On the comparability of different programming language routes through A-level Computer Science

Elizabeth Harrison

A Research & Development Project

Submitted for the MSc Educational Assessment 2020

DEPOSIT AND CONSULTATION OF THESIS

One copy of your dissertation will be deposited in the Department of Education Library via the WebLearn site where it is intended to be available for consultation by all Library users. In order to facilitate this, the following form should be completed which will be inserted in the library copy your dissertation.

Note that some graphs/tables may be removed in order to comply with copyright restrictions.

Surname	Harrison
First Name	Elizabeth
Faculty Board	Education
Title of Dissertation	On the comparability of different programming language routes through A-level Computer Science

Declaration by the candidate as author of the dissertation

1. I understand that I am the owner of this dissertation and that the copyright rests with me unless I specifically transfer it to another person.

2. I understand that the Department requires that I shall deposit a copy of my dissertation in the Department of Education Library via the WebLearn site where it shall be available for consultation, and that reproductions of it may be made for other Libraries so that it can be available to those who to consult it elsewhere. I understand that the Library, before allowing my dissertation to be consulted either in the original or in reproduced form, will require each person wishing to consult it to sign a declaration that he or she recognises that the copyright of this thesis belongs to me. I permit limited copying of my dissertation by individuals (no more than 5% or one chapter) for personal research use. No quotation from it and no information derived from it may be published without my prior written consent and I undertake to supply a current address to the Library so this consent can be sought.

3. I agree that my dissertation shall be available for consultation in accordance with paragraph 2 above.

Abstract

Fairness in assessment is as important as validity and reliability. When question or paper choice is used in an assessment it is important that the various routes are comparable so that fairness to all students is maintained. Previous literature on question choice suggests that offering choice often leads to significant differences in exam performance by route, sometimes by several marks. This dissertation compares the five possible routes through a Computer Science A-level assessment. One paper was responded to using a choice of five different programming languages and it was possible that not all languages were equally accessible to students or equally applicable to the exam paper tasks. Test bias was evaluated using recently developed IRT-based differential test functioning procedures that were developed to assess unfairness due to demographic characteristics. These enabled a detailed test-level, sub-section, and item-level analysis at various levels of student ability. Some small differences between two routes were found but overall, the assessment behaved in a fair and equitable manner. It is likely that this was achieved through careful question design and the strict application of a common paper and generic mark scheme. It is recommended that the performance of the programming paper is monitored in future exam series.

A few anomalies observed in the data suggested the possibility that classroom experience might have had more of an impact on students' performances than programming language choice. This suggests that there is a need for research targeted at better understanding how teachers are preparing their students for the exam, and how the students are approaching the paper.

While choice has not led to unfairness in this analysis, concerns around the use of choice in assessment remain. Further research is proposed to look at the impact of choice on exam fairness in a broader selection of subjects that do not use generic questions or mark schemes. In addition, research is suggested to better understand student and teacher attitudes to the pros and cons of offering choice in assessment.

Finally, it is suggested that as students are judged on their overall test scores, rather than their item scores, test fairness should be analysed at test level, rather than item level as is common practice. The use of differential test functioning analysis is recommended for any assessment where concerns of unfairness towards a sub-group of students exist.

Table of Contents

Abbreviations

A-level	Advanced level (a high-stakes summative assessment usually taken at age 18)
AIC	Akaike information criterion
AO	Assessment objective
BIC	Bayesian information criterion
DfE	Department for Education (government department)
DFIT	A range of DIF and DTF statistics
DIF	Differential item functioning
DBF	Differential bundle functioning
DTF	Differential test functioning
GCE	General certificate of education, a group of qualifications, including A-level
GCSE	General certificate of secondary education, a group of qualifications usually taken at age 16
ICC	Item characteristic curve
IRT	Item response theory
JCQ	Joint council of qualifications. This body is made up of and represents all UK exam boards and awarding organisations.
Ofqual	Office of Qualifications and Examinations Regulation
PCA	Principal component analysis
PCM	Partial credit model
SIBTEST	Simultaneous item bias test
TCC	Test characteristic curve
sDBF	Signed differential bundle functioning statistic
uDBF	Unsigned differential bundle functioning statistic
sDTF	Signed differential test functioning statistic
uDTF	Unsigned differential test functioning statistic

List of Tables

List of Figures

1 Introduction

Students need to be confident that their attainment will be measured fairly when their learning is assessed (Osterlind, 1983). High-stakes, summative assessments should discriminate between students only on the relevant construct and be uncontaminated by irrelevant factors. For example, when a qualification involves a choice of questions, essays, or papers as part of its assessment, all possible routes to achieving the qualification should be equivalent in both difficulty and demand. If a particular route were found to be easier to do well in, any belief in the equivalence of each route would be incorrect and the principal of fairness would be undermined.

Choice, or optionality, is frequently used in the assessment of UK A-level qualifications, which might be offered to teachers in terms of choosing the topics they teach, or to students who then choose which questions to answer within a paper. There has been relatively limited research published on its use (Bramley & Crisp, 2017). One A-level where choice is offered is ExamBoard's[1] Computer Science. Teachers choose between one of five different programming languages to teach their students. This dissertation looks at the equivalence of the optional routes[2] used to teach and examine A-level Computer Science.

1.1 An introduction to A-levels

A-levels (advanced levels) are high-stakes public qualifications taken by UK[3] students usually at age 18; and are often required for university entrance. Typically, students study three or four separate subjects over a period of two years. A wide variety of subjects are available, and students choose from those offered by their school or college. In most subjects, A-levels are assessed using written examinations at the end of the study period, but occasionally practical and written coursework form all or part of the assessment. Written papers for A-levels are often mixed format; using multiple-choice, short answer, and constructed response questions. For security reasons, A-level assessments are not pre-tested, that is each question used in an exam paper has not been pilot-tested on a similar group of students to verify that it will perform as expected before administering to the intended cohort, and each

[1] A ficticous name has been used for the exam board.
[2] The term route is usually applied to the combinations of questions a student used to complete a paper that uses question choice. Here, the term is also applied to the five possible routes available to achieve ExamBoard's A-level Computer Science.
[3] In Scotland most take the Scottish qualification: Advance Higher.

paper is a “one-off” that will not be used again. Nonetheless, for each subject, the broad structure, range of mark tariffs (maximum marks) and weightings of assessment objectives used in the final examination remain the same each year.

For each A-level, the total mark across all its component parts is summarised as a grade. Psychometric models such as Rasch or Item response theory (IRT) models are not employed to optimise how question marks are combined, or to create scaled, standardised scores. The final overall mark is reported as a grade: A* (high), A, B, C, D, E (low) or U (unclassified/fail). It is these grades that students share as evidence of their attainment in future applications for higher education or employment. Consequently, it is important to all stakeholders (e.g. students, schools, universities, colleges, or employers) that the grades always carry the same value every year. A-levels are offered by a limited number of exam boards that are regulated by Office of Qualifications and Examinations Regulation[4] (Ofqual). Conventional common-item equating techniques cannot be used to link the marks between exam boards or across years as no questions are common across them. In the interests of fairness to former and current students, standards are maintained using the comparable outcomes process (for more details see: Bramley & Vidal Rodeiro, 2014; Ofqual, 2018; Robinson, 2008). This process provides statistically recommended grade boundaries that account for changes from previous years in both the prior attainment profile of candidates, and in the difficulty of the papers. Awarding meetings are then held to enable examiners to scrutinise scripts from a sample of students with total marks on and around the recommended grade-boundary marks. Occasionally, examiners make minor adjustments to grade boundaries away from those suggested by the comparable outcomes process, if they judge this necessary to better reflect the desired standard.

1.2 Teaching and assessment of computer science in the UK

There has recently been a drive in the UK to see more students studying and using computing[5] throughout their schooling. This includes encouraging students to gain qualifications in the academic subject of computer science at senior school, rather than the formerly popular vocational subject, ‘ICT’ - Information and Communication Technologies (RS, 2012, 2017). An initial Royal Society report emphasised that “computing is of enormous importance to the economy” (RS, 2012) but observed that enthusiasm for the subject is dwindling and there are few specialist

[4] Ofqual is a non-ministerial government department responsible for regulating qualifications in England.
[5] The term *computing* covers ICT, IT and digital literacy (RS, 2012).

teachers able to teach it. They noted that achieving greater student engagement with computing requires additional teacher recruitment, professional development for current teachers, and investment in school infrastructure. In their second report, the Royal Society notes some improvement has occurred but "evidence shows that computing education across the UK is patchy and fragile" (RS, 2017). As a result, they recommend urgent action to increase the uptake of computer science teaching and learning in schools.

1.2.1 Computer Science at ExamBoard

ExamBoard's precursor first offered computer science as an A-level qualification in 1970. It was then, and still is, a rigorous, academic branch of the field of computing, requiring a considerable amount of mathematical skills. Currently, ExamBoard offers their GCSE[6] and A-level qualifications in computer science with a choice of programming language to enable teachers to teach in a language they know well and to avoid potentially reducing student entry numbers by restricting teachers to a single language. At A-level, the programming languages are C#, Java, Pascal/Delphi[7], Python and VB.NET. Entry numbers for each language vary considerably. The A-level is assessed using two written exam papers and an extended piece of coursework[8] undertaken in the months prior to the written papers. The programming language is not declared on the student's qualification certificate.

As mentioned above, assessment of A-levels normally involves either hand-written responses to exam papers or coursework. Unusually, one component of ExamBoard's computer science assessment is undertaken onscreen[9] in exam conditions. Hence, candidates can write and run code before they submit it for marking. This means that examiners can assess which elements of coding students can perform without support (unlike in the coursework where students could avoid certain procedures or look at reference materials). The onscreen assessment is made up of a single set of questions that exam writers believe are equally accessible for students in each programming language. About 30% of the questions are not language specific, whereby the remaining 70% require responses in program code. Half of the questions are related to pre-released material[10]. Students must respond to

[6] GCSEs are qualifications taken at 16 years old.
[7] Hereon referred to as Pascal.
[8] The coursework involves a practical investigative project on a topic that interests the student.
[9] Access to the internet and any other source of help is disabled.
[10] The preliminary material, includes a Skeleton Program (available in each of the Programming Languages) for use in the exam.

the programming-language dependent questions in the chosen language, as the relevant software application will have been previously set-up for them. A common mark scheme is used for all programming languages, although example responses are given in each language. The second paper is undertaken in the usual way, a pen and paper written response, and it concerns computer science that is less directly related to programming, such as computer architecture, databases and networks, and written code is not required in the responses.

Responding with a particular skillset is an unusual use of optionality but not unique to computer science. A similar approach can be taken in religious studies where students respond to philosophical or ethical questions from within the context of their chosen faith. The comparability of the optional routes through A-level computer science is important for fairness, particularly because the outcomes are used for entry into university courses. Appendix A lists BSc Computer Science course entry requirements for the top twenty ranked UK universities for the subject. Few of these universities require students to have taken A-level computer science prior to entry but if the subject *is* taken, in most cases grade A is required. For schools, a key performance measure is the percentage of students achieving grades AAB or higher, including at least two facilitating subjects[11]. Therefore, in the interests of fairness to schools and students, the comparability of the optional routes should be assessed as part of any post-hoc review of the exam's performance, which can then inform the development of future test papers. The next sections will address the issue of fairness, the history of its importance to assessment, and how the use of optionality may have an impact on the fairness of assessment.

1.3 Test fairness

Fairness in any assessment goes beyond a consistent understanding of the meaning of each grade over time. It also means that each year students of comparable ability should be equally likely to achieve that grade, irrespective of their demographic characteristics (Camilli, 2006) or the questions they answered if there was optionality of any form (Bramley & Crisp, 2017). Fair assessment also concerns social equity (Camilli, 2006): for example, a further issue when choice is made by teachers is how much their choice reflects the resources available to them and their schools.

[11] Computer science is not a facilitating subject, so a computer science student would only be included in this measure if their other A-levels were two of: Mathematics, Further mathematics, Biology, Chemistry, Physics, English literature or a modern foreign language.

1.3.1 A history of the concept of fairness in educational assessment

The understanding of what constitutes a fair assessment has evolved considerably over the 20th and 21st centuries. Historically, assessments were done one-to-one between the assessor and the student. Introducing written exams in the 18th century gave a level of anonymity to an assessment (Isaacs, Zara, Herbert, Coombs, & Smith, 2013), meaning that students were far more likely to be judged on merit (Camilli, 2006;.Gipps & Murphy, 1994) However, the style of questioning and the contexts used could still introduce social bias. For example, early versions of the IQ test were later found to be fundamentally unfair due to cultural bias towards white Americans – candidates outside this group tended to perform poorly, reflecting environmental disadvantage rather than true ability (Gould, 1996). Hernstein & Murray (1994) quote the National Education Association who advocated in 1972 for a ban on standardised intelligence testing. They believed that a third of American citizen's suffered educational disadvantage before completing elementary (primary) school due to "… linguistically or culturally biased standardized tests". Prior to the mid-1960s, assessment experts tended to believe that their tests were objective; if a social or ethnic group scored poorly it was believed to reflect real group differences, in part because the differences confirmed widely held beliefs about some groups being intellectually weaker. Later this view was challenged by some educators who suggested that the frequently observed differences in achievement between these groups reflected differing learning opportunities (Cole & Zieky, 2001).

The 3rd edition of *Educational measurement* was the first to discuss test bias, which had been acknowledged as an issue during the US civil rights movement of the 1950s and 1960s, and the subsequent women's rights movement (Cole & Moss, 1989). They describe test bias as differential validity:

> … bias is differential validity of a given interpretation of a test for any definable, relevant subgroup of test takers.

In the same edition, Messick (1989), emphasised that the construct validity of a test must be considered for two reasons: "in justifying test use in terms of meaning and relevance, then in justifying it in terms of social consequences in the final analysis". For Cole and Moss (1989), showing a lack of test bias was a necessary part of demonstrating the validity of the test for the purposes for which it was designed.

The achieved levels of validity and fairness in an assessment are always a matter of degree, and therefore, declaring a test as fair is not a simple matter (Cole &

Zieky, 2001). Cole and Zieky suggested that four aspects of test fairness should be addressed. Firstly, by reducing group differences in key social and ethnic groups, and secondly, by ensuring all students can perform as well as they are able to through tests that adequately represent the construct. Thirdly, the misuse of test results should be discouraged. While it is not obvious that this responsibility lies with assessment experts, guidance can be made available "… to promote sound use and valid interpretations of the data by the media and other stakeholders …" (AERA, 2014). Finally, they emphasise that group concerns also apply to individuals; however, they note that it can be difficult to accommodate individual differences.

In the current (4th) edition of *Educational measurement,* Camilli (2006) uses the term test *fairness*, rather than test *bias*:

> Fairness in testing refers to perspectives on the ways that scores from tests or items are interpreted in the process of evaluating test takers for a selection or classification decision.

Like Cole and Moss (1989), Camilli describes fairness as closely linked to validity and how tests are used, but he adds that fairness is also about how the tests are presented and scored. He highlights that issues such as culture, ethnicity, sex, and socio-economic status should all be considered when developing a test to ensure that no relevant group is unfairly disadvantaged in their assessment.

Nisbet and Shaw (2019), in the UK context, suggest that fairness is now equally integral to producing a good assessment as the consideration of validity and reliability. They argue that fairness in assessment applies not only to those taking the test but to their peers (who have not taken the test), employers, higher education institutions, and society at large.

In response to this increasing concern, fairness is now considered at all stages of the development and administration of an assessment. Isaacs et al. (2013), comment that "lack of equity is a challenge to the validity of an assessment" and that it is important to monitor and review the outputs of any assessment.

1.3.2 Reviewing fairness using differential item functioning and differential test functioning analysis

A fairness review requires the consideration of both quantitative and qualitative evidence over a broad range of social issues (Camilli, 2006; Isaacs et al., 2013). This is particularly important if the test is summative and the result is used for selection on a future course or for employment (Camilli, 2006). The investigation should compare relevant subgroups to the majority, or reference group. Some

statistical analyses use an external criterion; a relevant measure of success. Regression techniques are then used to assess whether the relationship between the test scores and the criterion differ by group membership (Camilli, 2006). However, this approach assumes that a suitable criterion exists that is not itself subject to test bias (Cole & Zieky, 2001). Such a criterion is not always available.

Since the 1960s, more complex statistical procedures have been developed that looked at the behaviour of individual questions or items by subgroup, to assess whether invariant measurement has been achieved. It is important to acknowledge possible prior differences in underlying ability when estimating group differences as it is not appropriate to simply compare the group means and so, some correction for underlying ability should be made. Hence, these statistical procedures require a matching variable to act as a proxy measurement of ability (Cole & Zieky, 2001). As unbiased external scores on a related test rarely exist, methods have been developed such as differential item functioning (DIF) procedures, that use an internal measure of ability, to evaluate and compare the behaviour of individual items at student-group level. The internal measure could be the total test score, or the latent ability estimate derived from a Rasch or IRT model.

DIF procedures focus on the behaviour of individual questions, however, sometimes questions that individually appear fair can combine to create an unfair test score – this is known as differential test functioning (DTF). Where DTF is identified, test takers may have the same ability on the construct but achieve different total test scores (AERA, 2014). Procedures to detect DTF do exist, although there has been far less research in this area than into DIF (Osterlind & Everson, 2009). Three possibilities are: Raju's DFIT statistics (Raju, 1990; Raju et al., 2009, 1995), which are based on IRT models; the simultaneous item bias test (SIBTEST, Shealy & Stout, 1993), which is based on raw test scores; or the recently developed DTF statistics offered in the R package MIRT (Chalmers, Counsell, & Flora, 2016), which uses Rasch/IRT models.

A Rasch/ IRT approach is preferred here as it enables a detailed analysis of the test data at test and item level, and at various student abilities. In the UK, there has been criticism of use of the Rasch model in assessment design and analysis, with suggestions that this is not an appropriate approach (e.g. Goldstein, 1979), while others have countered these criticisms (e.g. Panayides, Robinson, & Tymms, 2010; Preece, 1980). Criticisms include the Rasch ideal that items should *fit the model*, rather than analysts choosing a model that *fits the data*, which turns the usual statistical modelling approach on its head. The Rasch ideal means the data should

be unidimensional, so that only one mark or position on the latent trait is necessary to summarise a student's performance (Goldstein, 1979). When summarising an A-level assessment for a student only one mark is given per paper, and in fact this is summarised further by just a single grade on the final certificate, so effectively this assumption is implicit to any A-level qualification. Other writers suggest the unidimensional assumption can be relaxed slightly (e.g. Linacre, 1998), in reality it is improbable that only one dimension exits in any assessment and what is important for a valid unidimensional analysis is that there is one dominant dimension that represents the main ability trait of interest. Nevertheless, model fit should always be fully assessed when a Rasch/IRT approach is taken.

Determining whether real bias exists in a test is not simple (Cole & Moss, 1989) and caution should be used when interpreting the results of quantitative analyses. On this point, Camilli warns (2013):

> Quantitative methods can improve an understanding of the link between cause and effect, but arguments solely based on the authority of statistical methods are both flawed and obfuscating.

Any observed differences might be due to an unfair test, but they could also be due to factors outside of the test, e.g. opportunity to learn (Camilli, 2006; Cole & Zieky, 2001).

1.4 Fairness in A-levels

Syllabus requirements are set by the department of education after consultation with stakeholders, so that they reflect current needs in society and the workplace (DfE, 2014). Each exam board offers its own interpretation of the requirements through a detailed specification that includes the required knowledge, skills and attributes, and sample assessment materials. Question-writing expertise forms an important part of developing questions that do not disadvantage any sub-group of students by avoiding offensive content, or unfamiliar language or contexts. The document *Fair access by design* provides guidance to UK exam boards on "how good qualification and assessment design can give all learners the fairest possible opportunities to show what they know, understand and can do" (CCEA, 2015).

Past papers and mark schemes are available to students and teachers, which can be used for exam preparation and ensuring that students are familiar with the structure of their assessments. The exam boards then work with Ofqual to ensure the consistency of standards year-on-year in each of the qualifications offered (Ofqual, 2018). Despite these processes, unfairness might occur, and undertaking a fairness

review ensures that this possibility is investigated, discussed, lessons learnt and if necessary, future papers adapted. Where optionality has been used, the performance of the various routes should be carefully reviewed.

