this information back to the user.

## Vacancy

Add a property called hasVacancies to Hotel. A hotel has a vacancy if any room is vacant. Make any changes you need in the other classes to support this property, including HotelReport, HotelTest and HotelConfigure

# SUBMISSION NOTES:

Your submission should consist of a set of Java source code files. No other files are necessary, nor will be accepted. NESS has been set up to accept a relatively large number of Java files; this is simply an upper limit and not an expectation.

Where class names are given in the specification, you should use these class names exactly as given. These named classes should all be `public` and, therefore, in files with the same names.

All classes should be in the same Java package, which may be the default package.

## MARKING SCHEME:

Marks will be allocated as follows:

- Core Data Model: 70%
- Report/Test/Configure classes: 30%

For each of these:

- Correctness and Completeness: 70%
- Style and Readability: 30%

You will attract marks for:

- Good use of get/set methods
- Appropriate use of types
- Correct calculations
- Good use of object orientation
- Good error checking
- Use of privacy
- Appropriate constructors
- Correct indentation
- Good variable naming
- Clear use of comments

Please note that, that we can also mark negatively where the code is directly against the specification (inappropriate use of user interaction as described above, for example), or against the design principles described in the lectures (for example, overuse of the `static` keyword).

Last updated 2016-11-03 16:20:59 GMT

# Appendix F Full coursework specifications

***Coursework Specification 2016/2017***

**CSC1021: Assignment 2**

**Aims**

The assignment will exercise:

- The use of the basic language features of Java
- Designing and building a solution to a specification

**Background**

As part of a larger project, building an online hotel management system, you need to design and implement a system for storing and calculating some key information about a hotel.

**Task**

Your task is to create a prototype which allows capturing knowledge about the current state of a hotel.

You will need at least three classes:

`Hotel`
: This should store all the essential information about a hotel, including a name and some rooms.

`Room`
: This should store the number of beds in a room.

`Bed`
: This should store the size of a bed (i.e. single or double).

You will need to instantiate multiple objects of these classes. Pay attention to the design principles that you have been taught during the module. For instance, every `Hotel` should have a `Name` property, which will be accessible by `get/set` methods and be stored in a `private` member variable.
For later stages of this coursework, you will build a simple user interface. This is a prototype system, so the user interface does **NOT** need to be complex; a simple command-line interface using `Scanner` and `println` statements is entirely sufficient. However, as your classes may be used in several different situations, none of `Hotel`, `Room` nor `Bed` should require any user interaction; use of `println` or `Scanner` within these three classes will be considered an error, and will be penalised.

There are a number of features that you should attempt to achieve; please complete these in the order given.

**HotelReport**

Create a class called `HotelReport` which, when given a `Hotel` object will produce a short, textual report describing the name of the hotel, the number of rooms and, for each room, lists the number and size of the beds.

**HotelTest**

The initial version of your system should allow you to create objects describing one hotel, with several rooms, each with one or more beds. The completed `Hotel` object should be able to calculate its maximum occupancy. You should write a simple class called `HotelTest` which contains a `main` method which demonstrates this working.

**HotelConfigure**

Create a simple class called `HotelConfigure`. This should have a `main` method which provides a simple user interface, which allows you to choose a number of rooms, and for each room choose the number and size of beds.
After the user has completed this task, the `HotelConfigure` class should print display this information back to the user.

**Vacancy**

Add a property called `hasVacancies` to `Hotel`. A hotel has a vacancy if any room is vacant. Make any changes you need in the other classes to support this property, including `HotelReport`, `HotelTest` and `HotelConfigure`

**SUBMISSION NOTES:**

Your submission should consist of a set of Java source code files. No other files are necessary, nor will be accepted. NESS has been set up to accept a relatively large number of Java files; this is simply an upper limit and not an expectation.

Where class names are given in the specification, you should use these class names exactly as given. These named classes should all be `public` and, therefore, in files with the same names.

All classes should be in the same Java package, which may be the default package.

**MARKING SCHEME:**

Marks will be allocated as follows:

- Core Data Model: 70%
- Report/Test/Configure classes: 30%

For each of these:

- Correctness and Completeness: 70%
- Style and Readability: 30%

You will attract marks for:

- Good use of get/set methods
- Appropriate use of types
- Correct calculations
- Good use of object orientation
- Good error checking
- Use of privacy
- Appropriate constructors
- Correct indentation
- Good variable naming
- Clear use of comments

Please note that, that we can also mark negatively where the code is directly against the specification (inappropriate use of user interaction as described above, for example), or against the design principles described in the lectures (for example, overuse of the `static` keyword).