1.5 Optionality and choice

The use of optionality in an exam allows different students to achieve the same qualification without responding to exactly the same set of questions. For example, Geography students could choose between a question on glaciation or on coastal erosion, having learnt about both. Or an English literature teacher might choose the book that students will be asked about, e.g. *Lord of the flies* or *Of Mice and Men*. The offering of choice could occur within a paper (at the time of the exam), or between papers (selected before the exam). Offering choice however "creates problems for the examiner in providing equitable assessment of each candidate's level of attainment" (Willmott & Hall, 1975). Ideally, careful test construction and marking standardisation[12] mean that no adjustments are required to enable students to be compared and rewarded fairly.

There are many reasons for the use of choice. It is believed to be to the students' benefit that they or their teachers can choose the topics or approaches that are best for them (Wainer & Thissen, 1994). Optionality also avoids all students in a particular year group studying, for example, the same book for English literature, or the same period of history (Bramley & Crisp, 2017), thus allowing a wider domain coverage over the cohort, if not always for individuals (Wainer & Thissen, 1994).

Where students are taught the whole syllabus, they can choose the questions they feel best allows them to demonstrate their understanding and skills. Student choice adds a level of complexity to the exam; unfortunately students do not always choose the best option for themselves (Bell, 1997; Wainer & Thissen, 1994; Wang, Wainer, & Thissen, 1993), and this leads to some confusion on the construct that is being assessed, i.e. it could also include the ability to choose an option well. If it is assumed that an assessment is measuring a unidimensional construct that can be summarised by a single mark, then optionality risks creating a multi-dimensional construct: ability on the construct *and* the ability to choose well in a short time frame (Bradlow & Thomas, 1998). It is also believed that offering choice represents a more realistic scenario, as everyday life often presents choice situations (Fitzpatrick & Yen,

[12] A process where senior examiners validate the mark scheme based on a few sample student scripts at the beginning of marking (Morin, Holmes, & Black, 2018). The meetings also enable marking team leaders to become familiar with the mark scheme.

1995; Bradlow & Thomas, 1998). However, it seems unreasonable to use a school leaving assessment to measure this life skill.

Where optionality is offered, either to students or teachers, examiners design the optional questions believing that they are comparable in demand and difficulty (Fitzpatrick & Yen, 1995; Wang et al., 1993). Pollitt, Ahmed, and Crisp (2007) differentiate between these concepts:

> … by 'difficulty' we mean an empirical measure of how successful a group of students were on a question; by 'demands' we mean the (mostly) cognitive mental processes that a typical student is assumed to have to carry out in order to complete the task set by a question.

They also point out that "the influence of choice on overall demand is complex".

However, demand is difficult to quantify and as examiners write a test, they must judge whether questions are of equivalent demand based on their knowledge and expectation of what students should be able to achieve. But even in the absence of choice examiners find it difficult to judge the demand of a question (e.g. Bejar, 1983; Impara & Plake, 1998), hence it seems unlikely that examiners can always achieve comparability when choice is offered. Statistical measures of difficulty can be used as a proxy for demand after the test has been sat and used to assess comparability.

When teachers make the choice of what is to be studied, the complication of whether students have chosen the best for themselves is removed, but the various choice routes still need to be comparable. For A-level computer science, choice is offered to allow teachers to use the programming language they are most familiar with. However, just as there is the risk that one book might be more accessible than another in an English literature course (Bell, 1997), one programming language may be easier to learn, or give a more appropriate approach to answering a question fully. Hence, offering teacher choice might not be in the student's best interest.

All types of choice in an assessment effectively leads to different forms of the assessment occurring, with many routes available to achieve the same qualification. The comparability of optional routes through a paper is difficult to confirm post-hoc. For example, when option-mean marks are calculated, some options may appear easier than another, but it might be that the students taking that option were more able or better taught. Some authors have suggested that when choice is used, post-hoc adjustments to the marks should be made (e.g. Wang et al., 1993; Wainer & Thissen, 1994). However, this may involve a considerable and disproportionate effort

(Gulliksen, 1950). Even in the case of a limited number of choices it might be difficult to adequately quantify the post-hoc adjustments needed.

Often (outside of the UK), when different versions of a test are offered, their marks are aligned through a process known as equating (Kolen & Brennan, 2010). Wainer and Thissen (1994) argue that equating for differences in difficulty can make assessment with optional questions fair but takes some effort and requires trust in assumptions about the responses that students would give if they had the opportunity to answer all questions. An equating process often requires some students who have taken both forms of the test (common students) or some questions that are repeated on the different tests (common questions), and this information then informs the linking process. However, non-IRT equating procedures often assume that any missing question data are missing at random, which is not the case when students have chosen to answer some questions and to skip others. Unless additional data are available, unaffected by optionality, untestable assumptions need to be made about the students, such as assuming that the response to each optional question would be the same for students who did choose it as for those who did not (Wang et al., 1993). If this assumption holds and equating can be applied, it effectively means that choice is only fair when it is unnecessary, as it also means there was no benefit to the student in making a choice. Wainer and Thissen (1994) described the problem that students face if they know that choice questions will be aligned post-hoc as:

> If no equating is done, the instructions to the examinee should be: Answer the item that seems easiest to you. (And we hope that the examinees choose correctly, but we will not know if they do not.) If we equate the choice items (give more credit to harder items than easier ones), the instructions should be: Pick that item which, after we adjust, will give you the highest score.

Similarly, if teachers choose the book, period of history or programming language, knowing that post-hoc adjustments will be made, they will need to anticipate which choice would lead to the most favourable adjustment for their students.

If the technical difficulties of effective equating could be solved, the advantage to the student of making a good choice would be lost. Even in the simpler scenario of teacher choice, conventional equating techniques would make assumptions about how the students might have performed on the optional questions or papers that the students were not prepared for.

Using IRT techniques, Fitzpatrick and Yen (1995) suggest that using common (non-choice) questions is the only reliable way to adjust question statistics for ability

differences. This analysis is only possible when the non-choice questions represent a reasonable proportion of the assessment and adequately cover the construct. The onscreen computer science paper has a subset of questions that can be answered without using a specific programming language. These questions could be used in any Rasch or IRT analysis to adjust for ability differences.

Bramley and Crisp (2017) evaluated the performance of within-paper optionality in a situation where there were several optional essays but no common questions by using a Rasch model to create an internal estimate of student ability. They demonstrated that the extent of any lack of comparability can vary across the ability range, for example differences in performance may only be apparent at low abilities. Bramley and Crisp (2017) caution against aligning scores:

> Attempting to mitigate the undesirable effects of question choice by making statistical adjustments to scores could well create as many problems as it solves, and may work against the laudable aims of transparency and fairness.

In the case of teacher choice, any form of post-hoc alignment risks the possibility that some teachers might switch the topics they teach each year. They might conclude that one option is easier than another when they review grade boundaries at the time results are released.

Despite these technical issues, there are subjects where not offering choice is unreasonable, e.g. history, where there a large number of topics or eras that could be studied at school (Bridgeman, Morgan, & Wang, 1997). Wainer and Thissen (1994) suggest that possibly a well-designed test could be developed where options are comparable without the need for equating. For example, when optionality is used for English literature essays the questions vary but, as Bell (1997) notes, examiners believe “that it is the response to the question that determines the final mark”, suggesting that mark schemes reward the same skills demonstrated by the students, whatever the question. This situation could be made more restrictive by using a common, generic mark scheme for all equivalent optional questions.

As mentioned above, ExamBoard’s A-level Computer Science is offered with a choice of programming language where several questions are answered using the chosen language. A generic mark scheme is also used to support fair and equitable assessment. It is possible that these questions and associated mark schemes are designed carefully enough to not result in different performances from equally able students; this needs to be verified.

1.6 Aim and research questions

The aim of this dissertation is to review whether fairness has been maintained in assessing A-level computer science by using different programming language options.

The following research questions are investigated:

1. Can a Rasch or IRT model be successfully applied to the computer science test data?
2. To what extent is differential test functioning (DTF) impacting the different programming-language choices on average?
3. Does the DTF vary with ability? If so, which ability levels are most effected and is this impacting outcomes at the key grades of A and B?
4. Is the DTF due to differential item functioning (DIF) in specific questions or is it accrued over several questions that individually do not show DIF?

2 Methods

As discussed in the introduction, test fairness and DTF cannot be analysed by simply comparing the mean scores of the groups under investigation – more sophisticated methods are required that take into account group-wise variations in ability. Here an IRT approach to DTF analysis has been adopted. The participants of the study are described below, together with the ethical issues considered. The structure of the assessment is described in the materials section, followed by information on the type of data available to use in the analysis, and details of the data preparation procedure. A brief outline of the Rasch and IRT models used is given, with details of model evaluation. Finally, the associated DTF and DIF identification methods are described.

2.1 Participants

Secondary data were used in this analysis, namely examination data gathered by the exam board from students taking A-level computer science. The majority of the students were 18 years old and attended either schools or further education colleges. The cohort comprised a mixture of domestic (UK) and international students. The current version of the A-level computer science assessment consists of two written examination papers that are usually taken at the end of two years of study, together with a coursework component (see Table 1). The papers have been available in the summers of 2017, 2018, and 2019. For each paper, electronic records of marked student responses were available at item level for 3,611 to 4,349 students per year, together with the coursework component mark. The largest and most recent year's (i.e. 2019) data have been analysed here. Of the three components, only Paper 1 is programming language specific[13] and therefore only Paper 1 has been analysed in detail. A few students entered for the subject did not complete all three components and hence only those that achieved a non-zero mark for Paper 1 were included in the analyses presented here (4,317 students).

[13] The coursework does not need to be undertaken in the same programming language as the written paper.

Table 1 Current structure of ExamBoard's Computer Science A-level assessment

Assessment component	Delivery method	Language specific	% of final overall mark
Paper 1	Written paper	✓	40%
Paper 2	Written paper	x	40%
Course work	Project report	x	20%

2.2 Ethical Issues

The CUREC application approval for this research is shown in Appendix B. As part of its role as an awarding organisation ExamBoard retain the right to process assessment data as necessary for their legitimate interests, e.g. to maintain and develop their core products and services in a regulated environment. It is on this basis that the data are used here: to inform development of future assessments. As part of the approval process, permission was given to use these data, which were anonymised by removing candidate numbers, demographic, and teaching centre data prior to receipt of the dataset for the analysis Therefore, no individual student or teaching centre was identifiable. All data were stored on an encrypted laptop or exam board drives, which are subject to secure infrastructure and protected by multiple firewalls.

2.3 Materials

Students answered Paper 1 using one of five possible programming languages: C#, Java, Pascal, Python, or VB.NET. The paper was worth 100 marks and the time allowed was 2 hours 30 minutes. It consisted of four sections and 42 questions, with tariffs that ranged from 1 to 12 marks (see Table 2), and there were no multiple-choice questions. Section A (29 marks) concerned theory of computation and student responses did not depend on the programming language they used. Although not programming language dependent, the skills and knowledge assessed in section A were logically linked to programming skills, including fundamentals of programming, data structures and algorithms. Sections B, C and D required program code to be written and executed as part of the examination. Two sections (C and D) were linked to pre-released material, including skeleton code, which was referred to, developed and modified during the exam. Table 2 also shows the assessment objectives (AOs) that were targeted in each section. The AOs and their mark

allocations are AO1: knowledge (20 marks); AO2: application and analysis (30 marks); and AO3: programming, design and evaluation (50 marks). Sections B and D assess AO3, which targets practical programming skills, the other sections mainly assess AO1 and AO2.

Table 2 Structure of 2019 A-level Computer Science Paper 1

Section	Number of items	Tariff Range	Marks	Topic/task	Assessment objective (marks)
A	16	1-4	29	Theory of computation	AO1 (13), AO2 (16)
B	2	1-12	13	Algorithmic task to be written in chosen language	AO3 (13)
C	16	1-4	23	Questions on pre-released skeleton code (available in chosen language)	AO1 (7), AO2 (14), AO3 (2)
D	8	1-12	35	Modify and run pre-released skeleton code in chosen language	AO3 (35)

AO1: knowledge; AO2: apply/analyse; AO3: program, design and evaluate

The same printed question paper was presented to students whatever their programming language choice (i.e. nothing in the printed paper was language-specific), but students responded in their chosen language. The software and pre-released material made available to them reflected that choice. Students sat the paper on-screen in supervised examination conditions. The examination centre provided computers with a restricted set-up, i.e. without access to the internet, wi-fi, spell checkers or disallowed material. The computers allowed the students to write and run program code. Each student was required to copy and paste their final program source code into an electronic exam booklet along with screen-captures to demonstrate code outcomes (Appendix C, example 1 shows a question with a screen-capture request). A printed copy of the electronic booklet was then submitted for marking. Examiners were not expected to assess student code by executing it themselves. Item level data were then recorded per student as part of the marking process, it is these data that were analysed here.

Table 3 shows summary statistics for Paper 1 for all students receiving a non-zero total mark, by option and overall. Python was by far the most popular option chosen by teachers (48% of entries), with Pascal being the least popular (5%) –

representing only 12 centres. The mean total scores vary by several marks. Pascal students achieved the highest mean score (52.0). Java students achieved the lowest results with a mean mark of 46.7. However, this does not mean that Pascal students were advantaged by the paper, or that Java students were disadvantaged, as no allowance has been made for the students' underlying ability, which may differ across centres by language choice.

Table 3 Entry numbers, paper total means and sds

	All	C#	Java	Pascal	Python	VB.NET
Entries	4317	778	300	205	2087	947
Mean total mark	48.5	48.9	46.7	52.0	48.7	47.7
Total mark SD	21.4	21.4	20.3	22.2	21.2	22.1

2.4 Procedure

A-level assessments are not intended to be speeded, that is, time should not be a limiting factor for students, as the majority should be able to complete the paper in the allowed time. However, the effect of speededness may impact student behaviour during the test, resulting in items being skipped, not reached, or poor and rushed responses given to items that are positioned later in the examination (Stafford, 1971). If an item is unreached by many candidates, estimates of its true difficulty can be compromised. Table 4 shows the tariff for each item and item attempt rates, by option and overall. There are four items with attempt rates <80%: A2.2 (an early item), and D12.2, D13.1 and D13.2 (the final three items). The non-attempt rate was similar across the five options. It seems reasonable to assume some students skipped item A2.2 but it is unclear whether the final items were also skipped or were genuinely not reached. Items D12.2 and D13.2 were the screen-capture items associated with programming items D12.1 and D13.1. Students who did not complete the code required to respond to items D12.1 or D13.1 would be unable to submit screen-captures as evidence of code outcomes. Hence, item D12.1 can be viewed as the penultimate task for students, and as it had a reasonable attempt rate at approximately 89% this suggests that speededness was not a great concern. Therefore, all missing values were re-coded as zero for the analysis, which has the advantage of reflecting current A-level marking practice where all non-attempts are treated as incorrect and given zero marks.

Table 4 Entries, mark tariffs, item percentage attempt rates overall and by option

Item	Tariff	All	C#	Java	Pascal	Python	VB.NET
A1.1	4	81.6	78.8	82.0	86.8	81.6	82.7
A1.2	2	96.1	95.9	97.3	97.1	95.8	96.3
A1.3	2	93.4	92.7	95.3	94.1	93.4	93.0
A2.1	5	94.8	94.5	98.3	97.1	94.4	94.4
A2.2	1	79.6	77.6	85.0	77.1	79.3	80.6
A2.3	1	94.0	92.8	97.3	95.6	94.0	93.7
A2.4	2	95.1	94.6	98.3	95.1	95.2	94.4
A3.1	1	98.9	98.6	100.0	100.0	98.6	99.2
A3.2	1	92.7	90.0	94.7	91.7	93.7	92.3
A3.3	1	89.4	87.1	92.0	89.3	89.9	89.1
A3.4	1	88.5	86.2	89.3	89.8	88.7	89.3
A3.5	2	95.8	94.7	97.0	96.1	96.1	95.8
A3.6	1	84.2	82.9	86.0	88.3	84.9	82.6
A4.1	1	97.2	97.2	99.0	97.6	97.0	96.8
A4.2	1	92.6	92.3	95.7	94.1	92.3	92.1
A4.3	3	98.1	97.7	98.3	100.0	98.0	98.1
B5.1	12	93.9	93.6	93.0	95.6	93.6	94.8
B5.2	1	75.4	73.7	69.3	75.6	78.5	71.8
C6	2	97.5	96.8	98.7	95.6	97.8	97.3
C7.1	1	98.3	98.6	99.0	97.6	98.2	98.3
C7.2	1	94.6	95.1	92.7	92.7	94.4	95.9
C7.3	1	94.1	94.1	96.3	93.7	94.7	92.1
C7.4	2	81.9	81.2	82.0	82.9	82.1	81.7
C7.5	4	86.4	86.6	88.0	89.8	86.1	85.7
C7.6	1	89.2	87.9	91.3	91.2	89.4	88.7
C8.1	3	87.3	86.2	88.7	90.2	87.9	85.7
C8.2	1	81.2	79.4	78.7	85.4	82.5	79.5
C8.3	1	84.3	82.3	82.7	84.4	84.9	85.4
C8.4	1	82.0	81.2	81.0	86.8	83.0	79.7
C9.1	1	97.8	97.0	98.3	96.6	98.1	97.7
C9.2	1	97.5	96.9	97.7	96.6	97.8	97.5
C9.3	1	91.2	91.8	94.7	92.7	90.9	89.8
C9.4	1	98.5	98.2	98.3	98.0	98.8	98.3
C9.5	1	96.6	96.4	97.0	96.6	96.8	96.1
D10.1	4	96.5	95.1	96.0	96.1	97.4	95.9
D10.2	1	89.7	85.3	88.3	87.3	92.6	87.6
D11.1	7	90.2	91.0	89.0	87.8	89.7	91.9
D11.2	1	78.0	77.0	76.0	76.6	78.8	78.0
D12.1	12	89.2	89.3	87.3	87.3	88.9	90.8
D12.2	1	70.2	72.8	71.0	71.2	69.9	68.4
D13.1	8	73.5	72.1	71.3	74.1	73.9	74.3
D13.2	1	58.7	59.0	54.7	61.5	58.9	58.7

2.5 Data Analysis

As discussed in the introduction, Rasch and IRT methods are not commonly applied to A-level examination data but have been chosen here as they offer an alternative estimate of student ability to the total test score for use in DIF and DTF analysis. The models describe a probabilistic mapping of the observed pattern of student-item responses from an assessment to a latent trait (θ) that is assumed to represent the ability that the test is designed to assess. Both student abilities and item difficulties can then be described in terms of their position on the latent trait, rather than in terms of mean or total marks.

The analysis was undertaken in several stages, firstly the suitability of using a Partial Credit Model (PCM) was confirmed. Secondly, IRT models were identified that allowed DTF to be evaluated while accounting for differences in underlying ability by option group, which are referred to as multiple-group models[14]. While test characteristic curves for all five options could then be simultaneously examined graphically, the statistical evaluation of DTF could only be calculated between two option groups at a time. Where statistically significant DTF was found, the impact on students at certain key grade boundaries was assessed. Finally, in the case of significant DTF between options, the individual items were assessed to see if any displayed noticeable DIF. Each of these steps is described in more detail below. All models have been fitted using R statistical software (R Core Team, 2018) and the MIRT package (Chalmers, 2012), which provides DTF and DIF assessment procedures. Some of the model evaluations and item summaries have been performed using the TAM package (Robitzsch A, Kiefer T, 2020), which offers the alternative PCM parameterisation shown below in equation (3).