Last updated 2016-11-03 16:20:59 GMT

***Coursework Specification 2017/2018***

**CSC1021: Assignment 2**

**Aims**

The assignment will exercise:

- The use of the basic language features of Java
- Designing and building a solution to a specification

**Background**

As part of a larger project, building an online hotel management system, you need to design and implement a system for storing and calculating some key information about a hotel.

**Task**

Your task is to create a prototype which allows capturing knowledge about the current state of a hotel.

You will need at least three classes:

`Hotel`
: This should store all the essential information about a hotel, including a name and some rooms.

`Room`
: This should store the number of beds in a room.

`Bed`
: This should store the size of a bed (i.e. single or double).

You will need to instantiate multiple objects of these classes. Pay attention to the design principles that you have been taught during the module. For instance, every `Hotel` should have a `Name` property, which will be accessible by `get/set` methods and be stored in a `private` member variable.
For later stages of this coursework, you will build a simple user interface. This is a prototype system, so the user interface does **NOT** need to be complex; a simple command-line interface using `Scanner` and `println` statements is entirely sufficient. However, as your classes may be used in several different situations, none of `Hotel`, `Room` nor `Bed` should require any user interaction; use of `println` or `Scanner` within these three classes will be considered an error, and will be penalised.

There are a number of features that you should attempt to achieve; please complete these in the order given.

**HotelReport**

Create a class called `HotelReport` which, when given a `Hotel` object will produce a short, textual report describing the name of the hotel, the number of rooms and, for each room, lists the number and size of the beds.

**HotelTest**

The initial version of your system should allow you to create objects describing one hotel, with several rooms, each with one or more beds. The completed `Hotel` object should be able to calculate its maximum occupancy. You should write a simple class called `HotelTest` which contains a `main` method which demonstrates this working.

**HotelConfigure**

Create a simple class called `HotelConfigure`. This should have a `main` method which provides a simple user interface, which allows you to choose a number of rooms, and for each room choose the number and size of beds.
After the user has completed this task, the `HotelConfigure` class should print display this information back to the user.

**Vacancy**

Add a property called `hasVacancies` to `Hotel`. A hotel has a vacancy if any room is vacant. Make any changes you need in the other classes to support this property, including `HotelReport`, `HotelTest` and `HotelConfigure`

**SUBMISSION NOTES:**

Your submission should consist of a set of Java source code files. No other files are necessary, nor will be accepted. NESS has been set up to accept a relatively large number of Java files; this is simply an upper limit and not an expectation.

Where class names are given in the specification, you should use these class names exactly as given. These named classes should all be `public` and, therefore, in files with the same names.

All classes should be in the same Java package, which may be the default package.

**MARKING SCHEME:**

Marks will be allocated as follows:

- Core Data Model: 70%
- Report/Test/Configure classes: 30%

For each of these:

- Correctness and Completeness: 70%

- Style and Readability: 30%

You will attract marks for:

- Good use of get/set methods
- Appropriate use of types
- Correct calculations
- Good use of object orientation
- Good error checking
- Use of privacy
- Appropriate constructors
- Correct indentation
- Good variable naming
- Clear use of comments

Please note that, that we can also mark negatively where the code is directly against the specification (inappropriate use of user interaction as described above, for example), or against the design principles described in the lectures (for example, overuse of the `static` keyword).

Last updated 2017-11-22 12:10:56 GMT

***Coursework specification 2018/2019***

**CSC1021: Assignment 2**

**Aims**

This assignment will exercise:

- The use of the basic language features of Java
- Designing and building a solution to a specification

**Background**

As part of a larger project, building an online ordering system for a Sandwich company, you need to design and implement a system for calculating the costs and some other features about a sandwich. The system will allow you to build a sandwich from various ingredients.

If you are not much of a Sandwich expert, you can see a nice example of this kind of menu system at Subway. (I offer no opinions on their sandwiches one way or the other).

**Task**

Your task is to create a prototype which allows cost calculations for sandwiches.

You will need at least four classes.

`Sandwich`
: This should carry all the essential information about a sandwich, including some bread and some toppings.

`Bread`
: This class should describe the bread.

`Filling`
: This class should describe the fillings.

`Topping`
: This class should describe any toppings or sauces

You will need to instantiate multiple objects of each of these classes. You will need several `Bread` objects, and at least four `Filling` objects and a number of `Topping`. You will need to store this in an appropriate data structure for use within your program.
Pay attention to the design principles that you have been taught during the module. For instance, I would expect that each ingredient will have a `Cost` property, which will be accessible by `get/set` methods, and be stored in a `private` member variable.