2.5.1 The Rasch model

The Rasch model (1960) for dichotomously scored data describes a student's ability or an item's difficulty as a function of the responses on the exam paper. The items' difficulties are then described by a set of parameters β_i that give their location on the latent trait (θ), higher values of β_i indicate more difficult items. Student abilities

[14] The Pascal option was the least popular (see Table 4). There were two marks from two polytomous items that did not occur in the Pascal data (items D10.1 and D13.1). In order to achieve an assessment of DTF in the Pascal option, a low scoring 'dummy' candidate was added to the data, with scores in the unrepresented categories.

are also described as a location on the latent trait. The probability of a student giving a correct response increases as their level on the trait increases, i.e. higher values of θ indicates higher student abilities.

$$P(X_{ni} = 1) = \frac{e^{\theta_n - \beta_i}}{1 + e^{\theta_n - \beta_i}} \quad (1)$$

Where:

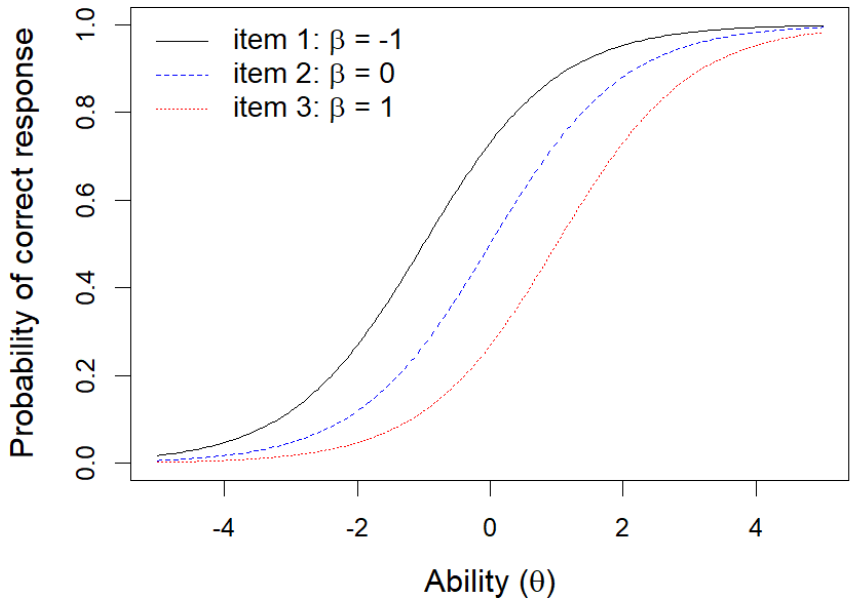
X_{ni} is the response (0 or 1) of the nth student on the ith item

θ_n is a parameter describing the ability of the nth student, $-\infty < \theta_n < \infty$

β_i is a parameter describing the difficulty of the ith item, $-\infty < \beta_i < \infty$.

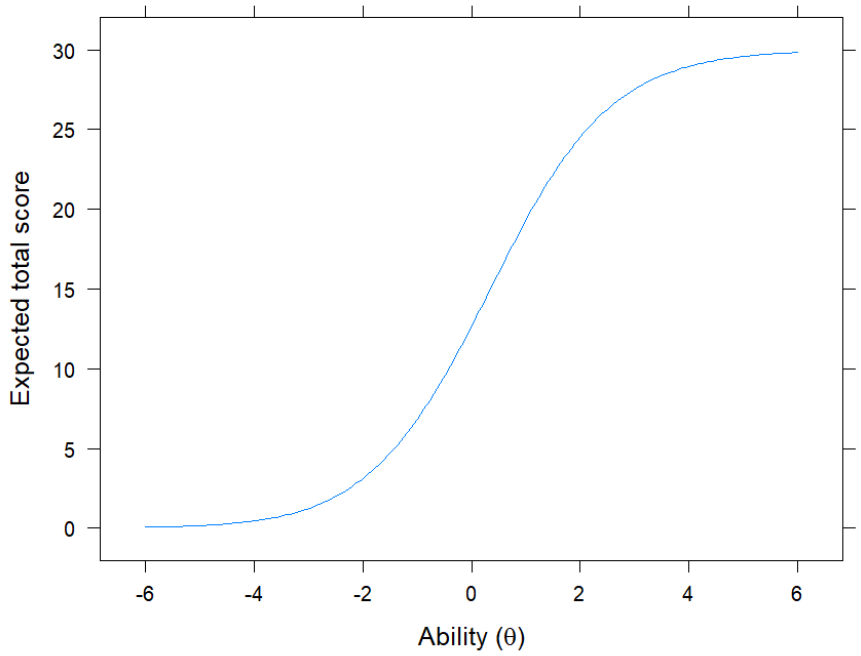
It can be shown that the log-odds (logit) of a correct response by student *n* on item *i* is given by the difference in their ability and the item's difficulty: $\theta_n - \beta_i$. When a student's ability is equal to an item's difficulty, the probability of a correct answer is 0.5. When their ability is greater than the item difficulty, the probability of a correct answer will be higher, and vice versa. For each item, the relationship between a student's ability and the chances of answering the question correctly can be illustrated graphically – see, for example, Figure 1, which shows the parallel curves for three items with various values of the β_i parameter. These curves are known as item characteristic curves (ICC). In Figure 1, item 1 is easier than items 2 and 3 in that students are more likely to give a correct answer to item 1 at any ability level.

Figure 1 Various item characteristic curves for the Rasch model



Students' predicted total-test scores can be calculated by aggregating their predicted item scores. Figure 2 illustrates the mapping between the theta estimates and the total score for a 30-item test made up of dichotomous items. This curve is known as the test characteristic curve (TCC).

Figure 2 Test characteristic curve for a Rasch model for a 30-item test



The Rasch model is also known as the one-parameter IRT model. The independently developed two-parameter IRT model (Birnbaum, 1968; Lord & Novick, 1968) adds an additional slope parameter allowing the ICCs to be non-parallel. For the Rasch model, the total score is a sufficient statistic for estimating person and item parameters in terms of the latent trait, meaning that students with the same total score will have the same position on the latent trait, whereas in the two-parameter IRT model, students with different patterns of responses but the same total score will differ in their positions on the latent trait. For A-level data, the Rasch model has an advantage over the two-parameter IRT model in that the resulting estimates of ability on the latent trait preserve the rank order of students based on total marks. This means that items retain the intended weight that the examiners created when allocating marks.

The partial credit model (PCM) form of the Rasch model is shown in equation (2) (Masters, 1982). The model allows items to have scores greater than one, where part marks on an item are awarded for partial success (otherwise known as polytomous items). The total number of marks for each question can vary. Each part

mark of an item is referred to as a category and must be awarded in a way that means increasing value represents increasing underlying ability.

$$P(X_{ni} = x|\theta_n) = \frac{exp\ (\sum_{s=1}^{x}(\theta_n - b_{is}))}{\sum_{r=0}^{m_i}[exp\ (\sum_{s=1}^{r}(\theta_n - b_{is})]} \quad (2)$$

Where:

$X_{ni} =$ observed response $(x = 0, \dots, m)$ for the n^{th} student on the i^{th} item where m_i is the maximum mark of item i

$\beta_{is} =$ threshold parameter for the associated category (mark) s of item i, $(s = 0, \dots, m_i)$

$\theta_n =$ parameter describing the position of the n^{th} student on the latent trait

The thresholds (β_{is}) of an item in equation (2) can be re-parameterised using a location plus deviation approach:

$$\beta_{is} = \beta_i + d_{is} \quad (3)$$

where β_i is known as the location parameter for item i, which describes the overall item difficulty (the average of all thresholds) and d_{is} describes the deviation from the location parameter at each adjacent category threshold. The location parameter is reported here to summarise PCM output.

Rasch and IRT models enable estimation of standard errors for the item parameters and the student ability estimates (Hambleton, Swaminathan, & Rogers, 1991) which facilitates the evaluation of the fit of item and student estimates. Each of the models described above assume that the latent trait is unidimensional, i.e., that there is no other significant dimension captured by the test other than the ability that the test was designed to measure. It is also assumed that, other than a common dependence on the latent ability, responses to the items are independent of each other; i.e. after considering student ability, there is no relationship between the response to pairs of items item (known as local independence). This assumption may be violated when a subset of items relate to the same topic in the context of assessing an ability that is reflected in various topics.

The PCM requires data from many students to evaluate fit. Adequate model fit is important because when the PCM fits the test data, student ability estimates are not test dependent and item parameter estimates are not dependent on the cohort of students who took the test (Hambleton et al., 1991). This means the results are generalisable to students who did not sit the same form of the test, such as occurs

when optionality occurs. If this assumption is reasonable, then useful insights can be gleaned from DIF and DTF analysis about the fairness of the test.

2.5.2 Evaluating model fit

An initial model was fitted to the raw data without taking programming option group into account. The adequacy of the model fit, and the assumptions made, should be verified before any conclusions are drawn about the fairness of the test. A-level assessments are often designed with related questions presented together referring to a common theme, which may lead to dependency between the responses to these questions. The existence of dependent items was assessed through analysis of item residuals after PCM models are fitted – the residuals should be uncorrelated (Andrich & Marais, 2019). Dependent items have been identified using the Q3 statistic (Yen, 1984) and flagged if |Q3|>0.3 (Andrich and Marais, 2019). Dependent items have been combined by adding their marks into a single item to remove violations of local dependence, and the PCM refitted and re-evaluated. Unidimensionality has been assessed using a principal component analysis (PCA) of the model residuals, which should not show evidence of a remaining dimension in the residual data. The person separation index (WLE) has been used to report reliability (Adams, 2005).

Individual item fit has been assessed using the Rasch measures of fit: infit and outfit mean square (Wright & Masters, 1990). The infit measure is outlier-robust and values should ideally lie between 0.8 and 1.2. Person fit has been assessed using infit-t, with person fit values lower than -2 or higher than 2 flagged as showing aberrant response patterns compared to model expectation.

2.5.3 Multiple-group model: anchor selection and equating options groups

Multiple-group models are IRT models that allow some or all the item parameters to vary by group. The PCM given in equation (2) could be viewed as a fully constrained multiple-group model, with no allowance made for group differences. To evaluate whether there are significant differences between the groups an initial fully independent multiple-group model was fitted assuming that the latent ability trait is distributed $N(0, \sigma^2)$ for all of the students from each of the five option-groups is the same, while allowing each item difficulty estimate to vary with group membership G – the independent model, equation (4).

$$P_i(X_{iGp} = x|\theta_p) = \frac{exp\ (\sum_{s=1}^{x}(\theta_p - b_{iGs})}{\sum_{r=0}^{m}[exp\ (\sum_{s=1}^{r}(\theta_p - b_{iGs})]} \quad (4)$$

Where:

$\theta_p \sim N(0, \sigma^2)$ for all groups, p=person (student)

b_{iGs} = threshold parameter category s of item i of group G

If the independent model shows a significantly better fit to the data, any large differences seen between group item-parameters could be due to both differences in ability and DIF. To assess DTF, it is important to first account for any differences in ability distributions that exist between the groups (Chalmers et al., 2016). Therefore, in order to align the item parameters effectively, the separate programming options need to be equated. The resulting equated model allows each group to be described with its own ability distribution: $\theta_p \sim N(\mu_G, {\sigma_G}^2)$ that will be scaled onto the common ability trait (θ). Once this is done any remaining differences in item parameter estimates reflect changes in difficulty alone, thus allowing DTF to be assessed. Ideally, the equated model should not differ significantly in fit from the independent model (Chalmers et al., 2016). The equated model is shown in equation (5), this was compared to the fully independent model of equation (4) using χ^2 likelihood ratio tests, the Bayesian information criterion (BIC), and the Akaike information criterion (AIC), where smaller values indicate a better fitting model.

$$P_i(X_{iGp} = x|\theta_{Gp}) = \frac{exp\ (\sum_{s=1}^{x}(\theta_p - b_{iGs})}{\sum_{r=0}^{m}[exp\ (\sum_{s=1}^{r}(\theta_p - b_{iGs})]} \quad (5)$$

Where:

$\theta_{Gp} \sim N(\mu_G, {\sigma_G}^2)$, G=option group, p=person (student), $\mu_1 = 0$

b_{iGs} = threshold parameter category s of item i of group G

Constraining $b_{iGs} = b_{is}$ over all option groups, for all items in the anchor

The process of equating the option groups requires the identification of a few items that are invariant over all options (i.e. showing no evidence of DIF), which can then be used to define the common latent ability trait (θ), these are referred to as the anchor set. Shealy and Stout, (2015) describe the anchor set as a valid subset of the test “against which the remainder is measured for bias”. However, Domingue et al.

(2017) argue that DTF can be fully assessed only if most of the items are allowed to differ by group and therefore it is important to constrain as few items as possible when selecting the anchor.

Identifying invariant items is a non-trivial task, but an uncontaminated anchor set is important, as later assessment of DIF and DTF can be undermined by a poorly chosen anchor (Woods, 2009). Anchor items were selected from the programming-language independent section (section A) so that all parameters for sections B to D items could vary by option choice. Items in section A should all be programming-independent, but as option choice might be linked to teachers' programming skills and experience in computer science, it is possible some of these items may also show DIF by language option.

2.5.4 Anchor selection using DIF procedures

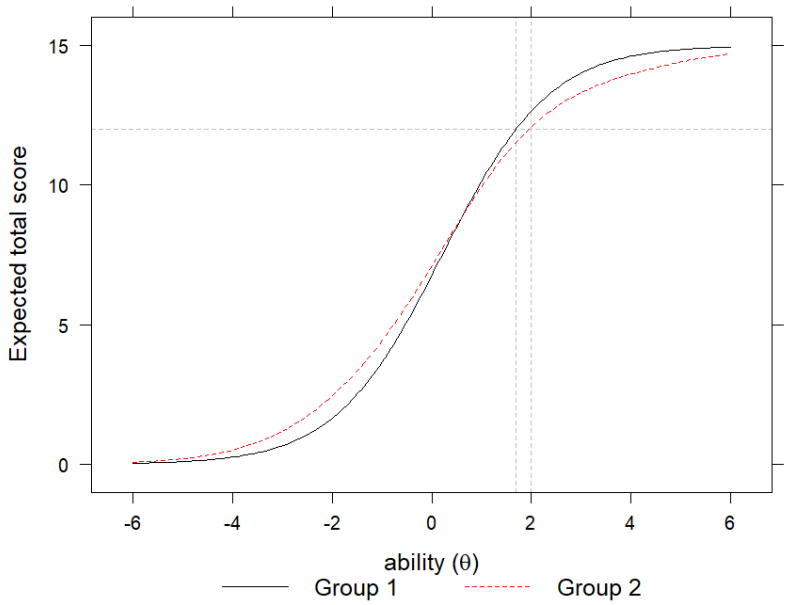
Many procedures have been proposed for selecting anchor items prior to an IRT DIF analysis; some of these were evaluated through simulation studies (e.g. Kopf, Zeileis, & Strobl, 2015; Lee & Geisinger, 2016; Woods, 2009). Unfortunately, none of these approaches were evaluated using the PCM. Therefore, here, a simple two-step likelihood-ratio approach to identifying DIF items was adopted to select items that showed no evidence of DIF. This involves comparing two nested IRT models using a likelihood-ratio test (Thissen, Steinberg, & Wainer, 1993). The first model was fitted with all item parameters constrained to be equal across the groups being compared with an assumed common latent trait distribution: the fully constrained model (2). Secondly, for each of the items in section A in turn, a model was fitted that allowed its threshold parameters to differ (with all other item parameters being held fixed). The new model was compared to the fully constrained model (2) using a likelihood-ratio χ^2 test. Items showing no evidence of DIF were then selected for the anchor, a conservative cut-off of $p>0.1$ was chosen.

The final, equated model (5) was then fitted, which allowed the abilities of the students from each option to be described in terms of their own mean and variance, and all non-anchor items parameters to vary by option. This model offers a simpler description of the data, with fewer parameters, however it should not be a significantly poorer fit to the data than the independent model (4) (Chalmers et al., 2016). As items may now have different location parameters by option, the rank order of students based on their latent ability estimate may differ from their raw total-mark rank order.

2.5.5 DTF assessment

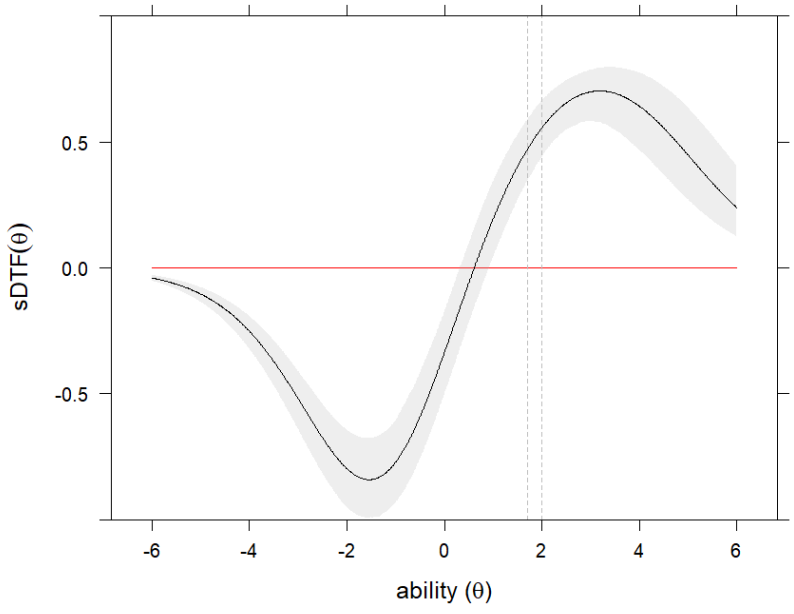
A joint plot of the option TCCs from the equated model gives a useful visualisation of any differences between the groups but does not indicate the significance of any apparent test bias, particularly when sample sizes are small. The methods offered by Chalmers et al. (2016) use bootstrap methods to enable estimation of the statistical significance of differences between TCCs while accounting for their sampling variability. Currently, the MIRT DTF procedure can compare only two groups at a time, meaning that several pairwise comparisons needed to be considered. For each comparison, overall DTF between the options was summarised using two statistics. Signed DTF (sDTF) is the average bias in one direction and identifies DTF that favours one group over another. Unsigned DTF (uDTF) summarises all deviations from a common TCC (average absolute bias). The two statistics have the same value when the TCCs do not cross. Figure 3 shows two TCCs that do cross, the uDTF summarises the average absolute difference between the curves is 0.453 marks with a 95% confidence interval of (0.39, 0.51). However, the two distinct areas cancel each other out and so the sDTF (the average difference between the curves) is −0.011 with a 95% confidence interval of (−0.10, 0.08). Figure 3 also shows a nominal cut-score of 12 marks, whereby a student in group 2 needs to have a higher ability level than a student in group 1 in order to achieve 12 marks.

Figure 3 Crossing TCCs for a 15-mark test, cut score of 12 marks shown with associated values in the latent ability trait



The MIRT DTF procedure uses bootstrap techniques to estimate the standard deviation of sDTF and uDTF, thus enabling the reporting of confidence limits. The statistics are reported on the same scale as the test i.e. in raw marks. The sDTF ranges in value from -TS to TS (TS=maximum total score, in this study TS=100), but uDTF can only take positive values and ranges from 0 to TS. As the lower bound for uDTF is zero, it is impossible to test the hypothesis uDTF=0 for the population but confidence limits can be reported. For sDTF, the hypothesis sDTF=0 can be tested. Both statistics were evaluated for abilities (θ) over the range [-6, 6] with 2,000 bootstrap samples. The value of the sDTF at any point on the latent trait ($sDTF_\theta$) is also available, with confidence limits, and can be used to assess differences in test performance in more detail. It can also be shown graphically with the confidence limits as illustrated in Figure 4, which shows the data from Figure 3 together with the two ability estimates associated with a score of 12 marks. The black curve indicates how $sDTF_\theta$ varies with ability, 95% confidence limits on $sDTF_\theta$ are shown as a grey area, and the red line indicates $sDTF_\theta$=0. For values of $\theta > 1$, the test has a positive sDTF and it favours group 1; for values of $\theta < 0$, sDTF is negative and the test favours group 2. If the shaded 95% confidence interval includes the red line, then at this point on the latent trait $sDTF_\theta$ does not differ significantly from zero. In Figure 4, at the abilities associated with a score of 12 marks ($\theta = 1.8$ or 2), $sDTF_\theta$ has a value of 0.4 to 0.5 marks and differs significantly from zero, p<0.05, and so at the cut score of 12, group 1 students outperform group 2 students by 0.4 to 0.5 marks.

Figure 4 $sDTF_\theta$ for the 15-mark test shown in Figure 3 (group 1 – group 2)



Ten pairwise comparisons can be made between the option groups and a subset of these were selected. Each of the most popular options were compared to each other (Python, VB.NET, C#), and the two minor choices (Java and Pascal) were compared to the most popular choice: Python. A Bonferroni correction was used to control the overall type I error rate: five comparisons were made at the 1% level in order to maintain an overall 5% type I error, and 99% confidence intervals were reported. The value of $sDTF_{\Theta}$ was considered at the relevant cut-points for the key grades of A and B on the paper.