For later stages of this coursework, you will build a simple user interface. This is a prototype system, so the user interface does **NOT** need to be complex; a simple command-line interface using `Scanner` and `println` statements is entirely sufficient. However, as your classes may be used in several different situations, none of `Sandwich`, `Bread`, `Filling` nor `Topping` should require any user interaction; use of `println` or `Scanner` within these three classes will be considered an error, and will be penalised.

There are a number of features that you should attempt to achieve; please complete these in the order given.

**Build a Sandwich**

The initial version of your system should allow enable you to create objects for a sandwich, with one kind of bread, several fillings and a sauce or topping. Each of the ingredients should have a cost associated with it. The finished sandwich should calculate it's cost, both inclusive and exclusive of tax. You should write a simple class called `SandwichBuild` which contains a `main` method which demonstrates this working.

**Create a user interface**

Create a new class called `SandwichChoice`. This should have a `main` method, and class should provide a simple user interface. This should provide you with a choose of one bread, a number of fillings and toppings. It should compose the objects that you created in the previous task. Once complete, your code should print a short receipt showing all the ingredients, the price for each and the total cost. This should be formatted for easy reading. You may create additional classes, beyond `SandwichChoice` should you find this useful.

**Named Sandwich**

Sometimes, there can be too many choices in life. Write a new class `SandwichMenu`, with a main method, which allows the user to choose from a menu of predefined sandwich's. For instance, a Caprese panini consists of tomato, mozzarella, with an basil topping, using a piada bread. You should provide the user with a choice of at least three sandwiches. As before, you should print a short report showing all the ingredients. You may create additional classes beyond `SandwichMenu` should you find this useful.

**A Facetted Browser**

Most websites use a Facetted Browser to allow users to search through a complex set of options. This provides a series of criteria that you can use to narrow down your options.

Modify your `Sandwich`, `Bread`, `Topping` and `Filling` classes so that they support description of a number of different facets — you should support at least `Vegetarian` and `NutFree`, but could also add `DiaryFree`, `Vegan`, `GlutenFree` and `Spicy` as other options. For example, a sandwich is vegetarian if and only if all the ingredients are. Extend the reports produced by the last set of tasks to show clear which characteristics a sandwich fulfils (e.g. it is vegetarian, nut free and spicy).

**SUBMISSION NOTES:**

You should write a short report. This should include:

- A description of which tasks your code achieves.
- Output from the program(s) for a defined set of input.

For example, you report might say:

Task: Hello World

My "Hello World" program fulfils all parts of the specification.

If run without input it prints:

Hello World

If run with sample input "Phil", it prints:

Hello, Phil

Each description should be on a new page, and should fit on one page.

- A one-page reflective description of your design, including: use of privacy, inheritance, interfaces or abstract classes.

Submission:

- You should submit an electronic version of report as a Word document.
- In addition you must submit an electronic version of your Java source code using NESS.
- Submission should be in an appropriate form. Code should **NOT** be copied into Word or Notepad, but should be left as source files.

**MARKING SCHEME:**

Java Code 80% Report 20%

Marks for code are split evenly between the four tasks and also split evenly between correct functioning of the code and good use of design (variable/method names, formatting, comments, privacy, use of classes and abstraction.

Last updated 2018-11-21 12:48:57 GMT

## Appendix G Unedited Assessment Experience Questionnaire

Please answer every item quickly by giving your immediate response.
Circle the appropriate code number to show your response to assessment.

| | strongly disagree | disagree | ? | agree | strongly agree |
|---|---|---|---|---|---|
| **1 Amount and distribution of study effort** | | | | | |
| I do the same amount of study each week, regardless of whether an assignment is due or not. | 1 | 2 | 3 | 4 | 5 |
| I can be quite selective about what I study and learn and still do well. | 5 | 4 | 3 | 2 | 1 |
| I only study things that are going to be covered in the assignments. | 5 | 4 | 3 | 2 | 1 |
| I have to study regularly if I want to do well on the course. | 1 | 2 | 3 | 4 | 5 |
| On this course, it is possible to do quite well without studying much. | 5 | 4 | 3 | 2 | 1 |
| In weeks when the assignments are due I put in many more hours. | 5 | 4 | 3 | 2 | 1 |
| **2 Assignments and learning** | | | | | |
| Tackling the assignments really makes me think. | 1 | 2 | 3 | 4 | 5 |
| I learn more from doing the assignments than from studying the course material. | 1 | 2 | 3 | 4 | 5 |
| In completing the assignments you can get away with not understanding and still get high marks. | 5 | 4 | 3 | 2 | 1 |
| The assignments give very clear instructions about what you are expected to do. | 1 | 2 | 3 | 4 | 5 |
| When I tackle an assignment it is not at all clear what would count as a successful answer. | 5 | 4 | 3 | 2 | 1 |
| The assignments are not very challenging. | 5 | 4 | 3 | 2 | 1 |
| **3 Quantity and timing of feedback** | | | | | |
| On this course I get plenty of feedback on how I am doing. | 1 | 2 | 3 | 4 | 5 |
| The feedback comes back very quickly. | 1 | 2 | 3 | 4 | 5 |
| There is hardly any feedback on my assignments when I get them back. | 5 | 4 | 3 | 2 | 1 |
| When I get things wrong or misunderstand them I don't receive much guidance in what to do about it. | 5 | 4 | 3 | 2 | 1 |
| I would learn more if I received more feedback. | 5 | 4 | 3 | 2 | 1 |
| Whatever feedback I get comes too late to be useful. | 5 | 4 | 3 | 2 | 1 |

| 4 Quality of feedback | strongly disagree | disagree | ? | agree | strongly agree |
|---|---|---|---|---|---|
| The feedback mainly tells me how well I am doing in relation to others. | 5 | 4 | 3 | 2 | 1 |
| The feedback helps me to understand things better. | 1 | 2 | 3 | 4 | 5 |
| The feedback shows me how to do better next time. | 1 | 2 | 3 | 4 | 5 |
| Once I have read the feedback I understand why I got the mark I did. | 1 | 2 | 3 | 4 | 5 |
| I don't understand some of the feedback. | 5 | 4 | 3 | 2 | 1 |
| I can seldom see from the feedback what I need to do to improve. | 5 | 4 | 3 | 2 | 1 |
| **5 What you do with the feedback** | | | | | |
| I read the feedback carefully and try to understand what the feedback is saying. | 1 | 2 | 3 | 4 | 5 |
| I use the feedback to go back over what I have done in the assignment. | 1 | 2 | 3 | 4 | 5 |
| The feedback does not help me with any subsequent assignments. | 5 | 4 | 3 | 2 | 1 |
| The feedback prompts me to go back over material covered earlier in the course. | 1 | 2 | 3 | 4 | 5 |
| I do not use the feedback for revising. | 5 | 4 | 3 | 2 | 1 |
| I tend to only read the marks. | 5 | 4 | 3 | 2 | 1 |
| **6 The examination and learning** (only to be completed if there is an exam) | | | | | |
| Preparing for the exam was mainly a matter of memorising. | 5 | 4 | 3 | 2 | 1 |
| Doing the exam brought things together for me. | 1 | 2 | 3 | 4 | 5 |
| I learnt new things while preparing for the exam. | 1 | 2 | 3 | 4 | 5 |
| I understand things better as a result of the exam. | 1 | 2 | 3 | 4 | 5 |
| I'll probably forget most of it after the exam. | 5 | 4 | 3 | 2 | 1 |
| In the exam you can get away with not understanding and still get good marks. | 5 | 4 | 3 | 2 | 1 |

Comments you would like to make about the way the assessment affected your learning on the course

# Appendix H Ethical Approval

## Ethics Form Completed for Project: Using Ipsative Assessment to improve Feedback Quality and the Student Assessment Experience in University Computer Science.

Policy & Information Team, Newcastle University <noreply@limeservice.com>

Tue 13/12/2016 13:43

Inbox

To:Ryan Crosby (PGR) <r.crosby@newcastle.ac.uk>;

Ref: 10408/2016

Thank you for submitting the ethical approval form for the project 'Using Ipsative Assessment to improve Feedback Quality and the Student Assessment Experience in University Computer Science.' (Lead Investigator:Ryan Crosby). Expected to run from 26/09/2016 to 28/06/2019.