2.5.6 Differential item functioning analysis

Where significant DTF was found the items were evaluated for DIF to assess whether any items were the main contributors to DTF. Using the equated multiple-group models, DIF in the unconstrained items of sections B to D was evaluated. A step-down procedure was adopted, which adds constraints to each item in turn by fixing the item parameters to be equal across groups. Each constrained model is compared to the equated model using a χ^2 test to assess for significant loss in model fit. As this process can identify many DIF items when a dataset is relatively large, those items associated with a change in BIC>10 were identified as showing 'strong evidence' of meaningful DIF (Raferty, 1995). ICCs for the identified DIF items were plotted to illustrate which groups appear to be advantaged by an item, and where in the ability range the bias occurred.

2.5.7 Differential Bundle Functioning analysis

The MIRT software also offers the facility to assess bias in a group of items: differential bundle functioning (DBF). This means that in addition to item-level (DIF) and paper-level (DTF) analysis, the relative performance of sub-sections of the paper can be evaluated. This has been assessed in the equated model for the sections B to D, where the response is dependent on the language choice made by teachers. Again, unsigned (uDBF) and signed (sDBF) versions are available, each has been assessed for values of θ over [-6, 6] from 2,000 bootstrap samples. If the overall sDTF and the programming language specific sDBF statistics differ, it indicates that items in section A are impacting overall performance differently for students in each option, despite being language independent.

3 Results

3.1 Descriptive results

Table 5 (see next page) shows the mean item scores overall and by option choice. The mean response to several items in section C appeared to vary by language option and this was also seen in items A3.4 and A4.2. Items A3.1 and C9.4 appeared very easy, and items A2.2 and A2.3 appeared very difficult. Figure 5 shows histograms for the high tariff items, for many of these the modal score was a high mark – except for the final item where the most frequent mark was zero (some of these will be non-attempts).

Figure 5 Histograms for high tariff items

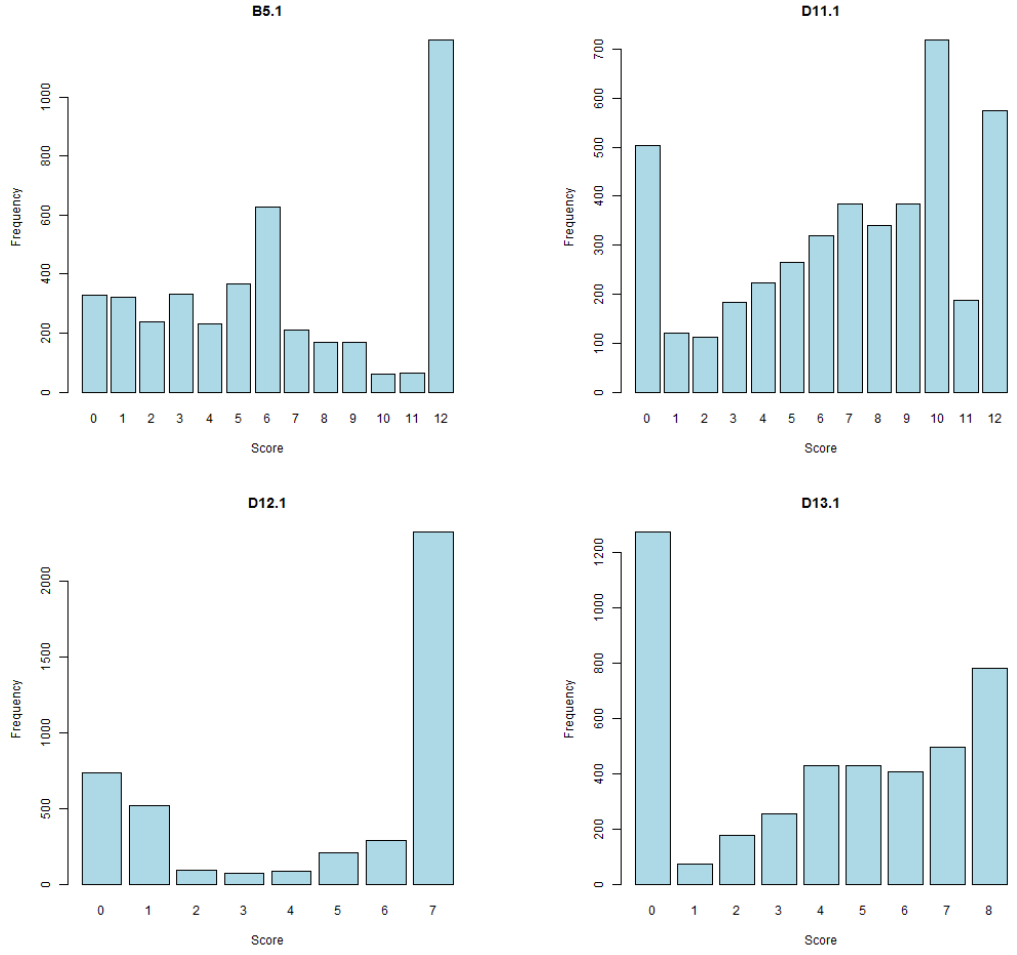


Table 5 Item means by programming-language choice

Item	Tariff	All	C#	Java	Pascal	Python	VB.NET
A1.1	4	0.36	0.35	0.31	0.65	0.32	0.39
A1.2	2	0.57	0.55	0.55	0.38	0.61	0.55
A1.3	2	0.82	0.80	0.80	0.98	0.81	0.82
A2.1	5	3.41	3.37	3.63	3.50	3.38	3.43
A2.2	1	0.08	0.08	0.09	0.13	0.08	0.07
A2.3	1	0.06	0.07	0.06	0.10	0.05	0.06
A2.4	2	0.56	0.58	0.60	0.64	0.53	0.57
A3.1	1	0.92	0.91	0.96	0.95	0.92	0.91
A3.2	1	0.33	0.31	0.27	0.35	0.33	0.36
A3.3	1	0.17	0.17	0.17	0.13	0.18	0.16
A3.4	1	0.15	0.20	0.11	0.25	0.14	0.14
A3.5	2	0.48	0.44	0.5	0.57	0.46	0.55
A3.6	1	0.38	0.38	0.35	0.49	0.38	0.38
A4.1	1	0.62	0.60	0.68	0.70	0.62	0.60
A4.2	1	0.65	0.62	0.67	0.76	0.65	0.63
A4.3	3	1.98	1.95	2.00	1.99	1.99	1.96
B5.1	12	6.57	6.54	6.13	7.12	6.74	6.24
B5.2	1	0.51	0.52	0.49	0.42	0.51	0.51
C6	2	0.79	0.90	0.90	0.83	0.76	0.71
C7.1	1	0.49	0.57	0.49	0.75	0.45	0.48
C7.2	1	0.43	0.44	0.43	0.51	0.43	0.42
C7.3	1	0.68	0.69	0.71	0.80	0.67	0.66
C7.4	2	0.40	0.48	0.26	0.55	0.37	0.43
C7.5	4	0.98	1.05	0.81	1.45	0.96	0.93
C7.6	1	0.26	0.24	0.27	0.40	0.25	0.26
C8.1	3	0.91	0.95	0.96	0.93	0.90	0.86
C8.2	1	0.13	0.11	0.09	0.20	0.14	0.13
C8.3	1	0.11	0.12	0.05	0.13	0.10	0.12
C8.4	1	0.23	0.22	0.17	0.34	0.24	0.23
C9.1	1	0.81	0.79	0.8	0.89	0.84	0.75
C9.2	1	0.81	0.79	0.83	0.88	0.82	0.78
C9.3	1	0.58	0.57	0.70	0.58	0.59	0.51
C9.4	1	0.87	0.80	0.90	0.95	0.87	0.90
C9.5	1	0.47	0.50	0.44	0.70	0.45	0.43
D10.1	4	3.26	3.19	3.14	3.31	3.36	3.14
D10.2	1	0.77	0.75	0.73	0.74	0.80	0.75
D11.1	7	4.70	4.74	4.18	4.67	4.78	4.66
D11.2	1	0.64	0.65	0.58	0.63	0.64	0.64
D12.1	12	6.96	7.24	6.67	6.84	6.89	7.02
D12.2	1	0.44	0.49	0.4	0.44	0.43	0.44
D13.1	8	3.98	3.96	3.67	4.12	4.07	3.87
D13.2	1	0.22	0.23	0.21	0.24	0.22	0.23

3.2 Partial Credit Model fit

The PCM was fitted to the item data from Paper 1 and evaluated. Correlations in the model residuals between several consecutive items (Q3>0.3) suggested the existence of local dependence. Examining these items showed that many were logically linked, e.g. in several of these pairings the second item asked the student to provide a screen capture that demonstrated the outcomes from the previous item's code (Appendix C gives an example). In other pairings, items referred to closely related concepts, or to the same resource given in the paper (examples are given in Appendix C). These consecutive pairs of items were combined, leaving 33 items. The PCM was then refitted and assessed for fit adequacy. The model showed good reliability at 0.915.

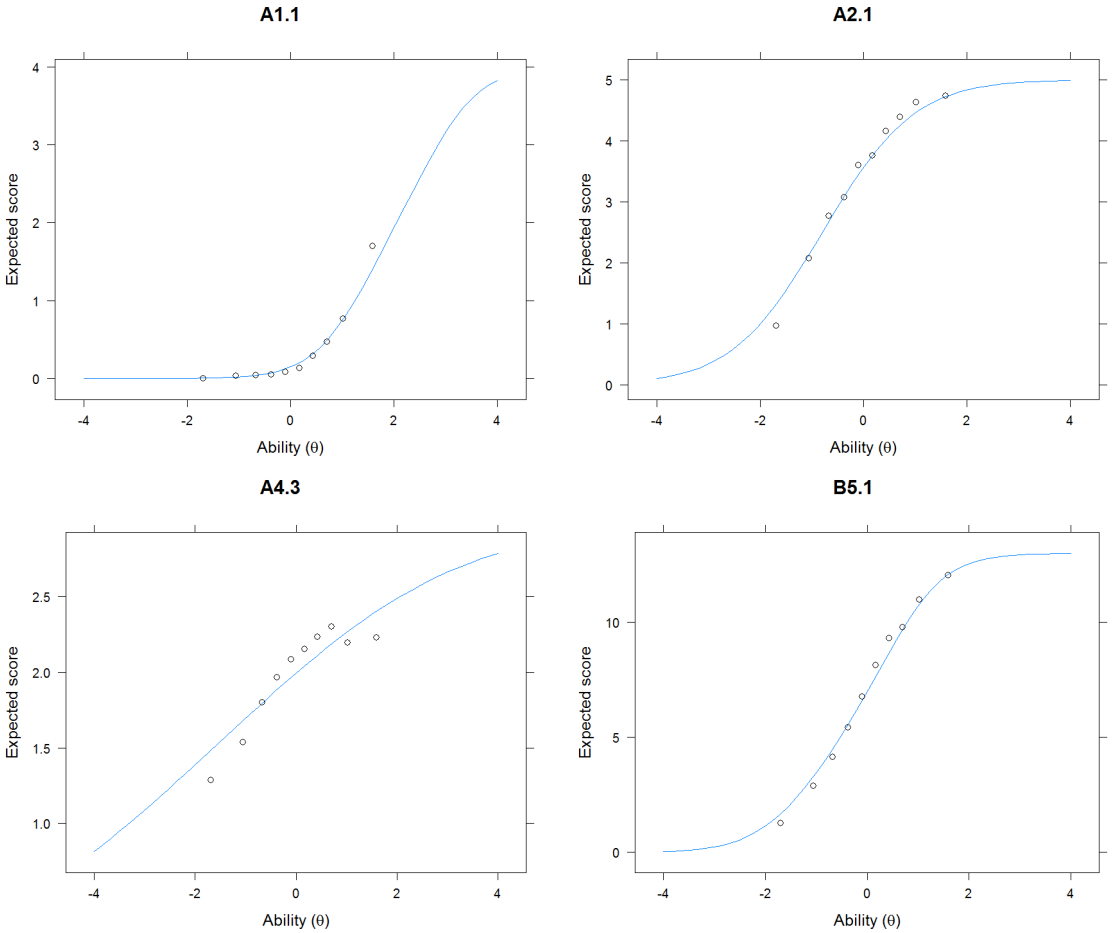
Table 6 shows the infit and outfit statistics from the PCM. Three items were flagged for misfit (infit>1.2): A2.1, A4.3 and the B5.1+B5.2 combination, but for each infit≤1.30, and so acceptable. ICCs for these items are shown in Figure 6, together with item A1.1, which showed poor outfit (<0.8). Visual inspection shows that the fit for these items appears adequate as the empirical observations do not deviate far from the model ICC. The high infit for item B5.1+B5.2 may be caused by the response pattern to B5.1, which showed multiple peaks (see Figure 5). Item A4.3 is a relatively easy item, where most students scored full or near full marks, it appears that the most and least able candidates do not always perform as well as predicted by the model, suggesting that this item did not discriminate between students effectively. Despite the high level of zeros, the final item (D13.1+D13.2) was not flagged as misfitting.

Table 6 PCM item location parameter estimates, standard errors and Outfit/Infit statistics

Item	Tariff	Location	s.e.	Outfit	Infit
A1.1	4	1.51	0.021	0.63	0.81
A1.2	2	0.91	0.024	1.02	1.01
A1.3	2	0.26	0.020	0.85	0.89
A2.1	5	-0.48	0.011	1.40	1.23
A2.2	1	2.59	0.057	0.68	0.91
A2.3	1	2.97	0.067	0.81	0.96
A2.4	2	0.91	0.024	0.86	0.89
A3.1	1	-2.62	0.056	1.00	1.00
A3.2+A3.3	2	0.94	0.023	1.13	1.11
A3.4	1	1.80	0.043	0.98	0.99
A3.5	2	1.17	0.026	0.85	0.89
A3.6	1	0.48	0.033	0.82	0.85
A4.1+A4.2	2	-0.46	0.019	0.98	0.95
A4.3	3	-0.64	0.017	1.48	1.26
B5.1+B5.2	13	-0.21	0.006	1.60	1.29
C6	2	0.50	0.025	1.05	1.05
C7.1	1	-0.01	0.032	0.93	0.93
C7.2	1	0.25	0.032	0.79	0.82
C7.3	1	-0.86	0.034	0.79	0.85
C7.4	2	1.17	0.025	0.84	0.92
C7.5	4	0.97	0.016	0.85	0.90
C7.6	1	1.11	0.036	0.94	0.97
C8.1+C8.2	4	0.70	0.015	1.01	1.01
C8.3	1	2.25	0.050	0.79	0.93
C8.4	1	1.25	0.037	0.82	0.90
C9.1+D9.2	2	-1.12	0.023	1.05	1.00
C9.3	1	-0.38	0.032	0.86	0.88
C9.4	1	-2.12	0.047	0.93	0.97
C9.5	1	0.11	0.032	0.99	0.99
D10.1+D10.2	5	-0.77	0.012	0.91	0.94
D11.1+D11.2	8	-0.42	0.008	1.08	0.92
D12.1+D12.2	13	-0.16	0.007	1.18	1.15
D13.1+D13.2	9	0.07	0.008	1.14	1.09

+ indicates combined items

Figure 6 Plots of items showing misfit in terms of outfit and infit from PCM



Two remaining pairs of items were flagged by the Q3-statistic as still indicating some local dependence (Q3>0.2), but the absolute between-item residual correlation was less than 0.3. These items were not consecutive and do not appear to be related in content, and so no further modification was made to the items.

About 6.9% of students were flagged as having aberrant response patterns, whereby the majority of these were underfitted the model (infit-t>2) with only 30 students showing responses that were too predictable (infit<−2). While this proportion of misfitting students is higher than ideal, all students were retained in the data in order to preserve the generalisability of the results.

PCA of residuals indicated evidence of a second dimension in the assessment data. The first eigenvalue was 2.36 and explained 7% of the total variation. A scree plot of the first 10 principal components is shown in Figure 7 and Table 7 shows the item weightings for the first principal component, where the larger weightings are highlighted. There appears to be a contrast between many of the AO1/AO2 questions of sections A and C (knowledge and analysis) and the AO3 questions of sections B and D (program, design and evaluate). For the second principal component and onward, no substantively interpretable pattern could be identified.

Figure 7 Scree plot of eigenvalues following PCA analysis of PCM residuals

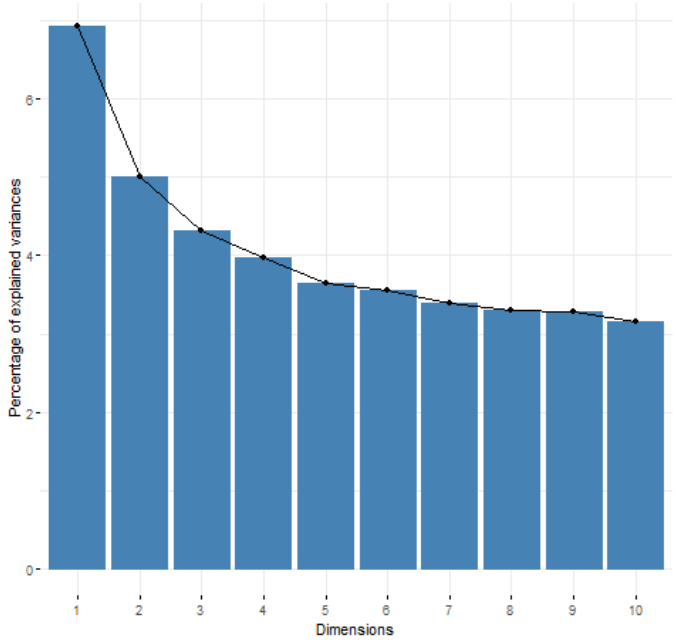


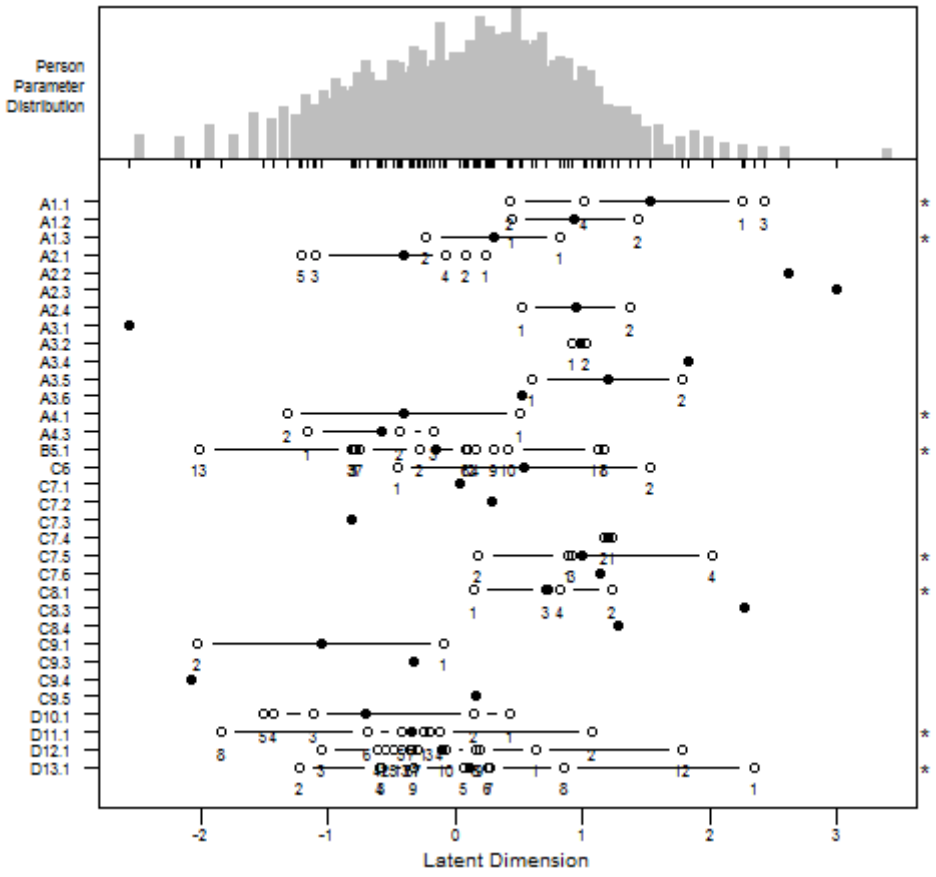
Table 7 Item weightings for first principal component after PCA of PCM residuals

Item	Assessment Objective	PC1
A1.1	AO2	-0.072
A1.2	AO1	-0.137
A1.3	AO1	-0.368
A2.1	AO2	-0.088
A2.2	AO2	-0.043
A2.3	AO2	-0.022
A2.4	AO1	-0.327
A3.1	AO1	0.027
A3.2+A3.3	AO1	-0.214
A3.4	AO2	-0.038
A3.5	AO1	-0.333
A3.6	AO1	-0.216
A4.1+A4.2	AO1	-0.290
A4.3	AO2	-0.088
B5.1	AO3	0.266
C6	AO1	0.018
C7.1	AO2	-0.093
C7.2	AO2	-0.091
C7.3	AO2	-0.190
C7.4	AO3	0.003
C7.5	AO2	-0.270
C7.6	AO2	-0.036
C8.1+C8.2	AO2	-0.009
C8.3	AO2	-0.054
C8.4	AO1	-0.260
C9.1+C9.2	AO1/2	-0.005
C9.3	AO2	-0.102
C9.4	AO1	-0.060
C9.5	AO1	-0.123
D10.1+D10.2	AO3	0.139
D11.1+D11.2	AO3	0.181
D12.1+D12.2	AO3	0.151
D13.1+D13.2	AO3	0.232

weighting>0.1
weighting<-0.1

Figure 8 shows the person-item map for the PCM estimates, which illustrates the student ability distribution and the location of the item difficulties on the ability scale. Most student abilities lie between -2 and 2 on the latent scale. The map shows that most items targeted this ability range, other than item A3.1 which was relatively easy (β<-2), and item A2.3 which was relatively difficult (β=3).

Figure 8 Person-item map for PCM estimates for Paper 1



Overall, the PCM fit appears reasonable but there is some evidence of minor item misfit, and of a small second dimension in the residual variance. As model misfit might be caused by DIF, each option was evaluated separately using a PCM model and this analysis is summarised in Table 8. The reliability was good for each model and no single option showed a disproportionate number of misfitting person estimates for the candidates. Some items show misfit in several language options, for example, item A2.1 is flagged as misfitting in every option model (detailed infit and outfit statistics by option are given in Table D1, Appendix D). Therefore, it appears that model misfit in the overall dataset is unlikely to be due to DTF.

Table 8 Summary of PCM model fit evaluation by option

	C#	Java	Pascal	Python	VB.NET
n	778	300	205	2087	948
Reliability	0.912	0.911	0.925	0.912	0.922
Number of flagged item misfits	4	3	7	4	3
Flagged items	A1.1	A2.1	A1.3	A1.1	A2.1
	A2.1	A3.2	A2.1	A2.1	A4.3
	A4.3	D12.1	A3.2	A4.3	B5.1
	B5.1		A4.1	B5.1	
			A4.3		
			B5.1		
			C7.2		
|residual correlation|>0.3	0	1	2	0	0
Flagged candidates	8.2%	7.0%	5.9%	5.9%	6.9%

3.3 Multiple-group model

A DIF analysis across the five options suggested few items in section A could be considered invariant across the language options (Appendix D, Table D2 gives further details). A multiple-group model using a small anchor made up of invariant items was fitted. The anchor consisted of 3 items (4 marks): A1.3, A2.2 and A3.6. This model was not significantly different from the independent model (χ^2=13.7, d.f.=12, p=0.32).This a small anchor, which risks the construct not being adequately represented, however using a larger anchor of five items defined with a cut-off of p(DIF)>0.05 led to a model that was significantly different from the independent model (χ^2=57.4, d.f.=32, p=0.004). Table 9 shows the model statistics for the unconstrained independent model and the three alternative equated models: the three-item and five-item anchor models, and one using all of section A as the anchor. Both the AIC and the χ^2-test favour the 3-item anchor model, whereas BIC favours the section A anchor model. As both the AIC and the χ^2-test supported the 3-item anchor model, it was chosen as the preferred model for further analysis.

Table 9 Multiple-group model fit statistics

	AIC	BIC	Log-like	χ2	d.f.	p
Independent model	241622.6	244839.7	-120306.3	-	-	-
Equated models:						
Section A anchor	241672.8	244167.4	-120443.4	274.2	112	<0.001
5 item anchor	241616.0	244629.3	-120335.0	57.4	32	0.004
3 item anchor	241612.2	244752.9	-120313.1	13.7	12	0.32

The estimated means and variances of the latent trait from the 3-item anchor model are shown in Table 10, the students studying the Pascal option appeared to have a higher average ability than the other students by 0.35 logits, and were more varied in their ability. However, the 95% confidence limits for the variance parameter estimates overlap suggesting that the differences are not statistically significant. The other option groups of students appeared similar in ability, although the VB.NET candidates were more varied.

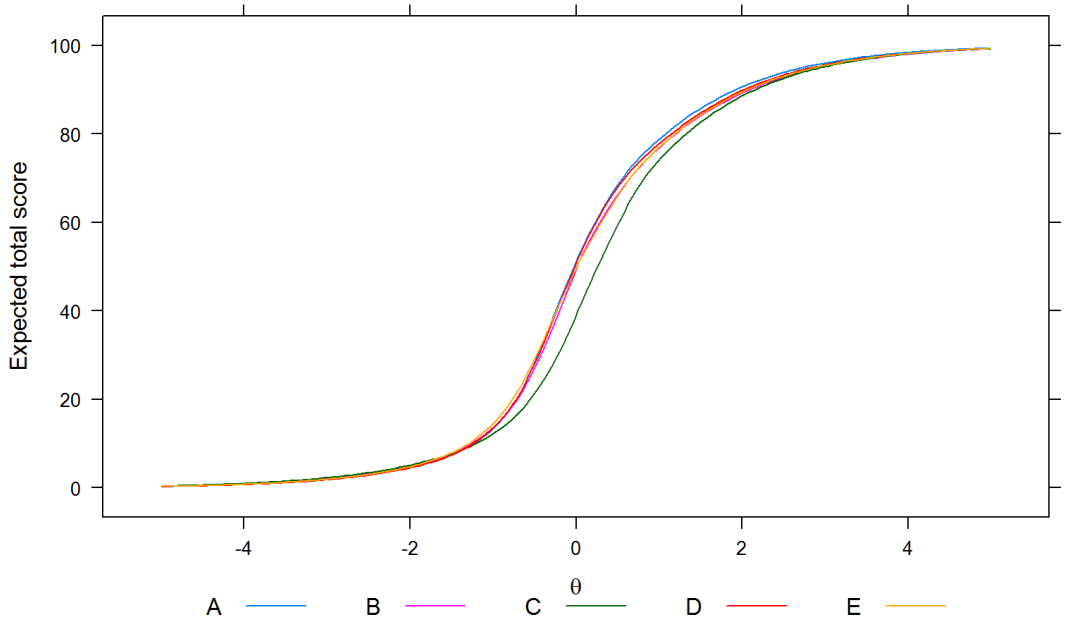
Table 10 Latent ability distribution mean and variance estimates by option from the equated multiple-group model, with 95% confidence limits

Option	Mean (95% CI)		Variance (95% CI)	
C#	0.00	-,-	0.37	(0.32, 0.41)
Java	-0.01	(-0.18, 0.15)	0.35	(0.28, 0.43)
Pascal	0.35	(0.16, 0.55)	0.50	(0.38, 0.62)
Python	0.01	(-0.09, 0.11)	0.38	(0.35, 0.41)
VB.NET	0.00	(-0.12, 0.12)	0.50	(0.44, 0.55)

Figure 9 shows the TCCs for the five options from the equated model. This suggests that, after accounting for ability, there were differences between the performances of the students from the Pascal option and those from the four other options.

Figure 9 TCCs for the five programming options

(A: C#; B: Java; C: Pascal; D: Python; E: VB.NET)



Items from the programming section were each assessed for DIF using the equated model (see Table 11). Several high tariff items appear to be performing differently across the five routes as they showed both significant DIF and BIC>10: items B5.1, C6, C7.4 and the whole of section D (Figure D3 in Appendix D shows ICCs for these items). The size and significance of the differences between the TCCs can only be assessed pairwise, the results of this analysis are described in the next section.

Table 11 Assessment of DIF in programming items from all-groups equated model

Item	ΔAIC	ΔBIC	χ^2	d.f.	p	Adjusted p
B5.1	-137.87	193.40	241.87	52	<.001	<.001
C6	-31.32	19.64	47.32	8	<.001	<.001
C7.1	-47.03	-21.55	55.03	4	<.001	<.001
C7.2	6.92	32.41	1.08	4	0.898	1.00
C7.3	1.25	26.73	6.75	4	0.150	1.00
C7.4	-29.73	21.23	45.73	8	<.001	<.001
C7.5	5.98	107.91	26.02	16	0.054	1.00
C7.6	1.03	26.52	6.97	4	0.138	1.00
C8.1	6.93	108.86	25.07	16	0.069	1.00
C8.3	-7.00	18.48	15.00	4	0.005	0.089
C8.4	0.42	25.91	7.58	4	0.108	1.00
C9.1	-35.28	15.68	51.28	8	<.001	<.001
C9.3	-29.87	-4.39	37.87	4	<.001	<.001
C9.4	-37.05	-11.57	45.05	4	<.001	<.001
C9.5	-22.47	3.01	30.47	4	<.001	<.001
D10.1	-63.53	63.89	103.53	20	<.001	<.001
D11.1	-28.72	175.14	92.72	32	<.001	<.001
D12.1	-0.74	330.53	104.74	52	<.001	<.001
D13.1	-16.81	212.53	88.81	36	<.001	<.001

Positive values of ΔAIC and ΔBIC suggest that the less constrained model fits better
Adjusted p-value: Bonferroni multiple-comparison correction applied

3.4 Group-wise differential test functioning comparisons

DTF analysis was considered pairwise between the programming-language options. To reduce the number of comparisons made and to control for the overall type I error rate, only five comparisons were investigated in detail, with a Bonferroni correction made (so each was tested at the 1% level and 99% confidence limits given, in order to maintain an overall 5% type I error). The first comparison is explained in some detail, while the other comparisons are provided in a more abridged form.

3.4.1 Python v VB.NET DTF/DBF evaluation

An eight-item anchor from section A was selected (11% of the total marks) excluding items: A1.1, A1.2, A2.1, A2.4, A3.2 and A3.5. This anchor gives a model that was not significantly different from the independent model (χ^2=9.13, d.f.=11, p=0.61). A model using all of section A as the anchor differed significantly from the independent model (χ^2=64.1, d.f.=28, p<0.001). Again, AIC favoured an invariant-anchor model, but BIC favoured the all-section-A-anchor model (see Table D4,

Appendix D).

The DTF analysis is summarised in Table 12. The DTF statistics suggest that there was not a significant difference between the options, either over the whole paper (sDTF=0.008, p=0.97,) or in the programming section (sDBF=0.536, p=0.41). The programming section (sDBF) shows a greater difference between options than overall (sDTF), and this is also reflected in the absolute differences (uDBF=1.169, compared to uDTF=0.359). This suggests that between-option differences in non-anchor items in section A reduce the impact of differences in the programming section on overall performance. However, neither the sDTF or the sDBF were significantly different from zero and so this may only be random variation.

Table 12 DTF statistics comparing Python and VB.NET

Whole test statistics		Programming section statistics	
sDTF	0.008	sDBF	0.536
p(sDTF=0)	0.97	p(sDBF=0)	0.41
uDTF	0.359	uDBF	1.169
uDTF 99% CI	(0.22, 0.60)	uDBF 99% CI	(0.78, 2.26)

The uDTF statistic (0.359) is higher than the sDTF (0.008), this indicates that any differences between the options changes direction across the ability range (i.e. the TCCs cross). This can be seen in Figure 10, which shows the detailed $sDTF_\Theta$ estimates with their 99% confidence limits across the ability range, the $sDTF_\Theta$ is above zero at higher ability values, and negative at lower ability values. However, at almost every value of θ, the horizontal red line (which indicates $sDTF_\Theta=0$) is within the shaded confidence interval. Table 13 gives the confidence limits for $sDTF_\Theta$ at the θ-values that correspond to the grade boundaries for the key grades A and B on the two TCCs. The confidence intervals include zero, showing that for these two options there is no evidence of a significant difference in the likelihood of achieving these key grades.

Figure 10 $sDTF_\Theta$ for Python and VB.NET comparison with 99% confidence limits

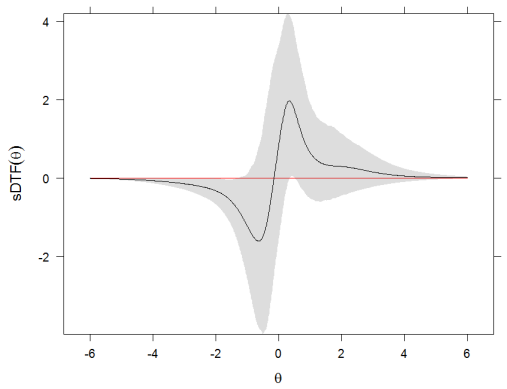


Table 13 $sDTF_\Theta$ at key grade boundaries for the Python and VB.NET comparison

	A (68 marks)				B (54 marks)			
	θ	$sDTF_\Theta$	99% CI		θ	$sDTF_\Theta$	99% CI	
Python	0.46	1.83	(-0.00,	3.69)	0.07	1.21	(-1.05,	3.39)
VB.NET	0.53	1.66	(-0.06,	3.42)	0.10	1.35	(-0.88,	3.47)

3.4.2 Other comparisons

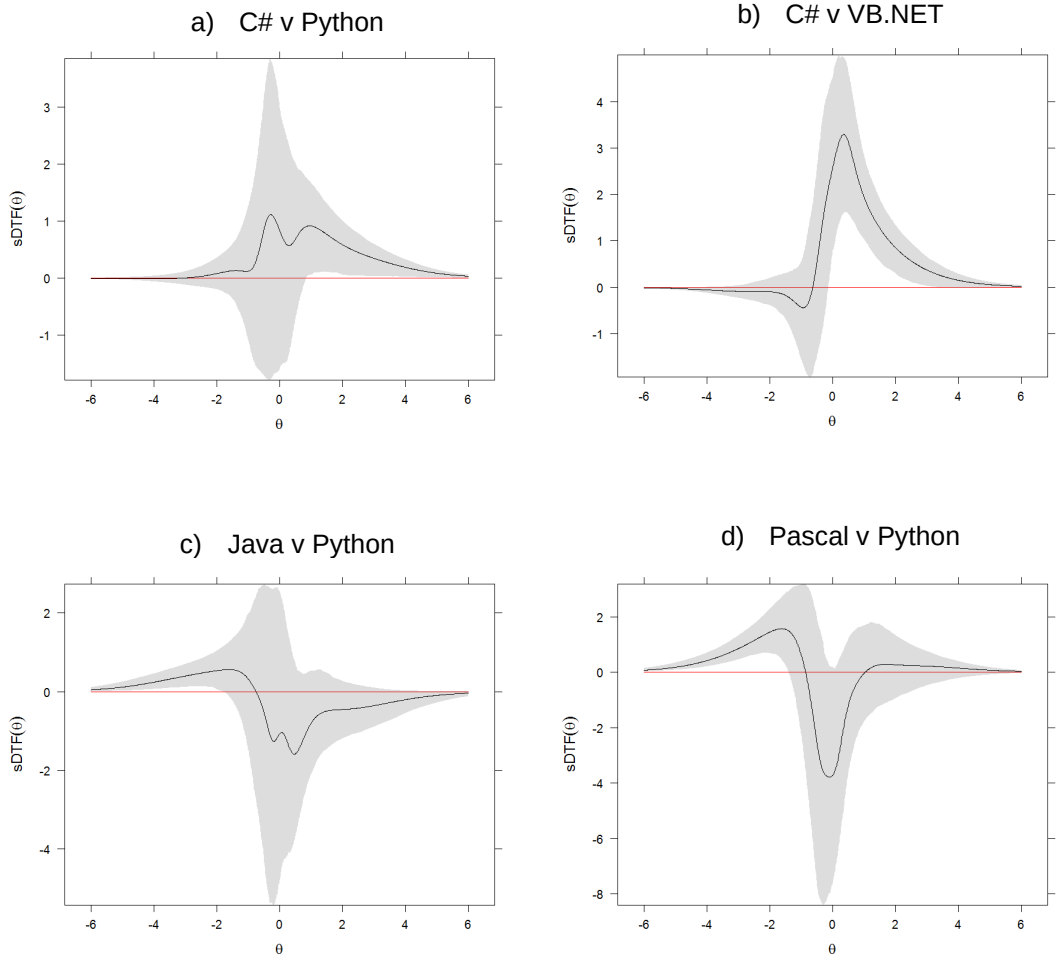
Table 14 and Figure 11 show the results for the four other comparisons that were selected for detailed consideration. Again, for each comparison an anchor was chosen from a subset of section A, as not all section A items were found to be invariant (Table D4 lists the selected items, Table D5 gives model comparison statistics – see Appendix D). For the comparison of Python and Pascal, only five items were found invariant. Over each comparison the average difference in the programming section scores was greater than seen in the overall scores (i.e. sDBF>sDTF). However, for most comparisons, there was no evidence of a significant difference between the programming options. The exception was the C# and VB.NET comparison where there was a significant difference between these options, overall and at the key grade boundaries of A and B. In the Pascal and Python comparison, no overall DTF was found but there does appear to be evidence of $sDTF_\Theta$ at lower abilities ($\theta < -1.5$, see Figure 11(d)).

Table 14 DTF statistics for other comparisons

Comparison	whole test statistics				programming section statistics			
	sDTF	sDTF[1] p-value	uDTF	uDTF 99% CI	sDBF	sDBF[2] p-value	uDBF	uDBF 99% CI
C# & Python	0.272	0.075	0.274	(0.11, 0.66)	0.682	0.26	0.682	(0.16, 2.25)
C# & VB.NET	0.447	0.003	0.539	(0.32, 0.83)	1.869	<.001	1.955	(1.26, 2.92)
Java & Python	-0.138	0.57	0.405	(0.21, 0.91)	-1.442	0.12	1.449	(0.25, 3.20)
Pascal & Python	0.057	0.86	0.676	(0.49, 1.17)	-1.966	0.071	2.153	(0.66, 4.74)

[1]Test of p(sDTF=0)
[2]Test of p(sDBF=0)

Figure 11 $sDTF_\theta$ for pairwise option comparisons with 99% confidence limits



3.4.3 C# v VB.NET DTF/DBF evaluation

In this comparison an overall sDTF of 0.45 marks was found, with 99% confident interval (0.07, 0.80). The uDTF was 0.539, indicating that the TCCs cross and the bias doesn't favour C# across the whole ability range (see Figure 11(b)). The programming section statistic was larger than the overall statistic: sDBF=1.87 marks ($p<0.001$), indicating that some of the differences seen in the programming section were cancelled out by differences seen in section A. In Figure 11(b), the $sDTF_\Theta$ plot indicates that there may have been an issue around the grade boundaries of A and B. Extra details of the $sDTF_\Theta$ values at grades A and B are given in Table 15, suggesting that at mid to high abilities, the paper was biased in favour of C# students over VB.NET students by approximately three marks. The significant DIF items from the programming section for this comparison are shown in Table 15, and those where ΔBIC>10 are illustrated in Figure 12, which shows that many of these consistently favour the C# option. Inspection of the ICCs also shows that the DIF is complex and not always uniform (i.e. consistently favouring one route in size and direction across the ability range). In two items (D10.1 and D12.1) the ICCs cross indicating that the DIF changes direction, favouring C# for high ability students and VB.NET for lower ability students. The main contributors to DTF were items: B5.1, C7.5 and D12.1. For the high tariff items B5.1 and D12.1, the bias towards the C# route was particularly noticeable in the range: $0 < \theta < 2$.

Table 15 $sDTF_\Theta$ at key grade boundaries for the C# and VB.NET comparison

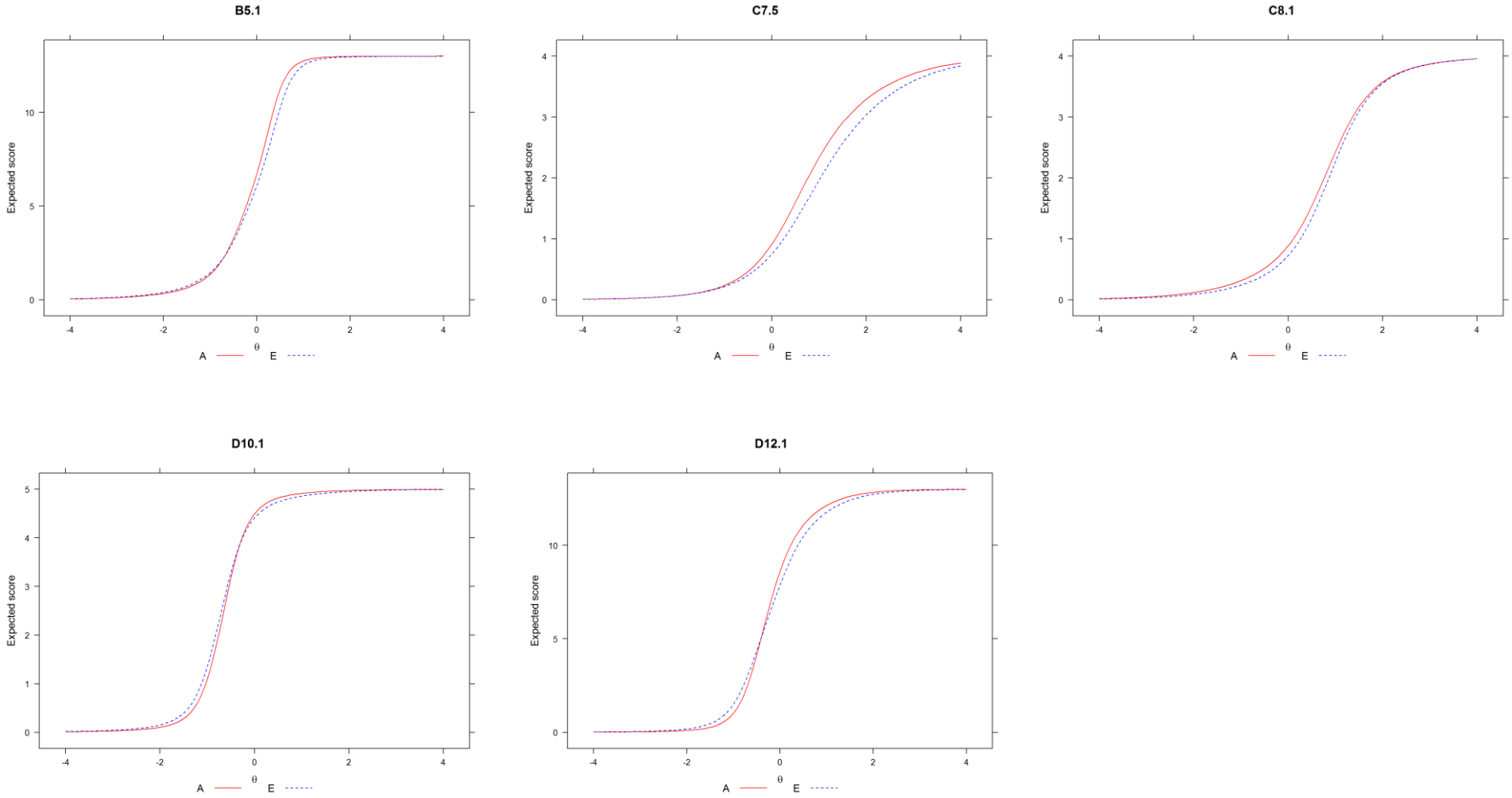
	A (68 marks)				B (54 marks)			
	Θ	$sDTF_\Theta$	99% CI		Θ	$sDTF_\Theta$	99% CI	
C#	0.46	3.20	(1.64,	4.69)	0.07	2.79	(0.97,	4.61)
VB.NET	0.57	2.96	(1.56,	4.33)	0.14	2.98	(1.18,	4.75)

Table 16 Significant DIF items in the C# v VB.NET comparison

Item	ΔAIC	ΔBIC	χ^2	d.f.	p	Adjusted p
B5.1	-10.5	60.4	36.5	13	0.001	0.010
C6	-38.1	-27.2	42.1	2	<.001	<.001
C7.1	-15.2	-9.7	17.2	1	0.000	0.001
C7.5	-10.5	11.3	18.5	4	0.001	0.018
C8.1	-2.9	18.9	10.9	4	0.027	0.52
C9.3	-6.4	-1.0	8.4	1	0.004	0.070
C9.4	-33.1	-27.7	35.1	1	<.001	<.001
C9.5	-9.4	-4.0	11.4	1	0.001	0.014
D10.1	-12.6	14.6	22.6	5	0.000	0.008
D12.1	-15.7	55.2	41.7	13	<.001	<.001

Positive values of ΔAIC and ΔBIC suggest that the less constrained model fits better
Adjusted p-value: Bonferroni multiple-comparison correction applied

Figure 12 Items showing significant DIF in the programming section of Paper 1 between options C# (A) and VB.NET (E)



3.4.4 Additional comparisons

Plots showing $sDTF_{\Theta}$ for the five comparisons not considered in detail here are shown in Figure D6 (Appendix D), which were each also considered with 99% confidence limits. There is some evidence of a significant overall difference between Pascal and Java sDTF=1.10, p=0.002, however the associated sDBF for the programming section was not significant (sDBF=1.98, p=0.10) and so does not appear to be a programming-language related issue. Each of the additional comparisons showed no evidence of significant $sDTF_{\Theta}$ for mid to high ability students. There was evidence of $sDTF_{\Theta}$ at low abilities in comparisons involving Pascal students (with the Pascal students performing better on average).

4 Discussion

This dissertation has evaluated whether offering A-level Computer Science with a choice of programming languages in its teaching and assessment has led to a fair qualification. Specifically, it has evaluated whether a paper that allows student responses to be given in one of five programming languages functioned in an invariant manner for each of the resulting sub-groups of students in terms of their overall test scores. This is important because both students and schools are judged on the outcomes of the assessment. Students may potentially be disadvantaged in university applications if the paper is biased against the language they used, and schools might underperform in one of the performance measures that feature in school league tables. Therefore, it is necessary to carefully evaluate whether offering language choice, which was done in order to enable greater access to the subject, has resulted in unfairness to students.

Previous research on optional questions indicated that offering choice in an assessment may lead to inequity. Here, only five possible routes were available making it more straightforward to use DTF methods that are normally applied to broader issues of fairness where a group of students might be disadvantaged on a paper due to a common background characteristic. The possible presence and extent of test bias has been evaluated statistically using a Rasch/IRT modelling approach to estimate student abilities. This has meant that the overall functioning of the paper could be looked at for evidence of any systematic bias favouring or disadvantaging any programming language group of students, once ability had been accounted for. In addition, outcomes were compared at certain abilities, and the performance of a subset of questions was assessed. The following sections outline this study's results and discuss their implications for language choice in A-level computer science. The strengths and weaknesses of the Rasch/IRT analysis are discussed along with the more general use of choice in assessment. An appeal for more prevalent use of DTF analysis in A-level awards is made, in the interests of better evaluating broader issues around fairness in assessment. Finally, future research ideas are discussed.

4.1 Summary of findings

4.1.1 RQ1: Can a Rasch or IRT model be successfully applied to the computer science test data?

The PCM showed a reasonable fit to the data. The items appeared to target

the ability range of the students effectively and there was no evidence that a more complex model was required. There was some indication of a second dimension in the data – which appeared to be linked to the various assessment objectives targeted by the exam indicating that, on average, students do not give similar performances on each of the assessment objectives; where students may have been differentially weaker for some objectives and stronger in another. However, it was concluded that as a dominant dimension was present in the data a Rasch/IRT approach to estimating student ability was appropriate for further analysis.

In terms of raw scores, Java students appeared to be the weakest, and Pascal the strongest, however it is also possible that Java students were disadvantaged on the test and Pascal students advantaged. Both routes were relatively unpopular with teachers and so these results could reflect the strengths and weaknesses of a small number of teachers, rather than problems with programming language choice. A multiple-group IRT model was fitted to the data, which enabled a DTF evaluation that accounted for any differences in the ability of each language-group of students. This model found that the Pascal students were more able on average than the other students by about 0.35 logits. Students from the other routes appeared similar in ability. Visual inspection of the resulting TCCs showed that the curves for C#, Java, Python and VB.NET students were similar, indicating that their performances on the test were broadly comparable across the whole ability range. This suggests that there had been no disadvantage for the Java students. However, the curve for Pascal students differed considerably from the others. The Pascal students appeared to underperform on the paper relative to all the other groups, i.e. Pascal students achieved lower total scores than those using any of the other languages across most ability levels, except at low- or high-test scores. This may indicate a cause for concern for test developers, but it is also possible that the multiple-group model has not adequately described these students.

4.1.2 RQ2: To what extent is differential test functioning (DTF) impacting the different programming-language choices on average?

The statistical significance of any overall test bias (DTF) was assessed pairwise for a subset of pairings. The Python route was taken by nearly half of the students and all DTF comparisons with the Python route were non-significant, indicating that the use of programming language choice did not lead to five sets of equally able students performing very differently in their final marks. In addition, assessment of bias in the programming-language dependent section (DBF) showed

that comparisons between Python and all other routes were non-significant. This suggests that students from each alternative route performed similarly in the examination to the most popular route in the programming-language dependent section. The analyses presented here showed a small statistically significant difference in student achievements for Paper 1 between two of the major non-Python language choices (C# and VB.NET), whereas other pairwise comparisons did not show evidence of significant DTF. Overall, the paper appears to be performing in a fair manner for all programming language options.

When considering the programming-language dependent sections there was always a larger DBF effect than whole paper effect (DTF), indicating an interaction between student performances on section A and the rest of the paper. This suggests several possibilities, including that students might be strong in one aspect of computer science (section A is targeted the assessment of knowledge and analysis; sections B to D concerned writing, designing and evaluating programming code), or that their teachers also have strengths and weaknesses, and student performance reflects this, or finally that some students did not give each section the effort it was due, in particular because they were moving quickly to the questions that were based on the pre-released material (sections C and D) – a concern that has been raised by examiners[15]. This feature could be linked to evidence that indicated the presence of a minor second dimension in the data, but the differences are not large enough to suggest that two marks should be reported per paper. However, it does appear that the presence of DIF items in section A counteracted any DBF seen between the language groups across sections B to D. Thus, resulting in final scores that appeared comparable by language choice with little or no evidence of DTF. As the responses to section A items were independent of programming language choice, this may indicate that differing teaching and learning experiences for the students of each route impacted exam performance more profoundly than language choice.

4.1.3 RQ3: Does the DTF vary with ability? If so, which ability levels are most effected and is this impacting outcomes at the key grades of A and B

Each multiple-group IRT model used to assess a pairwise comparison also allowed detailed inspection of the differences in performance at various levels of ability ($sDTF_\Theta$). For most comparisons considered, $sDTF_\Theta$ rarely differed significantly

[15] Noted in correspondence with examiners.

from zero for the abilities targeted by the paper. Implying that for the range of abilities where most students fell, the test performed in a fair manner with no evidence of DTF.

For the significant DTF comparison between the languages C# and VB.NET, a closer inspection showed that mid to high ability students ($0 < \theta < 2$) using C# performed better than VB.NET students by 1 to 3 marks, suggesting that use of C# might have advantaged students at this ability level. This means that for those students near the key paper grade boundaries of A or B, C# students were more likely to achieve the higher grade than VB.NET students. However, whilst this would be an important difference for the students concerned, it does appear that the paper performed fairly at most ability levels. For very high ability students ($\theta > 2$), DTF_{Θ} fell to zero, suggesting that for those students likely to achieve top marks and an A* grade on the paper, language choice was irrelevant. The routes were also comparable at lower abilities ($\theta < 0$).

4.1.4 RQ4: Is the DTF due to differential item functioning (DIF) on specific questions or is it accrued over several questions that individually do not show DIF?

In the significant comparison between the C# and VB.NET language routes, it appeared that the DTF accrued over a few items that showed significant DIF favouring C# students. For some items, the DIF was complex and inconsistent across the ability range and the ICCs crossed. This complex pattern of DIF may make identifying item-level unfairness difficult but for those items where the DIF always favoured the C# students, examiners should consider whether there was something about creating code using C#, rather than VB.NET, that made these items more straightforward for C# students. However, as there were also are items in section A that showed DIF, it cannot be ruled out that the overall DTF had been caused by differences in classroom experience. Suggestions for further research to explore this are discussed below.

4.2 Implications for A-level Computer Science

Overall, offering programming language choice has not resulted in large levels of DTF or DBF between the routes. Where DIF items have been identified, they have not caused overall test bias, except in one comparison. In any post-hoc review of paper performance, item writers and the exam preparation team should consider DTF and, where it exists, consider any items where statistically significant DIF was

seen that consistently favoured one group over another. Camilli (2013) cautions against being over-reliant on quantitative evidence and it should not be immediately assumed that a real problem has occurred that is definitely caused by the programming language choices. For example, it is possible that the observed difference in performance might be indicative of differing classroom experiences. Therefore, qualitative evidence is also important in confirming the source of the problem, e.g. complaints from teachers claiming a question was more difficult to answer in one language could be considered.

As suggested in section1.5, it seems unlikely that the use of optionality will always lead to equivalent assessment. In other studies looking at option choice, differences were also found. Bell (1997) found a one mark difference between English literature essays; Bramley and Crisp (2017) found five marks between the hardest and easiest combinations of three questions in a Geography paper; Willmott and Hall (1975) found candidates differing in Biology questions by as much as 10-15 marks. Here a significant difference of about three marks was found between two major option choices at two key grade boundaries. Thus, there is an accumulation of evidence, including the present study, that choice cannot often be offered with any guarantee of absolute comparability in terms of students having an equal likelihood of achieving any grade even when they have equal abilities in the subject.

Nonetheless, in this one academic year group, the problem of DTF appears to be minor. More generally, the possibility of DTF between the same pair of language choices should be considered every year. If there are repeated issue with the same pair of languages, then intervention to improve the fairness of the qualification might be needed. As discussed in the section 1.5, it is possible that the existence of DTF regularly occurring between optional papers would mean that the routes should be statistically aligned. The simplest approach to alignment within the framework of A-levels would be to ask examiners to judge this when setting grade boundaries. However, based on the evidence seen here, the current use of common grade boundaries for each programming route should be retained[16]. In part this would be because it is undesirable to risk the possibility that teachers interpret differing grade boundaries as indicating that some language choices are easier than others. It seems more logical to recommend that teachers choose the language they are either

[16] An exception should be made if during the marking season, it becomes clear that both quantitative and qualitative evidence shows that a task is genuinely more complex to tackle through one programming language. As the items are not pre-tested, this might happen despite exam writers' best intentions. This disparity could, for example, be addressed through modifications to the mark scheme.

best able to teach, or that they believe best equips their students for a future that will involve computing and writing code. At the same time exam writers should regularly review paper performances with respect to DTF and DIF so that they are aware when issues arise. They can then assess whether future item-writing practice should change.

4.3 Broader implications

4.3.1 Is the Rasch/IRT approach to DTF analysis appropriate for A-levels?

A-level assessments are not designed from the point of view of a future analysis with a Rasch/IRT model because If they were, items would not be written that are so clearly correlated in their responses. Consequently, the data have not leant themselves easily to this approach and some pre-processing was required. However, the methods have enabled a detailed interrogation of the paper and item performance between routes, over all abilities and at particular levels of the ability range.

4.3.2 Appropriate assumptions for assessing the impact of choice

As discussed in section 1.5, the Rasch/IRT approach to DTF/DIF analysis assumes that students would have responded to each question whilst using another language in the way predicted by the language-specific item parameters. It may not be reasonable to assume, for example, that how well a student tackles a task in Python indicates how well they could tackle the task in C#. In the context of this assessment, this is exactly the assumption that examiners are making when offering a choice of language, i.e., they do believe that the languages are close to equivalent approaches to answering the questions that they ask of examinees. Hence, in this context, the assumption that we can predict how well a student would perform in another language (if they had a similar chance to learn it) from how they have currently performed, is probably reasonable. This assumption would be less reasonable if the questions themselves differed, such as those used in an English literature paper that use slightly different questions depending on the book choice.

4.3.3 Generic mark schemes

Although students responded to most questions from Paper 1 in their programming language choice, the paper itself was presented as a single set of questions that showed no language specific content. It was also marked to a common, generic mark scheme. It is possible that this approach explains why major

test bias has been avoided. Bramley and Crisp (2017) suggested that if optionality cannot be avoided its use should be minimised, and they advocate for the use of generic mark schemes. In his analysis of English literature questions, Bell (1997) argued that the problems associated with choice had probably been minimised by the common nature of the skills being assessed and through the use of generic mark schemes, i.e. it is the analytical process of creating a response that is important not, for example, the detail of the characters in the book. This observation may also be true of the responses to programming questions, that it is the style of the solution that matters, not the syntax of the programming language used. Although, this should be verified by analysing the results from sittings of the computer science exams in other academic years.

4.3.4 DTF methodology in evaluating fairness in assessment

The MIRT-DTF methodology used here is relatively new but could be applied to many other scenarios in the post-hoc analysis of any assessment with respect to fairness. This would not just be around the issue of optionality but broader factors such as gender, ethnicity, SES or regional variations, where test bias might unfairly disadvantage a group of students. The method's key strength lies in the fact that it looks at the relative performance of students on a test once ability has been corrected for – this cannot be achieved if raw scores alone are analysed.

DTF is not a recent concept. In 2002, while not denying the importance of DIF analysis, Stout (2002) argued that as students are judged on their overall tests scores, rather than their item scores, test fairness should be analysed at test level not item level. However, there is very little literature on evaluating DTF. Chalmers et al. (2016) and Domingue et al. (2017) demonstrate the usefulness of this methodology to compare the test performances of different sub-groups. Chalmers et al. (2016) suggested that DTF is currently an overlooked but important issue in test evaluation. Domingue et al. (2017) also challenged the assessment community to consider this matter in more detail, arguing that DTF is an important issue that it is currently neglected in educational assessment.

4.4 Limitations of findings

4.4.1 Anchor selection problems

Choosing items for the anchor was not straightforward in the present study. Anchor items were chosen exclusively from the 14 items of section A where the students' responses were not programming-language dependent but required

programming related skills and knowledge. While it would seem logical that the items of section A should all prove to be invariant, they were not, and as these items impacted students' final outcomes on the paper it was decided to exclude them from the anchor. This problem was less notable in pairwise comparisons that did not involve the Pascal students (where 8 to 11 items were found), but considerable when all five routes were compared together (only three items were found invariant) or when Pascal was compared to Python (five items were found invariant). At some point in the five sets of pairwise anchor selection processes, every section A item played a role in an anchor. It could be that the process was over stringent in its definition of invariance and that all section A items should have been selected by default to the anchor. It is also possible that a few of the items in the programming-language specific section could have usefully been incorporated into the anchor. Further analysis, varying the method of anchor selection could assess the stability of the DTF results found.

Woods (2009) recommended that an anchor should be made up of 10 to 20% of the items. However, in contrast, Domingue et al. (2017) argued that the anchor should be small in order to enable better detection of DTF and in fact used only one item. Domingue et al. (2017) demonstrated that DTF estimates reduced as the number of constrained items increased. However, although the anchor should be uncorrupted by poorly chosen items that show even minor DIF, the anchor is used to define the ability scale and to link the option routes. Thus, the use of more items may lead to better construct representation (Woods, 2009). The anchor used in the all-routes model (section 3.3) used few items, which may not represent the construct adequately. Fortunately, the pairwise comparison models were usually based on an anchor made up of a larger proportion of section A and the conclusions drawn from these models should be more reliable. Further research is needed to provide clarity on appropriate anchor selection.

4.4.2 Pascal

The performance of Pascal students appeared to differ from that of students using other languages, both in terms of mean total scores, where they scored highly on average, and in the TCC plot showing all five routes, where their performance appeared weaker than the other groups. The model appeared to show that once their higher ability was accounted for, they should have performed even better on the paper than students that took other programming language options. However, these observations are not statistically significant and do not support an interpretation of

the test being biased against these students. When compared to the most popular route (Python) there was no evidence of statistically significant DTF or DBF.

Several issues undermined the analysis where Pascal students were concerned. First, few students used the Pascal route and IRT models require a large number of observations for an effective fit. Second, a “dummy” student was needed to allow a multiple-group model to be fitted. Finally, it is possible that the small number of anchor items used when creating the multiple-group models did not adequately capture the ability of the Pascal students, resulting in a poor estimate of their latent distribution. Woods (2009) noted that anchor selection is more complex when there are large number of items showing DIF, which may be the case with Pascal students’ data.

In contrast, Java was also unpopular, but the analysis of this route was straightforward with 10 items used in the anchor and no need to employ a dummy candidate. The results for this route appear reasonable.

4.4.3 Real and artificial DIF

DIF procedures were used at two stages of the analysis. First to define an anchor and second to assess which, if any, items contributed to any observed DTF. In any DIF identification process, the large number of comparisons being made leads to the risk of finding spurious DIF. At the anchor selection stage, the invariant status of any of the items was unknown and the status of each was assessed by assuming all the other items were invariant. Andrich and Hagquist (2015; Hagquist & Andrich, 2017) suggested that when assessing DIF in any test, the presence of real DIF in some items will lead to artificial DIF being found in others. They recommended that when real DIF is identified in an item, it should then be treated as an item per group in any further DIF analysis (referred to as resolving the item). The anchor selection process used here may have been vulnerable to the effects of artificial DIF resulting in some items being spuriously excluded from the anchor. Conversely, the second stage of DIF analysis was dependent on the adequacy of the multiple-group model. Here, the DIF identification process added constraints to items that were effectively already resolved and hence the artificial DIF scenario should not have occurred, if the anchor was adequate.

4.4.4 Confounding factors

As discussed in the introduction, determining whether real bias exists is not straightforward as observed differences might be due to factors outside of the test,

e.g. opportunity to learn (Camilli, 2006; Cole & Zieky, 2001). In a test that can be pre-tested, questions can be modified or dropped if they show evidence of DIF (this is common practice in the US). However, A-level questions are not pretested, and unanticipated unfairness can occur, even in the most carefully designed tests. The models used here have not explored the possibility of centre effects. It could be that centre characteristics such as private-school status or school size, affects a schools' ability to offer students a good preparation for this assessment. Some of these factors may interact with language choice. Unfortunately, there are insufficient data on centres to investigate this as little data were available on private or international school characteristics.

4.5 Future research ideas

The use of pre-released material in the test paper may have influenced how students approached the paper. It is unclear in what order students tackled the paper or whether they followed the timing guidance. This could be explored by interviewing students or by asking them to fill in a short questionnaire at the end of their assessment. If this research revealed that students do not follow the recommended timings, or orderings, future papers could be re-structured. If the pre-released material appeared to result in an unnecessary amount of classroom time being used to prepare students for a sub-section of the exam, then the exam board could release this material closer to the exam date.

The possibility that a programming task is more difficult to answer in one language than another should be discussed by examiners when there is statistical evidence that a difference has occurred. Isolating this possibility from differences in classroom experience for students could be investigated by interviewing teachers. The presence of DIF items in section A is unexpected and the reasons for this may also be better understood after discussions with teachers.

The anchor selection process was complex and vulnerable to the impact of artificial DIF. Identifying an invariant anchor was key to linking the different programming routes and was an important part of the DTF analysis. The impact of anchor selection could be investigated using simulated test data with varying levels of DTF between groups and varying numbers of items affected by real DIF. Identifying the accuracy of various anchor selection procedures could then inform future DTF analysis.

There has been little research on the use of choice in a question paper. The views of teachers and students could be sought on this to evaluate whether they

agree with the widely held view that question choice is for students' benefit. Barrance and Elwood (2018) looked at what students thought of subject choices made for them by their teachers and schools. It clear that those interviewed felt strongly that some of the decisions made for them were unfair, limited their opportunities, and that they would like to contribute to the discussion on the choice of subjects offered in their schools. If, in some subjects, it is found that concerns around unfairness mean that teachers and students would prefer that choice were not offered, this should be considered in future assessment designs. It is also possible that students and teachers would prefer a post-hoc correction to be made to the affected results.

A statistical review of the extent of DTF in a broader range of subjects that use question or paper choice could be undertaken The various forms of offering choice could be recorded, to assess whether using a common mark scheme has been more effective in maintaining fairness than using question specific mark schemes. This analysis would give evidence of the extent of any unfairness in the current examinations where choice is offered, and whether this is linked to either the styles of optional questions offered or to their mark schemes. If systematic unfairness is revealed, the affected assessments should be redesigned, either by avoiding offering choice, or by adopting the question or mark-scheme style of any assessments where choice did not result in meaningful unfairness.

5 Conclusion

An IRT-based DTF analysis has allowed a comprehensive statistical analysis of the performance of an A-level Computer Science paper. Item-level, section-level, and overall test-level performance has been reviewed across all ability levels, with a focus on the key grade boundaries of A and B that are important to schools and students. There did not appear to be a substantial issue with comparability between the five optional programming language routes across the range of student ability levels. Some small differences between two of the main language-choices were found around key boundaries and this might, of course, have had an impact on a small number of students and their schools. Such differences have been noted in studies of other qualifications using optionality, so computer science is not different in this respect. It was not possible to confirm that this difference was solely caused by the language choice as some differences were seen in questions that were not linked to the language, indicating the possibility that classroom experience played a part in the DTF. The observed differences do not warrant any changes in how the assessment should be offered or graded in the future, although the assessment should be monitored over time, as should any assessment that offers choice. Where DTF has been observed, associated DIF items should be considered by exam writers who may be able to identify issues that can inform future item writing practice.

Offering choice has been of benefit to students and teachers in computer science, and the fairness of the award has not been undermined. It is probable that this has been achieved through careful test design, with the strict use of a common paper and a common mark scheme. The literature indicates that offering question and paper choice does not always result in comparable assessments and therefore choice should be restricted in its use. Further research looking at the design of papers with optionality, and their associated mark schemes, may clarify when offering choice can be done fairly.

In contrast to item-level bias, test bias is currently a neglected area of test development and post-hoc review. As test bias has more impact on students than item bias, it should be assessed more frequently, not just for the issue of bias relating to question choice, but also for wider concerns such as gender, socio-economic status or ethnicity. Here, a Rasch/IRT modelling approach to data analysis has proven to be informative and is recommended for analysing other test data where bias is a concern. The DTF methods offered by Chalmers et al. (2016) enable a detailed interrogation of the data with estimation of statistical significance.

References

Adams, R. J. (2005). Studies in Educational Evaluation Reliabiltiy as a Measurement Design Effect. *Studies in Educational Evaluation*, *31*, 162–163. https://doi.org/10.1016/j.stueduc.2005.05,008

AERA. (2014). *Standards for educational and psychological testing* (5th ed.). Washington DC: American Educational Research Education Association, American Psychological Association & National Council on Measurement in Education.

Andrich, D., & Hagquist, C. (2015). Real and Artificial Differential Item Functioning in Polytomous Items. *Educational and Psychological Measurement*, *75*(2), 185–207. https://doi.org/10.1177/0013164414534258

Andrich, D., & Marais, I. (2019). *A course in Rasch Measurement Theory : Measuring in the Educational, Social and Health Sciences* (1st ed.). Singapore: Springer Texts in Education.

Barrance, R., & Elwood, J. (2018). Young people's views on choice and fairness through their experiences of curriculum as examination specifications at GCSE. *Oxford Review of Education*, *44*(1), 19–36. https://doi.org/10.1080/03054985.2018.1409964

Bejar, I. I. (1983). Subject Matter Experts ' Assessment of Item Statistics. *Educational Testing Service*, 303–310. https://doi.org/10.1177/014662168300700306

Bell, J. F. (1997). Question choice in English literature examinations. *Oxford Review of Education*, *23*(4), 447–458. https://doi.org/10.1080/0305498970230402

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick (Eds.), *Statistical theories of mental test scores.* Addison–Wesley: Reading, MA.

Bradlow, E. T., & Thomas, N. (1998). Item Response Theory Models Applied to Data Allowing Examinee Choice. *Journal of Educational and Behavioral Statistics*, *23*(3), 236–243. https://doi.org/10.3102/10769986023003236

Bramley, T., & Crisp, V. (2017). Spoilt for choice? Issues around the use and comparability of optional exam questions. *Assessment in Education: Principles, Policy and Practice*. https://doi.org/10.1080/0969594X.2017.1287662

Bramley, T., & Vidal Rodeiro, C. L. (2014). *Using statistical equating for standard maintaining in GCSEs and A levels*. *Cambridge Assessment Research Report*. Cambridge, UK: Cambridge Assessment. Retrieved from http://www.cambridgeassessment.org.uk/Images/182461-using-statistical-equating-for-standard-maintaining-in-gcses-and-a-levels.pdf

Bridgeman, B., Morgan, R., & Wang, M. (1997). Choice among Essay Topics : Impact on Performance and Validity. *Journal of Educational Measurement*, *34*(3), 273–286.

Camilli, G. (2006). Test Fairness. In R. L. Brennan (Ed.), *Educational Measurement* (4th ed., pp. 221–256). Westport, CT: American Council on Education/Praeger. https://doi.org/10.1016/j.jfca.2005.12.009

Camilli, G. (2013). Ongoing issues in test fairness. *Educational Research and Evaluation*, *19*(2–3), 104–120. https://doi.org/10.1080/13803611.2013.767602

CCEA. (2015). *Fair access by design*. Council for the Curriculum Examinations & Assessment Welsh Government.

Chalmers, R. P. (2012). mirt: A Multidimensional Item Response Theory Package for the R environment. *Journal of Statistical Software*, *48*(6), 1–29.

Chalmers, R. P., Counsell, A., & Flora, D. B. (2016). It Might Not Make a Big DIF: Improved Differential Test Functioning Statistics That Account for Sampling Variability. *Educational and Psychological Measurement*, *76*(1), 114–140. https://doi.org/10.1177/0013164415584576

Cole, N. S., & Moss, P. A. (1989). Bias in test use. In R. L. Linn (Ed.), *Educational Measurement* (3rd ed., pp. 201–219). Washington DC: American Council on Education.

Cole, N. S., & Zieky, M. J. (2001). The New Faces of Fairness. *Journal of Educational Measurement*, *38*(4), 369–382.

DfE. (2014). *GCE AS and A level subject content for computer science*. Department for Education. Retrieved from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/302105/A_level_computer_science_subject_content.pdf

Domingue, B. W., Lang, D., Cuevas, M., Castellanos, M., Lopera, C., Mariño, J. P., … Shavelson, R. J. (2017). Measuring Student Learning in Technical Programs. *AERA Open*, *3*(1), 233285841769299. https://doi.org/10.1177/2332858417692997

Fitzpatrick, A. R., & Yen, W. M. (1995). The Psychometric Characteristics of Choice Items. *Journal of Educational Measurement*, *32*(3), 243–259.

Gipps, C. V., & Murphy, P. (1994). *A fair test? : assessment, achievement and equity*. Buckingham: Open University Press.

Goldstein, H. (1979). Consequences of Using the Rasch Model for Educational Assessment. *British Educational Research Journal*, *5*(2), 211–220.

Gould, S. J. (1996). The Mismeasure of Man. In *The Mismeasure of Man* (pp. 176–263). London: Penguin. https://doi.org/10.2307/1494302

Hagquist, C., & Andrich, D. (2017). Recent advances in analysis of differential item functioning in health research using the Rasch model. *Health and Quality of Life Outcomes*, *15*(1), 1–8. https://doi.org/10.1186/s12955-017-0755-0

Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of Item Response Theory*. Newbury Park, California: Sage.

Herrnstein, R. J., & Murray, C. A. (1994). *The bell curve : intelligence and class structure in American life*. New York ; London: Free Press.

Impara, J. C., & Plake, B. S. (1998). Teachers ' Ability to Estimate Item Difficulty : A Test of the Assumptions in the Angoff Standard Setting Method. *Journal of Educational Measurement*, *35*(1), 69–81.

Isaacs, T., Zara, C., Herbert, G., Coombs, S. J., & Smith, C. (2013). *Key Concepts in Educational Assessment* (1st ed.). London: SAGE Publications Ltd.

Kolen, M. J., & Brennan, R. L. (2010). *Test Equating, Scaling, and Linking* (3rd ed.).

Kopf, J., Zeileis, A., & Strobl, C. (2015). Anchor Selection Strategies for DIF Analysis: Review, Assessment, and New Approaches. *Educational and Psychological Measurement*, *75*(1), 22–56. https://doi.org/10.1177/0013164414529792

Lee, H. S., & Geisinger, K. F. (2016). The Matching Criterion Purification for Differential Item Functioning Analyses in a Large-Scale Assessment. *Educational and Psychological Measurement*, *76*(1), 141–163. https://doi.org/10.1177/0013164415585166

Linacre, J. M. (1998). Detecting multidimensionality: which residual data-type works best? *Journal of Outcome Measurement*, *2*(3), 266–283.

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental test scores*. Reading, MA: Addison-Wesley.

Masters, G. N. (1982). A Rasch model for partial credit scoring. *Psychometrika*, *47*(2), 149–174. https://doi.org/10.1007/BF02296272

Messick, S. (1989). Validity. In R. L. Linn (Ed.), *Educational Measurement* (3rd ed., pp. 13–100). Washington DC: American Council on Education.

Morin, C., Holmes, S., & Black, B. (2018). *Online standardisation*. Retrieved from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/759206/Online_standardisation_-_FINAL64491.pdf

Nisbet, I., & Shaw, S. D. (2019). Fair assessment viewed through the lenses of measurement theory. *Assessment in Education: Principles, Policy and Practice*, *26*(5), 612–629. https://doi.org/10.1080/0969594X.2019.1586643

Ofqual. (2018). Inter-board comparability of grade standards in GCSEs, AS and A levels

2018, 1–11.

Osterlind, S. J. (1983). Introduction. In *Test Item Bias*. Sage Publications.

Osterlind, S. J., & Everson, H. T. (2009). Chapter 11. Specialized DIF Procedures In : Differential Item Functioning. In *Differential Item Functioning* (2nd ed., pp. 66–70).

Panayides, P., Robinson, C., & Tymms, P. (2010). The assessment revolution that has passed England by: Rasch measurement. *British Educational Research Journal*, *36*(4), 611–626. https://doi.org/10.1080/01411920903018182

Pollitt, A., Ahmed, A., & Crisp, V. (2007). The Demands of Examination Syllabuses and Questions. In *Techniques for monitoring the comparability of examination standards* (pp. 166–211). https://doi.org/10.1128/MCB.21.9.3083-3095.2001

Preece, P. F. W. (1980). On Rashly Rejecting Rasch: a response to Goldstein: With a rejoinder from Goldstein. *British Educational Research Journal*, *6*(2), 209–212. https://doi.org/10.1080/0141192800060209

R Core Team. (2018). R: A language and environment for statistical computing. Vienna, Austria.: R Foundation for Statistical Computing. Retrieved from https://www.r-project.org/.

Raju, N. S. (1990). Determining the Significance of Estimated Signed and Unsigned Areas Between Two Item Response Functions. *Applied Psychological Measurement*, *14*(2), 197–207. https://doi.org/10.1177/014662169001400208

Raju, N. S., Fortmann-Johnson, K. A., Kim, W., Morris, S. B., Nering, M. L., & Oshima, T. C. (2009). The item parameter replication method for detecting differential functioning in the polytomous DFIT framework. *Applied Psychological Measurement*, *33*(2), 133–147. https://doi.org/10.1177/0146621608319514

Raju, N. S., Van Der Linden, W. J., Fleer, P. F., Roju, N. S., Van Der Linden, W. J., & Fleer, P. F. (1995). IRT-Based Internal Measures of Differential Functioning of Items and Tests. *Applied Psychological Measurement*, *19*(4), 353–368. https://doi.org/10.1177/014662169501900405

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Measurement Research.

Robinson, C. (2008). Awarding Examination Grades : Current Processes and Their Evolution. In *Techniques for monitoring the comparability of examination standards* (pp. 97–123). https://doi.org/10.1006/jmps.2000.1351

Robitzsch A, Kiefer T, W. M. (2020). TAM: Test Analysis Modules. R package version 3.4-26, https://CRAN.R-project.org/package=TAM.

RS. (2012). *Shut down or restart? The way forward for computing in UK schools*. London: The Royal Society. Retrieved from https://royalsociety.org/-/media/education/computing-in-schools/2012-01-12-summary.pdf%0Ahttp://dx.doi.org.ezproxy.elib10.ub.unimaas.nl/10.1111/bjet.12453.

RS. (2017). *After the reboot : computing education in UK schools*. London: The Royal Society. Retrieved from https://royalsociety.org/-/media/policy/projects/computing-education/computing-education-report.pdf

Shealy, R. ., & Stout, W. (1993). A model-based standardization approach that separates true bias/DIF from group ability differences and detects test bias/DTF as well as item bias/DIF. *Psychometrika*, *58*(2), 159–194. https://doi.org/10.1007/BF02294572

Shealy, R. ., & Stout, W. . (2015). An Item Response Theory Model for Test Bias and Differential Test Functioning. In P. Holland & H. Wainer (Eds.), *Differential Item Functioning* (pp. 197–241). Hillsdale: Lawrence Erlbaum Associates.

Stafford, R. E. . (1971). The Speededness Quotient : A New Descriptive Statistic for Tests. *Journal of Educational Measurement*, *8*(4), 275–277.

Stout, W. (2002). Psychometrics: From practice to theory and back. 15 Years of Nonparametric Multidimensional IRT, DIF/Test Equity, and Skills Diagnostic Assessment. *Psychometrika*, *67*(4), 485–518. https://doi.org/0033-3123/2002-4/2002-1029-A

Thissen, D., Steinberg, L., & Wainer, H. (1993). Detection of Differential Item Functioning Using the Parameters of Item Response Models. In *Differential Item Functioning* (pp. 67–111). Hillsdale, NJ: Lawrence Erlbaum.

Wainer, H., & Thissen, D. (1994). On Examinee Choice in Educational Testing. *Review of Educational Research*. https://doi.org/10.3102/00346543064001159

Wang, X., Wainer, H., & Thissen, D. (1993). On The Viability of Some Untestable Assumptions in Equating Exams That Allow Examinee Choice.

Willmott, A. S., & Hall, C. G. W. (1975). *O level examined: the effect of question choice.* (Macmillan). London.

Woods, C. M. (2009). Empirical selection of anchors for tests of differential item functioning. *Applied Psychological Measurement*, *33*(1), 42–57. https://doi.org/10.1177/0146621607314044

Wright, D. B., & Masters, G. N. (1990). Computation of OUTFIT and INFIT statistics. *Rasch Measurement Transactions*, *3*(4), 84–85.

Yen, W. M. (1984). Effects of local item dependence on the fit and equating performance of the three-parameter logistic model. *Applied Psychological Measurement*, *8*, 125–145.

Appendix A BSc Computer science offers at the top 20 ranked UK universities for computer science

Rank	University	A-level grades required	Maths A-level?	Computer science A-level?	Notes
1	Cambridge	A*A*A	Y	Y	
2	Oxford	A*AA	Y		A* in Maths, Further Maths or Computing/Computer Science
3	Imperial	A*A*A - A*AAA	Y	recommended	Maths= A*
4	St Andrews	AAA	Y	or other numerate	
5	Warwick	A*AA	Y		Maths= A
6	Leeds	AAA	or CS	or Maths	Lower offers possible if aptitude shown
7	Glasgow	AAB	Y		
8	Durham	A*AA	Y		
9	Bath	A*AA - AAA	Y		
10	Edinburgh	ABB-AAA	Y		
11	Exeter	ABB - AAA	Y		Maths=B
12	Manchester	A*AA	Y		Maths=A*
13	UCL	A*A*A	Y		Maths=A*
14	Southampton	A*AA	Y		Maths=A, or A*A*B with A* Maths
15	Bristol	A*AA - AAB	Y		Maths=A*; Contextual offer: AAB including A in Maths
16	Birmingham	AAA	Y		Maths=A
17	Nottingham	AAA			AAB with A in Computer Science
18	Kings	AAA			Need at least one of Maths, Further Maths, Computing, or Computer Science
19	Sheffield	AAA	Y		or AAB with A in Maths and Computer science
20	York	AAB - AAB	Y		Maths=B

Data retrieved from: https://www.thecompleteuniversityguide.co.uk/league-tables/rankings/computer-science (Nov, 2019)

Appendix B Curec Approval

CUREC ED-C1A-20-067 - Approval

Laura Molway <laura.molway@education.ox.ac.uk> Wed, Dec 11, 2019 at 11:34 AM
To: [redacted]
C[redacted] Education Research Office <research.office@education.ox.ac.uk>

Dear[redacted]

Title: On the comparability of different programming language routes through A-level Computer Science
Ref: ED-C1A-20-067

The above application has been considered on behalf of the Departmental Research Ethics Committee (DREC) in accordance with the procedures laid down by the University for ethical approval of all research involving human participants.

I am pleased to inform you that, on the basis of the information provided to DREC, the proposed research has been judged as meeting appropriate ethical standards, and accordingly, approval has been granted.

Should there be any subsequent changes to the project which raise ethical issues not covered in the original application you should submit details to research.office@education.ox.ac.uk for consideration.

Good luck with your research study.

Yours sincerely,

Laura Molway
Member of DREC

Laura Molway
DPhil Candidate and Departmental Lecturer in Modern Languages Education
University of Oxford, Department of Education

Appendix C Examples of related questions

Example 1 Use of screen capture

0 5 Write a program that gets **two** words from the user and then displays a message saying if the first word can be created using the letters from the second word or not.

For example:

- The word EAT can be formed from the word ATE as the first word uses one E, one A and one T and the second word also contains one of each of these letters.
- The word EAT can be formed from the word HEART as the second word contains one E, one A and one T which are the letters needed to form the first word.
- The word TO can be formed from the word POSITION as the second word contains one T and (at least) one O which are the letters needed to form the first word.
- The word MEET cannot be formed from the word MEAT as the second word only contains one E and two Es are needed to form the first word.

You may assume that the user will only enter words that consist of upper case letters.

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

0 5 . 1 Your PROGRAM SOURCE CODE.

[12 marks]

0 5 . 2 SCREEN CAPTURE(S) showing the result of testing the program by entering:

- the word NINE followed by the word ELEPHANTINE.
- the word NINE followed by the word ELEPHANT.

[1 mark]

Example 2 Related topics

0 4 . 1 Explain the functionality of the * metacharacter when it is used in a regular expression.

[1 mark]

0 4 . 2 Explain the functionality of the ? metacharacter when it is used in a regular expression.

[1 mark]

Example 3 Referring to a common resource

0 8 A set is an unordered collection of values in which each value occurs at most once. The items in the game can be described using sets.

The set **A** contains all the items in the game.

The set **B** contains all the items that have "use" in their `Commands`.

The set **C** contains all the items that have "gettable" in their `Status`.

The set **D** contains all the items in the player's inventory.

The set **E** contains all the items which have a `Location` equal to the `CurrentLocation` of the player and that have "usable" in their `Status`.

The set **F** contains all the items which have a `Location` equal to the `CurrentLocation`.

Four operations that can be performed on sets are union, difference, intersection and membership.

0 8 . 1 Explain how the operations can be used with some of the sets **A–F** to produce the set of items that the player can use in the current game state.

[3 marks]

0 8 . 2 The set described in question **08.1** is a proper subset of some of the sets **A**–**F**.

List all of the sets (**A–F**) that it is a proper subset of.

[1 mark]

Appendix D Additional Tables and Figures

Table D1 Infit/outfit statistics by option for the PCM

	C#		Java		Pascal		Python		VB.NET	
item	Outfit	Infit	Outfit	Infit	Outfit	Infit	Outfit	Infit	Outfit	Infit
A1.1	0.63	0.79	0.58	0.83	0.68	0.88	0.60	0.80	0.68	0.80
A1.2	1.07	1.03	1.03	1.00	1.17	1.04	1.02	1.01	0.94	0.94
A1.3	0.84	0.87	0.81	0.85	0.66	0.73	0.89	0.92	0.84	0.89
A2.1	1.47	1.22	1.45	1.20	1.51	1.53	1.40	1.21	1.31	1.27
A2.2	0.65	0.91	0.67	0.90	0.79	0.94	0.68	0.92	0.65	0.90
A2.3	0.81	0.97	0.83	0.99	0.86	0.96	0.81	0.96	0.78	0.96
A2.4	0.87	0.91	0.89	0.90	0.81	0.87	0.84	0.88	0.88	0.87
A3.1	0.97	1.01	0.88	1.00	0.93	1.02	1.02	1.00	1.03	1.02
A3.2	1.12	1.07	1.39	1.21	1.18	1.22	1.13	1.12	1.06	1.05
A3.4	1.02	1.00	1.02	0.98	0.91	0.96	0.99	0.99	0.94	0.99
A3.5	0.82	0.87	0.93	0.96	0.83	0.88	0.86	0.89	0.84	0.87
A3.6	0.81	0.83	0.84	0.88	0.82	0.85	0.84	0.87	0.76	0.81
A4.1	1.09	0.99	0.94	0.94	0.69	0.77	0.94	0.93	1.02	1.00
A4.3	1.57	1.33	1.18	1.08	1.77	1.47	1.43	1.25	1.55	1.27
B5.1	1.70	1.44	1.13	1.13	1.41	1.45	1.68	1.21	1.60	1.34
C6	1.06	1.06	1.10	1.08	0.99	1.00	1.03	1.03	1.07	1.06
C7.1	0.93	0.92	0.96	0.97	0.83	0.91	0.94	0.94	0.90	0.92
C7.2	0.79	0.82	0.79	0.82	0.74	0.78	0.80	0.83	0.78	0.81
C7.3	0.79	0.85	0.78	0.86	0.83	0.89	0.81	0.86	0.74	0.81
C7.4	0.88	0.93	0.72	0.85	0.85	0.94	0.80	0.91	0.88	0.94
C7.5	0.76	0.82	1.05	1.00	0.83	0.84	0.84	0.90	0.90	0.94
C7.6	0.93	0.97	0.87	0.92	0.99	1.01	0.94	0.96	0.95	0.97
C8.1	1.04	0.98	1.01	1.02	1.10	1.10	0.99	0.99	0.99	1.03
C8.3	0.76	0.92	0.64	0.91	0.82	0.89	0.84	0.94	0.73	0.90
C8.4	0.83	0.91	0.81	0.92	0.87	0.92	0.83	0.90	0.78	0.88
C9.1	0.96	0.93	0.93	0.99	1.72	1.18	1.06	1.01	1.07	1.02
C9.3	0.86	0.88	0.84	0.89	0.78	0.82	0.86	0.89	0.84	0.87
C9.4	0.97	0.99	0.92	0.94	0.71	0.93	0.93	0.97	0.83	0.93
C9.5	0.99	1.00	0.95	0.97	0.93	0.99	0.98	0.99	1.02	1.03
D10.1	0.81	0.91	1.00	0.96	1.11	1.12	0.81	0.90	1.08	1.00
D11.1	1.12	0.87	0.91	0.80	0.95	0.93	1.05	0.91	1.08	0.99
D12.1	1.20	1.19	1.40	1.31	1.06	1.08	1.17	1.13	1.11	1.07
D13.1	1.14	1.08	1.18	1.07	0.98	1.03	1.20	1.13	1.03	1.03

Table D2 DIF analysis for all groups PCM

Item	Tariff	ΔAIC	ΔBIC	X^2	d.f.	p
A1.1	4	5.51	107.43	26.50	16	0.047
A1.2	2	-29.89	21.07	45.89	8	<.001
A1.3	2	5.74	56.70	10.26	8	0.247
A2.1	5	-8.33	119.08	48.33	20	<.001
A2.2	1	1.19	26.68	6.81	4	0.146
A2.3	1	-1.82	23.66	9.82	4	0.044
A2.4	2	-3.11	47.86	19.11	8	0.014
A3.1	1	-4.99	20.49	12.99	4	0.011
A3.2	2	0.93	51.89	15.07	8	0.058
A3.4	1	-18.93	6.55	26.93	4	<.001
A3.5	2	-20.99	29.97	36.99	8	<.001
A3.6	1	1.34	26.83	6.66	4	0.155
A4.1	2	-4.54	46.42	20.54	8	0.008
A4.3	3	3.09	79.54	20.91	12	0.052
B5.1	13	-133.92	197.35	237.92	52	<.001
C6	2	-43.36	7.61	59.36	8	<.001
C7.1	1	-79.01	-53.53	87.01	4	<.001
C7.2	1	5.24	30.72	2.76	4	0.598
C7.3	1	-7.97	17.51	15.97	4	0.003
C7.4	2	-40.82	10.15	56.82	8	<.001
C7.5	4	-17.10	84.83	49.10	16	<.001
C7.6	1	-11.82	13.66	19.82	4	0.001
C8.1	4	10.16	112.08	21.85	16	0.148
C8.3	1	-7.92	17.56	15.92	4	0.003
C8.4	1	-6.60	18.88	14.60	4	0.006
C9.1	2	-49.80	1.16	65.80	8	<.001
C9.3	1	-35.74	-10.26	43.74	4	<.001
C9.4	1	-49.11	-23.62	57.11	4	<.001
C9.5	1	-46.59	-21.11	54.59	4	<.001
D10.1	5	-67.73	59.68	107.73	20	<.001
D11.1	8	-36.06	167.80	100.06	32	<.001
D12.1	13	-10.95	320.32	114.95	52	<.001
D13.1	9	-2.08	227.26	74.08	36	<.001

Figure D3 Items showing significant DIF in the programming section of Paper 1 – all options

(A: C#; B: Java; C: Pascal; D: Python; E: VB.NET)

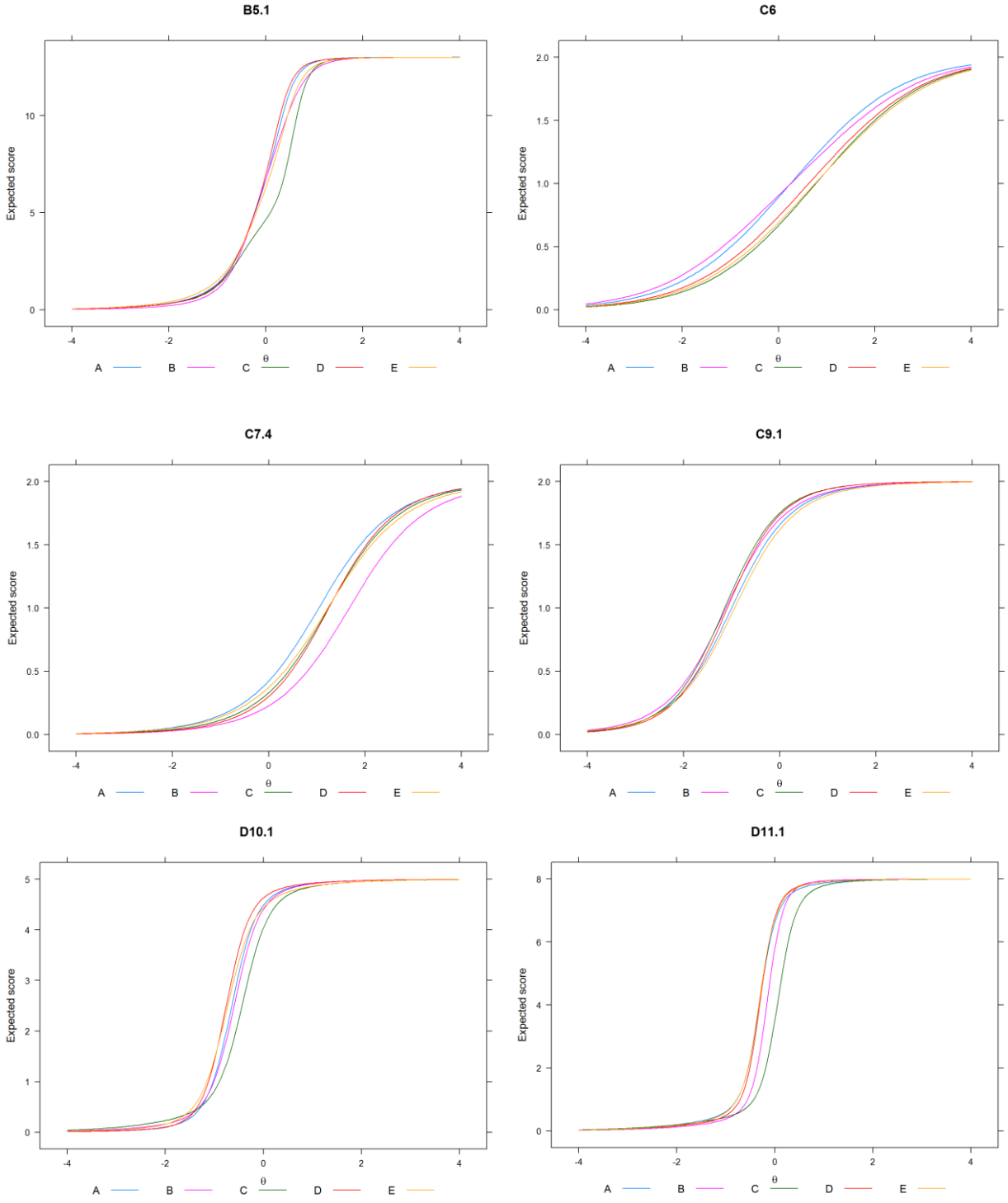


Figure D3 cont. **Items showing significant DIF in the programming section of Paper 1 – all options**

(A: C#; B: Java; C: Pascal; D: Python; E: VB.NET)

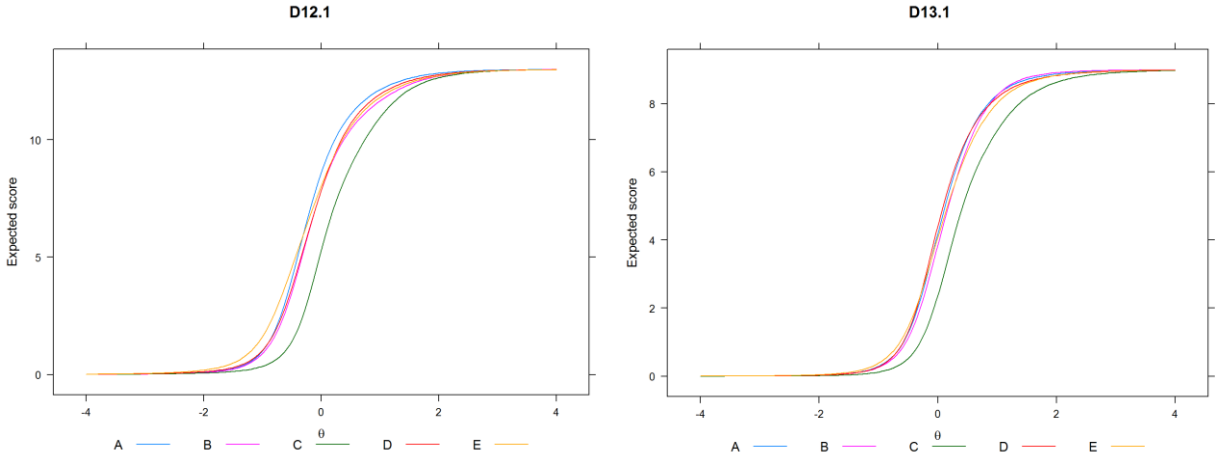


Table D4 Anchor items for other comparisons

Comparison	Anchor items	# items	% marks	χ^2-test[1] p-value
C# & Python	A1.1, A1.3, A2.2, A2.4, A3.1, A3.2, A3.5, A3.6, A4.1, A4.3	10	20%	0.64
C# & VB.NET	A1.1, A1.2, A1.3, A2.1, A2.2, A2.3, A2.4, A3.1, A3.6, A4.1, A4.3	11	24%	0.28
Java & Python	A1.1, A1.2, A1.3, A2.2, A2.3, A3.2, A3.4, A3.5, A3.6, A4.3	10	19%	0.67
Pascal & Python	A1.3, A2.1, A2.4, A3.1, A3.5	5	12%	0.13

[1]Comparison of equated model (with separate means and variances) to the fully independent model

Table D5 Model comparison statistics for pairwise DTF evaluations

Python v VB.NET

	AIC	BIC	logLik	χ^2	df	p
Independent model	168968.2	170183.8	-84282.1			
section A anchor	168976.3	170023.3	-84314.1	64.1	28	0.000
Invariant anchor	168955.3	170104.7	-84286.7	9.13	11	0.610

C# v Python

	AIC	BIC	logLik	χ^2	df	p
Independent model	160195.9	161399.9	-79896.0			
section A anchor	160193.7	161230.8	-79922.9	53.8	28	0.002
Invariant anchor	160174.2	161265.0	-79904.1	16.3	19	0.638

Java v Python

	AIC	BIC	logLik	χ^2	df	p
Independent model	132359.0	133526.1	-65977.5			
section A anchor	132346.4	133351.7	-65999.2	43.3	28	0.032
Invariant anchor	132337.9	133401.1	-65985.0	14.9	18	0.667

Pascal v Python

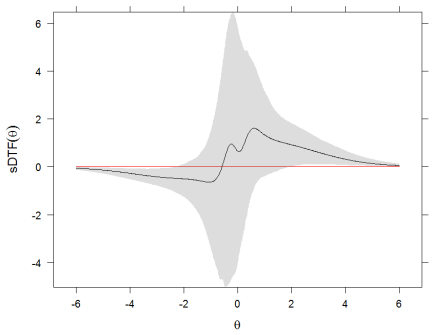
	AIC	BIC	logLik	χ^2	df	p
Independent model	127311.6	128470.6	-63453.8			
section A anchor	127380.3	128378.6	-63516.1	124.7	28	0.000
Invariant anchor	127305.8	128401.7	-63461.9	16.2	11	0.133

C# v VB.NET

	AIC	BIC	logLik	χ^2	df	p
Independent model	97689.3	98790.8	-48642.7			
section A anchor	97692.3	98641.1	-48672.1	58.9	28	0.001
Invariant anchor	97669.8	98645.9	-48655.9	26.5	23	0.280

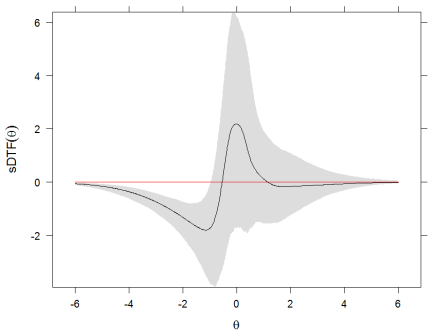
Figure D6 Additional sDTF$_\theta$ plots

C# v Java



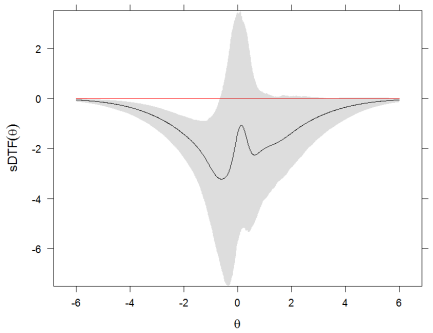
sDTF=0.204 (-0.70, 1.13)

C# v Pascal



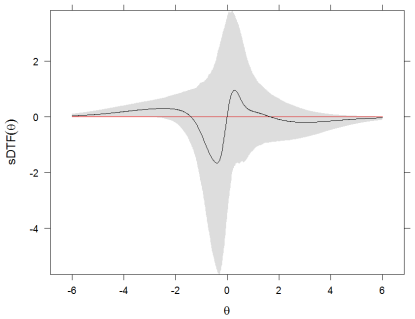
sDTF=-0.215 (-1.02, 0.54)

Java v Pascal



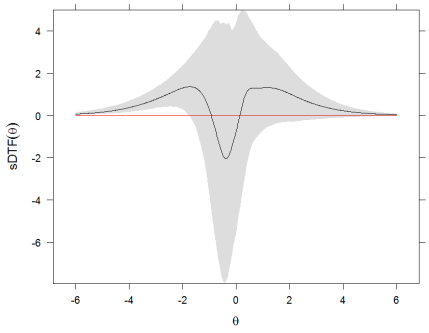
sDTF=-1.101 (-1.90, -0.07)

Java v VB.NET



sDTF=-0.024 (-0.72, 0.63)

Pascal v VB.NET



sDTF=0.438 (-0.72, 1.54)