Based on your answers the University Ethics Committee grants its approval for your project to progress. Please be aware that if you make any significant changes to your project then you should complete this form again as further review may be required. If you have any queries please contact res.policy@ncl.ac.uk

Best wishes

Policy & Information Team, Newcastle University Research Office

res.policy@ncl.ac.uk

*Example Consent Forms*

Newcastle University

## Informed Consent Form

I, the undersigned, confirm that (please tick box as appropriate):

| | | |
|---|---|---|
| 1. | I understand the reason for this experiment and consent to take part and have read the information sheet dated 07/02/2017 | ☐ |
| 2. | I have been given the opportunity to ask questions about the project and my participation. | ☐ |
| 3. | I voluntarily agree to participate in the project. | ☐ |
| 4. | I understand I can withdraw at any time without giving reasons and that I will not be penalised for withdrawing nor will I be questioned on why I have withdrawn. | ☐ |
| 5. | The procedures regarding confidentiality have been clearly explained (e.g. use of names, pseudonyms, anonymisation of data, etc.) to me. | ☐ |
| 6. | If applicable, separate terms of consent for interviews, audio, video or other forms of data collection have been explained and provided to me. | ☐ |
| 7. | The use of the data in research, publications, sharing and archiving has been explained to me. | ☐ |
| 8. | I understand that other researchers will have access to this data only if they agree to preserve the confidentiality of the data and if they agree to the terms I have specified in this form. | ☐ |
| 9. | I, along with the Researcher, agree to sign and date this informed consent form. | ☐ |

**Participant:**

| ______________________ | ________________________ | ______________ |
|---|---|---|
| Name of Participant | Signature | Date |

**Researcher:**

| ______________________ | ________________________ | ______________ |
|---|---|---|
| Name of Researcher | Signature | Date |

## Informed Consent Form

Information Sheet

28/09/2017

- This focus group aims to collect student views on the Student experience at Newcastle University.
- The focus group will be audio recorded and transcribed by the researcher who ran the focus group.
- The collected data both transcribed audio and the audio itself will be kept within accordance to the data protection act 1998.
- If you have any issues or further questions, contact r.crosby@ncl.ac.uk

# Appendix I – Feedback Criteria Percentage Results

| Feedback Criteria | 2012/2013 | 2013/2014 | 2014/2015 | 2015/2016 | 2016/2017 |
|---|---|---|---|---|---|
| **1** | 66.1 | 69.6 | 73.3 | 70.4 | 77.6 |
| **2** | 74.8 | 93.5 | 75.9 | 75.5 | 97.1 |
| **2a** | 82.6 | 85.9 | 85.1 | 85.1 | 91.7 |
| **3** | 73.0 | 74.5 | 81.5 | 92.6 | 100 |
| **4** | 61.7 | 67.4 | 78.5 | 78.7 | 76.8 |
| **5** | 60 | 61.4 | 79.5 | 84.7 | 82.2 |
| **6** | 20 | 40.2 | 63.1 | 17.1 | 14.9 |
| **7** | 27.8 | 37.5 | 61.5 | 47.2 | 38.2 |

# Appendix J – Feedback Criteria Example

***Matching Feedback Criteria Results***

**Exercise : Project 2**

You make use of inheritance, but your usage isn't particularly suitable. PizzaBase isn't a type of PizzaTopping and Pizza isn't a type of PizzaBase or PizzaTopping.

Your PizzaTopping has a number of map and lists. This can be troublesome -- you have to change data in a number of places if you add/remove a new topping. You also check if it is vegetarian based on its index, which adds further complications.

Your user interaction is good

Style and formatting is generally very good. Just be careful to follow the style rules throughout. For example, the bill method should start with a lower case, and you have a number of variable names that could be named better (e.g. 'p' for pizza).

Good overview of program and structure.

Good structure to your testing documentation, however you should provide at least 5 test cases. As well as testing the system as a whole, you should "build up" to it by doing unitised testing (i.e. one method or near as possible).

| | FC1 | FC2 | FC2a | FC3 | FC4 | FC5 | FC6 | FC7 |
|---|---|---|---|---|---|---|---|---|
| Researcher | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Checker | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

### *Feedback Criteria Does Not Match*

**Exercise : Project 2**

Your code shows a reasonable command of java, but it is very over complex. It would have made more sense to have multiple objects rather than using an int as a flag within both base and topping.

PizzaTopping contains multiple attempts to interact with the user which is in violation of the specification.

Your style is generally good, good use of privacy and comments.

The report is good.

Good tests in general, although they tend to cover too many classes at once, and would not aid well in debugging.

You have clearly made a significant effort here. Although your GUI is quite nice, you would could have thought more clearly about the specification and the core requirements. The Animation is quite nice.

| | FC1 | FC2 | FC2a | FC3 | FC4 | FC5 | FC6 | FC7 |
|---|---|---|---|---|---|---|---|---|
| Researcher | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| Checker | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |