School of Education

Development and Application of a Rasch Model Measure of Student Competency in University Introductory Computer Programming

Leela Waheed

This thesis is presented for the Degree of
Doctor of Education
of
Curtin University

November 2018

Declaration: To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no materials, which have been accepted for the award of any other degree or diploma in any university.

Signature: ……..

Date: 04/11/2018

Acknowledgements

I would like to mention Professor Rob Cavanagh first to show my gratitude. I am indebted to him for his kindness, and invaluable guidance throughout my doctorate studies, making this journey the most rewarding experience of my life. Similarly, I would also like to thank my secondary supervisor, Dr Audrey Cooke, who has supported me considerably. On the same note, I would like to mention Associate professor Genève Johnson who was my former supervisor. She provided endless support and helped me build self-confidence in what I was doing. I am not only thankful to them for being my supervisors and helping me in reaching the successful completion of my thesis, but also for being role models exemplifying the attributes of best student-supervisor relationships.

Just as the proverb goes "behind every successful man there is a woman", the reverse is also true. I am indebted to my husband, Hussain Rafeeu, who has been a great support in me getting this far, and who has relinquished greater opportunities in his own professional career to provide me with this opportunity. The support from my children is also highly appreciated. In particular, my daughter Rafha Rafeeu and my niece Thanaa Ismail, who have both helped me on different occasions, including proofreading sections of this thesis. My son, Raif Rafeeu, who is always delightful, and filled with compelling stories of ancient animals has been the biggest stress relief for me and deserves a special mention.

The seed of learning begins at home and parents are the first and foremost teachers, as well as the providers and greatest supporters in their children's education. My biggest inspiration are my parents who have proven a girl-child is no less than a boy-child by empowering all eight children of whom seven are girls through the provision of equal opportunities to education. I thank my parents and all my sisters and my brother for being positive role models to instill in me the value of education.

My appreciation also goes to all the colleagues and all the staff attending the weekly doctoral colloquium conducted by the school of education. The invaluable experience, feedback, and knowledge exchanged opened my eyes to new research issues, which I benefited immensely from, not only in my own research but also as a source of knowledge enhancement. I also thank Curtin school of education heads for facilitating these meetings and the lecturers attending for taking the time aside from their busy schedule.

Abstract

University introductory programming courses, commonly referred to as Computer Science 1 (CS1), are beset by a paucity of invariant measures for assessing CS1 student programming competency upon completion of a typical CS1 course. Evaluative tools such as university exam scores commonly employed in research inquiries have several flaws that limit the provision of meaningful data for pedagogic and research purposes. Thus, the veracity of statistical associations tested in these studies and the corresponding recommendations for pedagogic reform is questionable.

The purpose of this study was to develop a widely applicable interval-level measure to assess the student competence of first-year CS1 students at the completion of a typical CS1 course. In addition, the study aimed to examine whether statistically significant associations exist between student competences logit scores obtained by the instrument and commonly cited student and learning environment factors.

The sample comprised of 85 students from three universities – 54 from two universities in the Maldives (groups of 25 and 29) and 31 from a university in Malaysia. The participating students had just begun the second semester of their Computer Science course. Total population sampling was used due to the limited number of students studying for Computer Science degrees.

The methodology employed was a positivistic quantitative approach of methods, techniques, and procedures similar to those used in the natural sciences. A phase-based approach was used to guide the research design. The investigation began by conducting a review of the literature to operationalise the construct of CS1 student competence. The process collated data on different aspects of CS1 student competence construct by an in depth review of CS1 literature, widely used curriculum frameworks for CS1 course design, CS1 curriculums at universities, textbooks, and expert feedback. The literature review proposed a construct model representing the latent variable of CS1 student competence. The model consists of the four fundamental skills necessary for learning to program: they are hypothesised to form a hierarchy with the rank order defined by the four levels of the Structure of the Observed Learning Outcome (SOLO) taxonomy (Collis & Biggs, 1982), and five fundamental programming concepts. The students' mastery on each of these topics is to be tested against the skills to gauge student competence.

In Phase one, a construct map was developed to operationalise the construct of CS1 student competence as advanced in the construct model. A total of 20 items (questions) were developed to operationally realise the observable behaviours prescribed in the construct map which were then pilot-tested with ten students. The resulting 20-item test (CS1 measure) was administered to collect the data, which were then tested for fit to the Rasch Unconstrained Partial Credit Model to examine whether the data accorded with the requirements of the Rasch model. The resulting Rasch analysis outputs and displays were used to satisfy the requirements for measurement (see Wright and Masters, 1982). The functioning of polytomous items was tested, by generating category probability curves and item threshold statistics. The fit of the person and items were examined by estimating item and person residuals (< ± 2.5 logits as a benchmark). Item bias was tested by Differential Item Functioning (DIF) analysis using analysis of variance (5% alpha). Dimensionality was checked with Principal Component Analysis (PCA) of the Rasch residual data and t-test procedures, and local dependency was examined via Residual Correlation of items >0.3 above the average correlations. Reliability was assessed by estimating the Person Separation Index (PSI). In the second phase, the validity of data obtained from the measure was investigated using the theoretical frame endorsed by the American Psychological Association, American Educational Research Association, National Council on Measurement in Education, American Educational Research Association, and Committee on Test Standards (2014). In the third phase, the correlational analysis of statistical significance of student and learning environment factors with student competence logit scores obtained from the first phase was performed using one-way Analysis of Variance (ANOVA).

The data collected from the 20-item test developed in the first phase was analysed using RUM2030 and this demonstrated excellent PSI with no evidence of DIF, or misfit of the items or persons. However, there was some disordering of thresholds in data from two of the polytomous items. Hence, the middle two categories (categories with lowest response rates) of two other questions were collapsed into one. PCA of the Rasch residual data and t-test procedures demonstrated strict unidimensionality of the CS1 measure. The fit of the data suggested that a linear scale of persons and items calibrated in logits had been achieved, attaining Wright and Masters (1982) Measurement criteria. The validity framework was well exemplified by Rasch statistics and displays.

The findings of the third phase revealed that the students instructed with C programming language comparably had a slightly higher mean than those instructed with Java or Python respectively, with mean differences approaching a statistically significant level (p =

.050). The only student factors that were significantly predictive of student competence were students' prior programming experience and students' mathematics background, both showing a medium level effect size.

The fit of the data with Rasch requirements suggested that a linear scale of persons and items calibrated in logits had been constructed. There was strong evidence for an argument for project validity. Finally, the correlational analysis findings converged well with existing literature from studies of CS1 competency. The new instrument would be most suitable for future research into CS1 instruction and instructional design.

The main limitation of the study is the sample size. In particular, the limited responses received to some of the polytomous items implicated in the category threshold calculation of these items. Consequently it was uncertain whether the two items initially showed disordered category thresholds were collapsed correctly. Therefore, this is an area for potential consideration in the future calibrations of the measure.

Table of Contents

Chapter 1 – Introduction to the Problem 1
1.1. Introduction 1
1.2. The Research Problem 1
1.3. Aims and Research Questions 6
1.4. Significance of Study 6
1.5. Structure of Thesis 7
1.6. Summary 9
Chapter 2 – Background 10
2.1. Introduction 10
2.2. Current Assessment Practices in CS1 10
2.3. Existing CS1 Competency Instruments 13
2.4. Summary 16
Chapter 3 – Literature Review: The Construct of CS1 Student Competence 17
3.1. Introduction 17
3.2. Conceptualising the Construct of CS1 Student Competence 17
3.2.1. Computer Science curricula 2013 17
3.2.2. Other scholarly work 18
3.2.3. Core computer programming literacy skills 22
3.2.4. Models of human conceptualisation 26
3.2.5. A construct model of CS1 student competence 30
3.3. Factors Associated with Student CS1 Competence 31
3.3.1. Student factors 31
3.3.2. Learning environment factors 34
3.3.3. A model of factors influencing CS1 student competence 38
3.4. Summary 38
Chapter 4 –Measurement Development and Validity Evaluation Models 40
4.1. Introduction 40
4.2. Conceptions of Validity and the Unified View 40
4.2.1. Review of the evolution of the concept of validity 40
4.3. Comparison of Measurement Models 44
4.3.1. Overview 44
Item, test and person statistics 46
4.3.2. Model Fit 51
4.3.3. Reliability, internal consistency, and measurement error 52

4.3.4. Missing data 54
4.3.5. Score meaning 58
4.3.6. Scale properties 60
4.3.7. Practical implication of scale levels on statistical analysis 63
4.4. Rasch Measurement Theory 65
4.5. Summary 70
Chapter 5 - Methodology 71
5.1. Introduction 71
5.2. Aims and Research Questions 71
5.3. Methodology 71
5.4. Research Approach 72
5.5. Research Design 72
5.5.1. Phase one: CS1 measure development 73
5.5.2. Building block 1: Construct map 74
5.5.3. Building block 2: Item design 75
5.5.4. Building block 3: Outcome space 79
5.5.5. Building block 4: The measurement model 80
5.6. Phase Two: Validity Evidence 87
5.7. Phase Three: Correlational Analysis 90
5.7.1. Sample, instruments and data collection 91
5.7.2. Data analysis 92
5.8. Ethical Issues 92
5.9. Summary 93
Chapter 6 – Results 94
6.1. Introduction 94
6.2. Phase 1: CS1 Measure Development 94
6.2.1. Construct map 94
6.2.2. Item development 94
6.2.3. Outcome space 97
6.2.4. Rasch analysis of the data (Measurement model) 98
6.3. Phase 2: Validity Evidence 111
6.3.1. Evidence of the content aspect 111
6.3.2. Evidence of the substantive aspect 113
6.3.3. Evidence of the structural aspect 116
6.3.4. Evidence of the generalisability aspect 116
6.3.5. Evidence of the external aspect 117

6.3.6. Evidence of the consequential aspect 117
6.3.7. Evidence of the interpretability aspect 117
6.4. Phase Three: Correlational Analysis 118
6.4.1. Results 118
6.5. Summary 125
Chapter 7 – Discussion and Conclusion 126
7.1. Phase One: Instrument Development 126
7.1.1. Research question 1 126
7.2. Phase Two: Validity Evidence 136
7.2.1. Research question 2 136
7.3. Phase Three: Correlational Study 145
7.3.1. Research question 3 146
7.3.2. Research question 4 149
7.4. Summary and Conclusion 151
7.5. Limitations and Future Directions 152
References 155
Appendices 191

List of Tables

Table 2.1 Psychometric Properties of Some of the CS1 Student Competence Measures 11
Table 3.1 Fundamental Programming Concepts 18
Table 3.2 Summary of Topic Coverage Comparison with the Tew and Guzdial Conceptual Framework 20
Table 3.3 Fundamental Programming Concepts 21
Table 3.4 Descriptors of the Learning Outcomes and Their Associated Mastery Levels 24
Table 3.5 Clear et al.'s (2008) SOLO Categories for Code Writing 29
Table 4.1 Comparison of IRT Models 46
Table 5.1 Comparability between Java and Python Code for the Same Task 79
Table 5.2 Characteristics of the Subjects (N=84) 81
Table 5.3 Independent Variables of the Study 91
Table 6.1 Difficulty Rankings of Question 1 Based on Total Number of Correct Answers 96
Table 6.2 Summary of Overall Fit between the Data and the Rasch Model 99
Table 6.3 Category Response Frequencies 100
Table 6.4 Items with Disordered Thresholds 102
Table 6.5 Item Fit Statistics after Rescoring (N=20) 103
Table 6.6 Uniform and Non-uniform DIF Statistics (N=20) 104
Table 6.7 Residual Correlation Matrix of all Items after Taking the Rasch factor 108
Table 6.8 Principal Components Summary 109
Table 6.9 Principal Component Analysis of the Residuals Showing First Component Loadings 109
Table 6.10 Expected Vs. Observed Difficulty of the Items 114
Table 6.11 Overall Question Difficulty Distribution 115
Table 6.12 One-way ANOVA: Effect of Language of Instruction on Student Competence 120
Table 6.13 One-way ANOVA: Effect of Prior Programming Experience on Student Competence 121
Table 6.14 One-way ANOVA: Effect of Mathematics Background on Student Competence 122
Table 6.15 One-way ANOVA: Effect of High School Stream on Student Competence 123
Table 6.16 One-way ANOVA: Effect of High School CS Course on Student Competence 125

List of Figures

Figure 3.1. Different computer programming tasks found in Introductory CS books. 23
Figure 3.2. Difficulty hierarchy of the three main programming skills. 26
Figure 3.3. Proposed generic construct model of CS1 student competence 30
Figure 3.4. Two broad categories of factors associated with CS1 student competence..... 38
Figure 4.1. The six aspects of evidence in Messick's (1995b) unitary validity framework 42
Figure 4.2. Sample Item Characteristic Curve (ICC) for dichotomous items..... 47
Figure 4.3. A sample CPC for a polytomous item 47
Figure 4.4. Item Information Curves 48
Figure 4.5. Test Information Curve 49
Figure 4.6. Item-person map 50
Figure 4.7. Item curves based on p-levels..... 51
Figure 4.8. TIF and standard error of measurement..... 54
Figure 4.9. Linking scores to tasks. Adapted from "Applying the Rasch model to Psycho-Social Measurement: A Practical Approach" by Wu and Adams (2007), Melbourne: Educational Measurement Solutions, p. 12. 59
Figure 4.10. Linking student abilities to tasks in IRT models. Adapted from "Applying the Rasch model to Psycho-Social Measurement: A Practical Approach" by Wu and Adams (2007), Melbourne: Educational Measurement Solutions, p. 15. 60
Figure 4.11. ICC showing probability of success on an item as the ability increases 60
Figure 4.12. Student score distance variance in easy and difficult test in CTT. Adapted from "Applying the Rasch Model to Psycho-Social Measurement: A Practical Approach" by Wu and Adams (2007), Melbourne: Educational Measurement Solutions, p.11..... 61
Figure 4.13. Student score distance variance in easy and difficult test in IRT models..... 63
Figure 4.14. Formulae for the three parameterisations of the Rasch model. 66
Figure 5.1. Research design 72
Figure 5.2.Wilson's construct modelling approach incorporating validity aspects 74
Figure 5.3.The construct map for loop structure (construct 3)..... 75
Figure 6.1. Question 1D before and after re-wording and re-illustration 95
Figure 6.2. An expected level solution for Q2D at the 4th level (Extended abstract)..... 98
Figure 6.3. A sample student solution for Q2D scored at 3rd Level (Relational)..... 98
Figure 6.4. Category probability curves for Q5D before collapsing..... 101
Figure 6.5. Category probability curves for Q5D after collapsing..... 101
Figure 6.6. ICC for 4D showing a good fit to the model 103
Figure 6.7. ICC for Question 3D showing no significant DIF 107
Figure 6.8. Summary of independent t-tests 109

Figure 6.9. Item-person map 110
Figure 6.10. Item-person threshold distribution map for the CS1 measure 111
Figure 6.11. Category probability curves for Question 4D 116
Figure 6.12. Frequency distributions of students based on programming language 119
Figure 6.13. Frequency distributions of students with (Y) and without (N) programming experience 121
Figure 6.14. Frequency distributions of students with year 12 and year 10 mathematics background 122
Figure 6.15. Frequency distributions of students studied in Science and Commerce stream 123
Figure 6.16. Frequency distributions of students who had (Y) and who had not (N) studied high school CS 124

Appendices

Appendix I: Construct Map 191
Appendix II : Scoring Model for Writing Questions (i.e., part (d)) 193
Appendix III: Expert Feedback 194
Appendix IV: Sample Question Set for Loops (Java Version) 195
Appendix V: Ethics Approval 196
Appendix VI: Participant Consent Form 197
Appendix VII: Participant Information Statement 198
Appendix VIII: Sample Request for Approval 200
Appendix IX: Survey Form 201

List of Abbreviations

ACM	Association for Computing Machinery
ACT/SAT	American College Testing and Scholastic Aptitude Test
ANOVA	Analysis of Variance
AP	Advanced Placement
APU	Asia Pacific University
BRACE	Building Research in Australasian Computing Education
ITiCSE	Innovation and Technology in Computer Science Education
CS	Computer Science
CS1	Computer Science 1
CS2	Computer Science 2
CS2013	Computer Science Curricula 2013
CTT	Classical Test Theory
DIF	Differential Item Functioning
ERG	Expert Review Group
I/O	Input/output
ICC	Item Characteristic Curve
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IIF	Item Information Function
IRT	Item Response Theory
IT	Information Technology
KA	Knowledge Areas
KU	Knowledge Units
MCQ	Multiple Choice Questions
MNMI	Multinational, Multi-institutional
MNU	Maldives National University
OF	Objects-First
OO	Object Oriented
OOD	Object-Oriented Design
OOP	Object Oriented Programming
PF	Procedures-First

PCA	Principal Component Analysis
PCM	Partial Credit Model
PSI	Person Separation Index
RMT	Rasch Measurement Theory
SDF	Software Development Fundamentals
SEM	Standard Error of Measurement
SOLO	Structure of the Observed Learning Outcome taxonomy
TIF	Test Information Function

Chapter 1 – Introduction to the Problem

1.1. Introduction

This thesis endeavours to understand issues relating to the continued low performance of university students in introductory programming, Computer Science (CS1). CS1 is often seen as the cornerstone of all Computer Science (CS) degree programs. The performance in this course has significant implications for student progress and influences students' decisions to study further programming courses. A secondary purpose of the study is to develop an interval level measurement of CS1 competency. This is required for the assessment of CS1 students, to conduct research into the effects on this ability, and research into the design and implementation of CS1 programs.

Chapter 1 provides the context of the study, specifying the research problem. This is followed by the study's aims and objectives, then the significance of the research, and finally an outline of the structure of the thesis is given.

1.2. The Research Problem

Introductory computer programming often referred to as CS1 is typically the first study unit in university-level computer programming (Cardell-Oliver, 2014). According to recent reports, there is an enrolment surge for Computer Science (CS) in the United States of America and Canada (Computing Research Association, 2017). Although this does not represent worldwide trends in CS education, presumably, it is an indication that globally we are beginning to recover from the enrolment crisis caused by the collapse of the dot.com bubble in 2002. With this rejuvenated interest in Computer Science courses, the next critical step concerns how to retain enrolled students. Unfortunately, the attrition and dropout rates of CS and IT related courses are still the highest according to some reports (Chen, 2013), which is often linked to programming courses (Raigoza, 2017). One study reported that the global pass rate of CS1 was estimated to be 67%, however, large variations in the pass, fail, abort, and skip rates were also reported (Watson & Li, 2014).

The literature has consistently revealed overall low levels of student performance in CS1 on traditional examinations, tests, and assignments. A study conducted by the Innovation and Technology in Computer Science Education (ITiCSE) 2001 working group (McCracken et al., 2001) examined the programming ability of 216 students from eight different institutions. The result showed a low average score of 22.89 out of 110 points on an assessment of the basic

concepts of programming. Similarly, Lister et al. (2004) assessed the code reading ability and tracing skills of 556 students from six different institutions and concluded that many lacked the ability to analyse and answer small snippets of code. This led to a major shift of attention in the study to investigate plausible factors associated with student competence, with the aim of redressing this untenable situation.

Within CS education research, there have been several studies exploring the determinants of student competence in CS1. The antecedent research focused on predicting and filtering the likely people who would make a successful career in the emerging computing industry based on Aptitude tests (Fincher et al., 2006). However, following an increased demand for Information Technology (IT) professionals and the decrease in the number of CS graduates as a result of high failure and drop-out rates, the focus shifted to researching student factors such as demographic (Sauter, 1986), psychological (Whipkey, 1984) and cognitive traits (Barker & Unger, 1983) as predictors of performance (Watson, Li, & Godwin, 2013). Afterwards, the research attention shifted away from single factors into more explanatory modes of inquiry such as the use of linear regression models to explain performance factors (Bennedsen & Caspersen, 2005; Ramalingam, LaBelle, & Wiedenbeck, 2004; Wilson & Shrock, 2001; Zingaro & Porter, 2016). Most significantly, some researchers have even tried to predict performance based on dynamic student behaviour as they interact with computer program development environments by logging and analysing student programming behaviour (Watson et al., 2013).

However, despite these advances in identifying the potential success factors of students, little research (Lopez, Whalley, Robbins, & Lister, 2008) has focused on measuring the dependent variable (student competence) in a meaningful way. This is despite the availability of measurement models that can provide measures comparable to the physical sciences. The majority of research conducted previously utilised aggregated raw scores of either classroom-based assessments (Hagan & Markham, 2000; Norman & Adams, 2015; Wilson & Shrock, 2001) or researcher developed instruments for their specific purposes (Price & Smith, 2014; Sekiya & Yamaguchi, 2013). The quality of these instruments depend on several institutional factors such as the competence of the instructor who develops the test, test development standards and the control mechanisms set in place by each institute. Generally, these evaluative tools are characterised by a lack of standardised scaling protocols, the absence of construct models to inform instrument design, and inconsistent selection of substantive content; as studies conducted within the last decade suggest (Petersen, Craig, & Zingaro, 2011; Sheard et

al., 2013; Sheard et al., 2011). Consequently, the measurement properties present in the summed scores of these evaluative tools do not meet the measurement criteria for parametric operations.

While studies dedicated to the development of CS1 student evaluative tools such as Decker (2007) and Tew (2010) address several aspects of measurement construction, none of these studies have employed a stringent measurement theory to guide the measurement development processes. Similar to the implications of using raw scores of classroom evaluative tools, the scores from these tools also manifest the same methodological and theoretical limitations. All of which may affect the outcome and subsequent calculations performed on the resulting students' scores. This is because the main body of current CS1 research assumes that the raw scores obtained from these sources are 'sufficient' measures of student competence on which parametric analysis can be performed. However, many authors (Embretson, 1996b; Wright & Masters, 1982; Wu & Adams, 2007) argue that raw scores have limited applicability, and that it is only with using interval scaled data that researchers can justify conducting a parametric analysis.

The use of only nominal and ordinal data has been a recurring topic of debate within psychometricians (Embretson & Reise, 2000). Although some researchers believe when the summed scores of measurement instruments manifest a normal-distribution, parametric analysis can be performed, this idea is refuted by many authors (Embretson, 1996a; Forrest & Andersen, 1986; Maxwell, Delaney, & Manheimer, 1985; Romanoski & Douglas, 2002). Forrest and Andersen (1986) report that when several items are measured on ordinal scales it is questionable whether summed scores manifest even ordinal properties. Similarly, Bond and Fox (2015) raise concern on treatment of data derived from Likert scales being used as interval-level data by researchers during statistical analysis. Several studies (Embretson, 1996a; Maxwell et al., 1985; Romanoski & Douglas, 2002) of the past demonstrated that failing to achieve interval measures can lead to erroneous results in inferential statistics.

Although validity is unarguably the cornerstone of any educational testing and performance measure (Goldstein, 2015), validity issues have not been addressed as a serious concern in most of the CS1 instrument development investigations of the past. Consequently, the few measures developed for gauging CS1 student competence were not evaluated for validity of the score interpretations. Validity of score interpretation is important as they may be used for informed decision making (Kane, 2013). In a pedagogical context, the data may be used to leverage student learning such as the design and development of intervention programs,

evaluation of the effectiveness of remedial programs, or curriculum reform or may even be used in policy decision-making, formulation and implementation (National Research Council, 2001). Because of these potential consequences, Messick (1995) stressed the importance of dealing with validity issues diligently and systematically at the same level of importance as the other aspects – reliability, comparability, and fairness.

The concerns discussed generally stem from an approach associated with the measurement development process. Although the majority of the CS1 research community are unware, the instrument development procedures associated with past instrument development endeavours were mostly CTT based theories. There are several concerns related to CTT based theories which makes it unsuitable for most of our measurement requirements. Firstly, the CTT based theories and its associated procedures fail to comply with cancelation conditions, a condition required for any attribute to be quantitative (Graves, 2013). Similarly, the measures of CTT based theories do not entail invariance across different measurement contexts such as different gender cultural backgrounds (Salzberger, 2013). Secondly, such theories do not provide a common unit to describe the magnitude of the latent variable, which is a fundamental characteristics of all measurements of the physical sciences (Salzberger, 2013). The pseudo-units that are commonly used such as the number of standard deviations are sample dependant, which is hard to interpret independently. A third issue with these theories is the rudimentary methods used for assessing the construct validity of the latent variable which is mainly based on convergent validity, discriminant validity and monological validity (Borsboom, Mellenbergh, & van Heerden, 2004; Michell, 2000). A construct theory about a psychological construct under investigation necessarily requires criteria or an ontological claim that are themselves valid which can be tested. In other words, to validate scores on measures one needs to substantiate them with a structure of existing knowledge to which one can relate those scores (Strauss & Smith, 2009). Unfortunately, none of the validity procedures of the CTT are suitable for establishing the ontological claim entailed by the theory (Borsboom et al., 2004)

Opportunely, contemporary measurement models exist which can deal with both the measurement and the validity issues. The Rasch model advanced by the Danish mathematician George Rasch (Rasch, 1960) is founded on the same core requirements of measurement in physics; namely the requirement of *invariant comparison*. The expected responses based on the Rasch model for measurement have been shown to comply with the axioms of a quantity – the theory of simultaneous, or additive, conjoint measurement (Luce & Tukey, 1964). Therefore, if an instrument data fits to the Rasch model requirements, it provides validity to

the successful quantification of latent variables, and thus, a measurement (Strauss & Smith, 2009). The Rasch model is also a tool that is able to address many of the validity issues that are incapable of being addressed by CTT based methods. In addition to being a valuable tool to address the validity issues, Rasch model is also the only system of true measurement of a latent variables.

Messick's unified validity framework (Messick, 1995) remains as the most comprehensive conceptualisation of the instrument validity testing process available since its initial introduction in 1989 (Sawatzky et al., 2017; Tran, Griffin, & Nguyen, 2010). The framework has been strongly endorsed by professional bodies such as the American Educational Research Association (AERA), American Psychological Association (APA) and the National Council on Measurement in Education (NCME) since 1989 (Brown, 2010). The instrument development investigations carried out within Rasch Measurement Theory (RMT) can be linked to the validity aspects of Messick as exemplified by different authors. For example, Bond (2003), Smith (2001), and Wolfe and Smith (2007) exemplified how the diverse outcomes from a Rasch analysis of the data can provide evidence to support validity arguments. Therefore, the Rasch approach to measurement development is a unified method that can address both validity and measurement issues.

Therefore, given that an instrument is developed within the principles of Rasch measurement theory, it can offer a variety of benefits for teachers. Assessments are an integral part of the teaching and learning cycle in any educational setting, which enable instructors to gather evidence and make judgements about student achievement. Similarly, it is a powerful tool to assess the quality of one's own instructional practices and provide feedback to overcome their learning gap. Several large scale empirical studies (Black & Wiliam, 1998; Lee, 2012; MOK, 2010; Narciss & Huth, 2004) have consistently demonstrated through providing meaningful diagnostic feedback how student assessments can inform and support further learning. Rasch based assessments could provide more fine-grained, evidence-based feedback about student performance allowing them to diagnose their learning progress, identify the gaps and link what they know to the scores achieved. Similarly, it provides information about the test such as reliability, and construct validity, which are all important to ensure the fairness of a test. Furthermore, given the items of the assessment fits the Rasch model requirements, interval level scaling of the data is allowed (Tennant & Conaghan, 2007). Interval level scores are more informative and parametric analysis can be performed on the scores to make valid predictions to improve many aspects of instructional practice.

In summary, there is an obvious need for developing CS1 measures based on stringent measurement theories including attention to validity issues and general instrument defensibility. Hence, a measure developed upon these foundations is indispensable to the research community as well as to the CS1 instructors. Its benefits are multi-fold: (a) to draw inferences about student competence; (b) as a dependant variable in CS1 research; (c) as a criterion variable to test the validity of other similar instruments; and, (d) to improve CS1 pedagogical practices.

1.3. Aims and Research Questions

As described in Chapter 1, the main aim of this research was to develop an objective measure of CS1 student competence commensurate with the principles of contemporary measurement validity theories. The research questions were:

1. Can a measure of student competency in CS1 be constructed?
2. What evidence is available to support an argument for the validity of the project?
3. Are there statistically significant associations between student competency in CS1 and student and classroom learning environment characteristics?
4. What are the consequences of the research for the design and delivery of CS1 instruction?

1.4. Significance of Study

Given the low level of student competence and the high attrition rate in CS1, it is important to investigate what affects student competence. However, a threat to further research remains with the paucity of tested instruments for measuring student competence. Typically, in most CS1 pedagogical research, the raw scores obtained from in-class tests and university exam scores are treated as measures of student competence. These evaluative tools are characterised by a lack of standardised scaling protocols, the absence of construct models to inform instrument design, and inconsistent selection of substantive content, which preclude measuring the true competence of students.

Unlike other disciplines in the human sciences, the CS1 research community so far has not explored the theoretical and practical benefits of invariant, interval-level measures derived by the application of modern measurement theories to improve instructional practice. This study is distinctive in that it is guided by an established measurement development model and underpinned by the principles of modern measurement theory. This study complies with the principles of the contemporary validity framework expounded by Messick (1989). The

anticipated outcome of the study is creation of a linear scale obtained by calibrating both the students and items on a common logit scale. This will enable direct comparisons of the students, in addition to being able to link student scores to items so that the scores have substantive meaning in terms of any underlying proficiencies. Such a measure can serve a variety of pedagogical needs such as the provision of data to improve instructional practices, drawing inferences about student competence and designing remedial work, and reporting performance to stakeholders in a meaningful way. Furthermore, Rasch-derived measures manifest interval-level properties on which parametric analyses can be justifiably performed without having to assume linearity or homogeneity of the scores. Therefore, its use can be extended beyond the classroom including: (a) as a dependant variable in CS1 research; (b) as a criterion variable to test the validity of other similar instruments; and, (c) as part of inferential studies to improve CS1 pedagogical practices. Therefore, the CS1 student competence measure (CS1 measure) is an addendum to the existing need for CS1 student competence evaluative tools for CS research in addition to being a valuable tool to assess classroom learning.

The correlational analysis is not the prime goal, but rather a secondary objective of this study. However, this will answer the question leading as to whether the choice of programming language influences the CS1 student competence, which is a heavily debated topic in CS1 literature. There has been no research so far which has examined these factors by utilising an interval-level scale as the dependent variable. Similarly, this study would also reveal several other factors that influence CS1 student competence. Therefore, the result of this study may benefit both students and educators by leading to a better understanding of the factors associated with student competence. Consequently, this information would be useful to inform designing learning support systems to remediate these factors. Furthermore, the conclusions established could potentially be used to improve student selection criteria for CS programs. Similarly, the outcome may also be vital for CS1 instructors and curriculum developers to reflect on their current teaching practices for making informed decisions with regard to improving instructional practice such as the choice of programming language for CS1 instruction.

1.5. Structure of Thesis

The thesis is organised into seven chapters, which provide the background to the research topic, an in-depth literature review of current CS1 measurement practices, a comparison of traditional and contemporary measurement models, the research methodology

and methods employed, and the results of the study. The following is a brief overview of these chapters:

Chapter 1 provides an overview of the study introducing the main problem with a focus on key concerns within the measurement practices of CS1 that threaten validity, and which eventually affect the outcome of investigations employing these measures. The chapter also outlines the research questions, aims and significance of this study to the existing scholarship of CS1.

Chapter 2 maps out the background of the study, focusing on the measurement concerns by describing issues with current measurement practices and the implication on validity and reliability of outcomes. It also discusses the available instruments for gauging CS1 student competence and their insufficiency by identifying the gaps and clarifying the focus of the overarching research objectives.

Chapter 3 reports the findings of the literature review on efforts to establish the construct of CS1 student competence. The chapter focuses on bringing together constituent elements of CS1 student competence constructs that are found within CS1 scholarship. The review concludes by proposing a theoretical model conceptualising the construct of CS1 student competence. The second part mainly focuses on a critical review of the relevant literature about the external factors that influence CS1 student competence.

Chapter 4 commences by reviewing models and frameworks related to the construction of measurements in the social sciences. The chapter is divided into three sections. The first section centers on the critical role of validity frameworks in measurement development leading to a brief outline of the developmental stages of validity. The second section presents a comprehensive critical review of two of the most commonly used measurement models for instrument development; thus giving the reader a clear indication of why certain deliberations were made pertaining to choice of measurement models used in the study. The chapter concludes with a brief review of the Rasch Measurement Theory.

Chapter 5 describes the methodology and methods of the study. The chapter begins by explaining the rationale for choosing certain methods for guiding the research. Next, the research aims and objectives are revisited before presenting the three main phases of the research: instrument development; validity evidence; and, the correlational analysis of factors associated with CS1 student competence.

Chapter 6 reports the findings of the three phases of the investigation. It is divided into three sections. The first section focuses on the results of the instrument development process and the

application of the Rasch analyses. The second section exemplifies Wolfe and Smith (2007a, 2007b) validity aspects utilising the results obtained in the instrument development process. Finally, the third section presents the results of correlating the factors associated with CS1 student competence with competence scores.

Chapter 7 discusses the key finding of the three phases in light of the literature. Then, each of the research questions is answered by drawing important conclusions from the discussions section and the results chapter in light of relevant theories and models. The chapter concludes by presenting the limitations and future direction of the study.

1.6. Summary

This chapter introduced the study with a brief overview of the motivation for undertaking this investigation including the significance of the study to CS1 body of knowledge. The next chapter discusses the background of this investigation, highlighting the main measurement concerns in assessment practices of CS1.

Chapter 2 – Background

2.1. Introduction

This chapter identifies the main issues in the assessment of CS1 students. The chapter begins with current assessment practices in CS1 in which the main concerns are reviewed followed by a critical evaluation of existing CS1 instruments. The evaluation compares the different types of measures available to measure CS1 student competence. The chapter concludes by summarising the main issues discussed.

2.2. Current Assessment Practices in CS1

The quality of measurements taken in a study is crucial for the production of defensible research outcomes. However, in the case of empirical research in the CS1 domain, the literature reveals that the majority of studies measuring CS1 student competence have a long tradition of relying on instruments that are not psychometrically sound. The common forms of evaluative tools mostly used as measures of student competence are university exam scores and researcher developed tools constructed for their specific purposes. Typically, psychometric properties are not reported and a stringent measurement model is not applied during development (See Lambert, 2015; Lister et al., 2004; McCracken et al., 2001; Owolabi, Olanipekun, & Iwerima, 2014; Zingaro & Porter, 2016).

Generally the summative tests constructed in schools and higher education institutes for measuring student learning do not follow the same standard procedures of psychological measurement construction. Mainly, the scoring is based on the number of individual items a student answered correctly irrespective of their difficulty levels. Perhaps, the reason could be the intended use of these tests are different from the psychological measures developed for research purposes. Most of the pedagogical research in the CS1 literature are conducted by educators, the same standard approaches of summative test construction appears to carry on the development of CS1 measurements for research purposes. This is evident in Table 2.1 which demonstrates the measures used in past CS1 research and their psychometric properties. As the table suggests, the majority of the papers did not even report most of the basic psychometric data, validity and reliability of the tests, of the instruments normally reported in other domains of research. Consequently, the outcomes of these measures are not appropriate to perform most of the parametric analysis involved in quantitative research studies as the outcomes are not of an interval level in addition to other issues prevalent in these measures.

Table 2.1

Psychometric Properties of Some of the CS1 Student Competence Measures

No	Study Name	Author(s)	Psychometric data	Measure
1	Mathematics Ability and Anxiety, Computer and Programming Anxieties, Age and Gender as Determinants of Achievement in Basic Programming	Owolabi, Olanipekun, & Iwerima, 2014	Not reported	Semester exam score
2	An Investigation of Potential Success Factors for an Introductory Model-Driven Programming Course	Bennedsen, & Caspersen, 2005	Not reported	Lab test
3	A Multi-National Study of Reading and Tracing Skills in Novice Programmers	Lister et al., 2004	reliability = 0.75	A test with 12 MCQ,
4	A multi-national, multi-institutional study of assessment of programming skills of first-year CS students	McCracken et al., 2001	An informal inter-rater reliability test on scoring	Three related programming tasks
5	Modelling programming performance: Beyond the influence of learner characteristics	Lau, & Yuen, 2011	Not reported	13 item (25 min test)

Content validity evidence is an important aspect of overall construct validity argument of a measure. It addresses the extent to which a pool of items adequately represent all the facets of the construct in question (Kimberlin & Winterstein, 2008). To establish the content facet of construct validity, it is important to focus on several aspects of the content representing the measure. These include the evidence of representativeness of the content, relevance, and the technical quality of the items representing the construct under investigation (Messick, 1989). However, in the CS1 domain, assessment development has been a very casual matter with little emphasis on the construct validity aspect. This is evident from the research conducted by Chinn et al. (2012) in which eleven academics from eight different universities in Australia and Finland were interviewed. The study revealed that the typical practice of writing exams was based on experience, inherited models, intuition, and pragmatics, without considering any pedagogical theories or validity frameworks. Another extensive review of 20 exam papers sourced from 10 institutions also revealed that there was no consistency among the universities on coverage of topics, question styles, skills required to answer questions and the level of

difficulty (Sheard et al., 2011). In another relevant study, a similar conclusion was drawn by Petersen et al. (2011) in their review of 15 final exam papers from various North American institutions. This study showed the majority of the exam questions required students to understand multiple programming concepts in each question prohibiting demonstration of the concepts they were familiar with. Likewise, Sheard et al. (2013) investigated the level of consistency in exam questions from different universities. These evidenced that the complexity level of exam questions among universities varied, thus masking the real differences in student programming competence. Therefore, in general, the content representativeness of CS1 measurement tools in general is arguably untenable.

Another threat to the validity of previous studies is the application of raw untransformed scores of tests without adherence to quantification requirements, particularly linearity. Most of the research completed in the past has assumed university exams scores or other forms of summed scores are reliable predictors of student knowledge. However, without considering item difficulties, assuming equal differences between pairs of raw scores is a fundamental error (Boone, 2016). In many cases, the researchers overlooked deficiencies in the measurement properties of the scores and proceeded to perform statistical analyses designed for interval data. There have been several studies reporting spurious effects when conducting complex statistical analysis of raw scores (Embretson, 1996a; Forrest & Andersen, 1986; Maxwell, Delaney, & Manheimer, 1985; Romanoski & Douglas, 2002). These studies identified faulty conclusions, induced errors and the undermining of significance. This has been demonstrated by Embretson (1996a), who specified several conditions under which interaction effects in factorial analysis (ANOVA) estimated from raw scores can be misleading and biased. Since such an interaction effect usually reflects the major research hypothesis, the inferences drawn are of questionable veracity.

Part of the problem of the questionable accuracy of student competency quantification also emanates from the measurement theory upon which most of the CS1 student competence evaluative tools have been traditionally tested and validated. Classical Test Theory (CTT), codified by Lord and Novick (1968); Novick (1966), is the predominant model for past CS1 assessment development. The theoretical foundation of the model and true-score theory, in general, is the basic formula $X = T + E$ in which the observed score is comprised of a true score (T) and an error (E). However, this does not specify under which circumstances X represents a measure, thus, the existence of a measure or of a level of measurement within the dataset can neither be justified nor be falsified (Salzberger, 1999). Another problem is distinguishing

between different or types of levels of measurement in data in order to select the statistical test that is most appropriate for the data set. The common assumption of CTT-based scales is that they produce interval-level measurement where unit increases in the scale are equal (Bond & Fox, 2015; Linacre, 2005). Advocates of Item Response Theory (IRT) models contend that summed scores resulting from the CTT based instruments do not manifest the characteristics of a true measure, rather they are simply counts which do not maintain requisite magnitude equivalence (Embretson & Hershberger, 1999). The common factor analytical procedure applied to most CTT-based measures, which is applied to demonstrate the quality of the measure, has also been criticised. The common belief that assessment tools that had gone through factor analytical process manifest linearity was refuted by Embretson and Hershberger (1999), who warned that "there is an incomplete relation between measurement construction and factor analysis" (p. 91). The fact is that none of the CTT procedures transform raw scores into interval-level scores (Embretson, 1993; Mullner, 2009). Consequently, reservations are expressed about the application of any form of linear statistical calculations designed for interval or ratio level scores on raw scores as it may result in erroneous results (Embretson, 1996a; Hambleton & Jones, 1993).

2.3. Existing CS1 Competency Instruments

When one endeavors to design a new instrument, it is important to review existing scales of a similar nature to identify present shortcomings, limitations and strengths that can be learned from (Wolfe & Smith, 2007a). The existing CS1 evaluative tools can be broadly categorised into three types. The first type consists of international qualification exams developed, tested, administered and owned by various national and international bodies. The second types of measurements are those developed by the CS education research community under research projects funded by well-known associations of the CS domain. The third category includes instruments developed by individual researchers or institutes. Each of these types has strengths and weaknesses as evaluative tools of CS1student competence.

AP Computer Science A (AP Comp Sci A) is one of the popular courses offered to high school students under the flagship of Advanced Placement (AP) examinations that is equivalent to CS1. It is registered under the trademark of the College Board and National Merit Scholarship Corporation, New York (AP Central, 2018). This course is offered by some high schools and colleges of America and across the world in addition to its availability to study online through virtual platforms through Khan Academy, Virtual High School, or a college or university (AP Central, 2018). A key feature of this course is that some colleges and

universities of America and universities in more than 60 countries recognize AP in the student admission process and grant students credit, placement, or both on the basis of successful AP exam scores (AP Central, 2018). A similar widely known exam - Cambridge International AS and A Level Computer Science (9608) is offered by the United Kingdom as a secondary school leaving examination (Cambridge International Examinations, 2018). Similar to AP courses, credit from these courses also normally carries credit towards CS1 study at college or university level. These exams have several advantages over university exams; however, they are not readily available for general use because they are subject to copyright, with testing administration controlled by their parent bodies.

The second types of measures are those developed by the CS research community usually as funded projects. The literature reveals few attempts at developing instruments for measuring various computing skills and overall competence of CS1 students. One well-known body of work was led by a working group (McCracken et al., 2001) of the Conference on Innovation and Technology in Computer Science Education (ITiCSE). The aim was to develop an instrument for assessing the programming skills of first-year students (McCracken et al., 2001). This study was comprehensive and focused on assessing the full range of CS1 topics and skills students should have learned at the conclusion of a typical CS1 course. The instrument was tested on multinational, multi-institutional (MNMI) level students. They also developed detailed scoring rubrics to guide scoring and increase consistency and accuracy among the markers when rating the student responses. Additionally, the students were given the choice of using the programming language they were most comfortable with to complete the programming tasks. However, there were accounts of criticism such as excessive mathematical flavor in some of the items, which may have caused these items to be measuring something other than programming ability (Decker, 2007; Lister, 2011). McCracken et al. (2001) in their review of the study acknowledged some of these flaws (as cited in Lister, 2011).

Similarly, an ITiCSE working group lead by Lister et al. (2004) conducted a follow-up study that resulted in an instrument being constructed for testing the reading and tracing skills of CS1 students. The instrument was based on a Multiple Choice Question (MCQ) only format. Well-constructed MCQ tests have several advantages, specifically in MNMI studies such as Lister et al.'s (2004) study. Firstly, with a MCQ format, no students will be disadvantaged by subjective bias when multiple instructors score the student responses (Stanger-Hall, 2012), resulting in higher levels reliability (Haladyna, Downing, & Rodriguez, 2002). Similarly, a MCQ format allows a breadth of sampling of any topic with less effort in a shorter time period

on larger samples (Haladyna et al., 2002; Stanger-Hall, 2012). Despite these facts, there are several drawbacks to the sole use of a MCQ in a test. Firstly, the MCQ format is good at assessing lower levels of student knowledge, however, it becomes progressively difficult to assess taxonomically higher-order cognitive processing such as interpretation, synthesis and application of knowledge (Case & Swanson, 1998). Likewise, another issue prevalent in employing a MCQ is that it encourages students to use guessing to get the correct answer, in addition to fostering recall (Becker & Johnston, 1999). Guessing has been identified as a major threat to the validity of a test score and can be a source for construct irrelevant variance (Royal & Hedgpeth, 2013). Furthermore, multiple choice questions can be unfair under certain conditions. For example, when MCQs are designed to assess high order thinking levels and a student selects a wrong answer, there is no way to give credit to the knowledge the student knows. Therefore, it is generally, agreed that the MCQ format alone should not be used as the sole assessment method in summative examinations (Al-Rukban, 2006).

The third type of measurement is the work undertaken by Ph.D. students. The literature search revealed two examples of which one was developed by Decker (2007) and the other was developed by Tew (2010). Decker (2007) developed and tested a short answer format test written in Java. It was tested in one institute, and its applicability limited only to Java programming language. In addition, its validity evidence could not be generalised beyond the institution level. In contrast, a major advantage of Tew's (2010) effort is that the instrument was written independently of any particular programming language using pseudocode, which had been tested on samples from different institutes instructed with a variety of programming languages. Moreover, a variety of metrics were applied to demonstrate evidence of validity. IRT was used to demonstrate the item quality, discrimination power, and extent of guessing.

However, despite the desirable properties of the aforementioned instruments compared to university exam scores, in principle, the resulting raw scores are not measures. While raw scores are necessary as input to achieve measures, construction of measures come between the collection of observation and analysis of measures developed from the observations as explained by Bode and Wright (1999). Wright (1999) asserts that before applying any linear statistical methods, linear measures need to be constructed from the observed data by applying a measurement model. Therefore, no matter how much effort is spent on these instruments, the direct observations would not result in an interval-level scale without the step for discovering the additive structure of quantity in the data (Michell, 1990, 1997). There has been a lot of debate in the measurement literature in regards to issues pertaining to summed scores being

used as measures. As warned by Bode and Wright (1999), the summed scores at best have only a rank order which is always biased in favor of central scores and against the scores at the extreme ends (Wright, 1999). This applies to all forms of raw scores resulting from dichotomous, partial credits and rating scale responses as well. The issue with raw scores is that the application of any linear statistical method like the analysis of variance, regression or factor analysis will produce systematically distorted results (Embretson, 1996a; Wright, 1999). However, this is not to imply that a measure is created simply with the transformation step, a true-interval-level measure additionally requires that the item development be grounded on a theoretical model of substantive knowledge and preferably, a developmental trajectory is included.

2.4. Summary

This chapter has identified several issues pertaining to the current assessment practices of CS1 and its implication on research outcomes. Furthermore, the chapter also reviewed the available choices of instruments and their inadequacies for measuring CS1 student competence underscoring the need for a CS1 evaluative tools to address the current measurement concerns. The following chapter examines the theoretical literature of CS1 to conceptualise the constituent elements underpinning the construct of CS1 student competence.

Chapter 3 – Literature Review: The Construct of CS1 Student Competence

3.1. Introduction

The chapter begins by explaining the primacy of a theoretical framework to establish the validity of the measure. Following this, a critical analysis of the literature is presented with the focus of uncovering concepts pertaining to the construct of CS1 student competence. The chapter then establishes the core skills or competencies expected to be learned at the conclusion of a typical CS1 course. Next, drawing from the conclusions already discussed, a conceptual model underpinning the construct of CS1 student competence will be inferred and illustrated as a visual model. The second part of this chapter is dedicated to identifying the factors, which have been cited in the CS1 literature to influence CS1 student competence.

3.2. Conceptualising the Construct of CS1 Student Competence

The current view of validity is an investigative process for providing evidence to support the intended uses and interpretation of scores of the measure (Messick, 1989; Wolfe & Smith, 2007a). Any kind of evidence about the test can contribute to the validity of the interpretation of the score meaning; however, the contribution becomes stronger if the degree of fit of the information with a theoretical rationale underlying score interpretation is explicitly evaluated (Messick, 1989). The connection between particular evidence and its uses and interpretations is made possible by a carefully laid out theoretical framework for the instrument (Wolfe & Smith, 2007a). The primacy of a theoretical framework to support instrument development has been made explicit by Messick on several occasions (Messick, 1989, 1995). Therefore, the purpose of this section is to review the empirical and theoretical literature on CS1 to conceptualise the construct of CS1 student competence.

3.2.1. Computer Science curricula 2013

The computer science education community has generated and suggested curricula since 1968 with the most recent edition of ACM/IEEE-CS Computer Science Curricula 2013 (CS2013) matching the latest developments in the discipline. This is the most widely accepted CS undergraduate degree curriculum development framework and is used by universities across the world as a benchmark to design and evaluate the quality of CS degree programs.

Comparable differences were observed in CS2013 as compared to earlier versions of CS curriculum frameworks (ACM/IEEE-CS CS2001 and ACM/IEEE-CS CS2008). The new framework allows institutions to customize CS curricula to suit for their specific CS degree

program needs. Similarly, unlike previous versions, the guiding principles, and the organisation of CS2013 CS's body of knowledge is substantially different, whereby, there is no exact mapping for Programming Fundamentals (previous CS1 curriculum suggested in CS2001) in CS2013. Alternatively, the curriculum encompasses Knowledge Areas (KA) organised by themes called Knowledge Units (KU) and grouped using a three-tiered classification scheme. All the Core-Tier1 KU's are a compulsory part of all CS programs, which are typically covered in the introductory courses. Whereas Core-Tier2 and Core-Tier3 are more advanced level topics which build upon the Core-Tier1 concepts in which 80% of Core-Tier2 must be covered; Core-Tier3 are electives to mix and match the needs of a variety of different CS programs. In CS2013, Programming Fundamentals is one of the four KU's of Software Development Fundamentals (SDF) KA (see CS2013 for more details), which is classified as Core-Tier1. The KAs or KUs by themselves are not courses, thus, KU's can be fleshed out and customised in novel ways by combining other KU's or materials outside the scope of the KAs to suit the particular need of a course. The Programming Fundamentals' KUs identify all the foundational concepts that are common to all programming paradigms; this can serve as the skeleton of a typical CS1 course. Table 3.1 shows these concepts.

Table 3.1

Fundamental Programming Concepts

#	Concept
1	Basic syntax and semantics of a higher-level language
2	Variables and primitive data types (e.g., numbers, characters, Booleans)
4	Expressions and assignments
5	Simple I/O including file I/O
6	Conditional and iterative control structures
7	Functions and parameter passing
8	The concept of recursion

3.2.2. Other scholarly work

When new instruments are developed numerous "measures" of student competency for a variety of purposes are available. These include studying correlates of CS1 student competency (Alvarado, Lee, & Gillespie, 2014) and assessing programming ability (Lister et al., 2004; McCracken et al., 2001). Bergin and Reilly (2006) listed more than 19 studies focused on factors associated with student competency. One notable aspect of these studies is

the use of evaluative tools such as laboratory tests, final exam scores or final composite scores as the dependent variable. Similarly, despite the strengths of popular multi-institutional and multi-national student ability assessment studies (Lister et al., 2004; Lopez et al., 2008; McCracken et al., 2001) as compared to common classroom test scores, these tools show a lack of clear evidence of content validity. Therefore, these instruments do not provide a clear framework to specify what concepts constitute a typical CS1 course.

The most extensive study consistent with the goal of the current study was found to be the work of Tew and Guzdial (2010), which was part of the development of a language-independent CS1 student competency measure. Unlike other similar studies, the merit of this study was the application of validity standards (the American Educational Research Association, the American Psychological Association, and the National Council on Measurement in Education and the Joint Committee on Standards for Educational Psychological Testing, 1999) in the instrument development process. There are a number of other reasons as to why Tew and Guzdial's work is suitable as a starting point for a study similar in nature: Firstly, the congruent goal of developing a CS1 student competency measure that is widely applicable, irrespective of instructional paradigms or the language choice for instruction; secondly, the theoretical model or the conceptual content underpinning the instrument has been well documented and published; and, thirdly, the authors validated their conceptual framework by multiple methods. These include document analysis of the most widely used CS1 books for curriculum instruction in universities, and conducted an expert review to evaluate the relevance and representatives of the conceptual framework to the construct domain. As well as this they also scoped the contents according to ACM/IEEE-CS CS2001 curriculum guidelines (The Joint Task Force on Computing Curricula, 2001). Their list of CS1 concepts conforms to the Fundamental Programming Concepts KU of CS2013 with the exception of the Input/output concept (I/O). I/O was part of their initial list, however, later it was removed as it was considered difficult to test in a language-independent manner because I/O routines are tied to the specific programming languages. Therefore, the conceptual framework underpinning the Tew and Guzdial (2010) instrument was chosen as the initial set of constructs for the current study. The topics also reflect the content suggested by CS2013 in general (See Table 3.1 and Table 3.2 for a comparison) with the exception of Object-oriented basics.

To promote applicability and validity, the conceptual framework of Tew and Guzdial (2010) was further compared and benchmarked with other sources of information (Caceffo,

Wolfman, Booth, & Azevedo, 2016; Lee & Ko, 2015; Petersen et al., 2011; Stephenson & West, 1998). For example, the initial list was compared with the CS1 curricula of Maldives National University (MNU) and Villa College of Maldives in addition to Asia Pacific University (APU) and Malaysia and Royal University of Bhutan (see Table 3.2). The first column represents the constructs of the Tew and Gudzial study with related concepts combined; the other columns represent concept significance based on curricula review and feedback from the students and instructors of each institute. "X" denotes adequate coverage of the construct based on the sources used.

Table 3.2

Summary of Topic Coverage Comparison with the Tew and Guzdial Conceptual Framework

Concepts	APU (Python)	MNU(C++)	Villa college (Java)	Royal University of Bhutan (C)
Fundamentals (variables, assignment, etc.)	X	X	X	X
Logical Operators	X	X	X	X
Selection Statement (if/else) (subsumes operators)	X	X	X	X
Loops (subsumes operators)	X	X	X	X
Arrays	X (lists instead of arrays)	X	X	X
Methods (includes functions, parameters, procedures, and subroutines)	X	X	X	X

Recursion	Not covered	Not in detail Not covered in the exam	Not covered	Not covered
Object-oriented basics	Not covered	Not in detail Not covered in the exam	X	Not covered

Previous studies (Caceffo et al., 2016; Hertz, 2010; Lee & Ko, 2015; Petersen et al., 2011; Sheard et al., 2011; Stephenson & West, 1998) show greater consistency about coverage of the first five concepts of Table. 3.2. The relatively light coverage of the concept of relational and logical operators in these studies is perhaps due to the concept being subsumed by the control structures (specifically, selection and loop). Therefore, combining logical operators with control structures in CS1 delivery is normal practice. The inconsistencies revealed in the coverage of Object-Oriented (OO) concepts are likely the result of the concept being connected to the OO paradigm, which is unlikely to be covered in other programming paradigms. While the topic Recursion is a popular concept often taught in CS1 courses and is one of the topics of the Programming Fundamentals KU, studies indicate that the topic is either not covered or tested in the exams. A partial explanation is that traditionally it was seen as an advanced topic often continued in the Computer Science (CS2) sequence. Taking these arguments into consideration, the inclusion of such concepts may impede achieving the goal of wider instrument applicability, and therefore the topics of OO Basics and Recursion were not included as fundamental concepts of programming. Therefore, considering the goal of wider applicability and generalisability as the main concern, the following topics (See Table 3.3) will constitute the construct of CS1 student competence.

Table 3.3

Fundamental Programming Concepts

#	Concept
1	Fundamentals (variables, assignment, etc.)
2	Selection Statement (if/else) (subsumes logical operators)
3	Loops (subsumes logical operators)
4	Methods (includes functions, parameters, procedures, and subroutines)
5	Arrays

3.2.3. Core computer programming literacy skills

The major focus of any CS1 curriculum obviously is learning to write program. However, reaching this goal requires a number of essential individual programming skills - tracing, reading and writing (See Herman, Salam, & Noersasongko, 2011; Lakanen, Lappalainen, & Isomöttönen, 2015; Lister et al., 2004; McCracken et al., 2001; Yamamoto, Sekiya, Mori, & Yamaguchi, 2012). Hence, the core objectives of the course are set around these skills, which 'in turn' reflects in the learning activities students engage in within CS1 classrooms, which eventually becomes the basis for assessment of student learning. Thus, it is important to explore these skills.

One way to inform what essential skills are being covered in a typical CS1 course is by examining the different assessment tasks across CSI courses; this is because assessments are supposed to reflect what learners have engaged in whilst learning the curricular content and achieving the learning outcomes (Chudowsky, Glaser, & Pellegrino, 2001). Several studies have examined CS1 exam content, which can provide an insight into the nature and range of skills students are required to demonstrate in their CS1 courses. For example, in an effort to develop a classification scheme to establish content and the nature of CS1 exam questions, Sheard et al. (2011) found that approximately 81% of the questions related to code writing, tracing, explaining, debugging, and modifying, with the highest emphasis on writing followed by code tracing and explaining. Only a very small percent of items were dedicated to less common skills such as program design. Similarly, a study which examined the exam content of fifteen exams from fourteen schools of North American institutions revealed four types of questions with program writing and reading being the most common (Petersen et al., 2011). Another study aimed at the development and testing of a benchmarking tool to compare learning outcomes of introductory programming students across courses, institutions, and countries, mainly tested four skills – tracing, reading, modifying and writing code (Sheard, Dermoudy, D'Souza, Hu, & Parsons, 2014). These questions were sourced from the exam papers of five institutions based in Australia and New Zealand as part of the workshop held in conjunction with the Australasian Computing Education conference in 2013 (ACE2013). Likewise, a large corpus of studies reported in the CS1 literature revealed that the main skills measured were typically tracing, reading, modifying and writing code (See Herman et al., 2011; Lakanen et al., 2015; Lister et al., 2004; McCracken et al., 2001; Yamamoto et al., 2012).

Another way to assay this evidence is to examine chapter-end exercises of popular introductory programming textbooks covering in the early CS1 concepts. These exercises will

give an indicator of the range of skills students are expected to learn in a fundamental computer programming course such as CS1. To identify these skills, the programming tasks of chapter-end exercises of five different introductory programming books (Deitel & Deitel, 2010; Hubbard, 1999; Johnson, 2012; Kochan, 2015; Streib & Soma, 2014) were reviewed. Figure 3.1 presents the different types of programming tasks and their total numbers as a percentage based on one chapter (loop structure). Despite the popularity of code explaining skills, none of these books had covered the skill. The typical skills are the same as discussed before, tracing, debugging, modifying and writing code with writing being the most popular, except for code reading. Debugging and modifying code typically involves code reading and writing, and code reading was not explicitly covered in these text books.

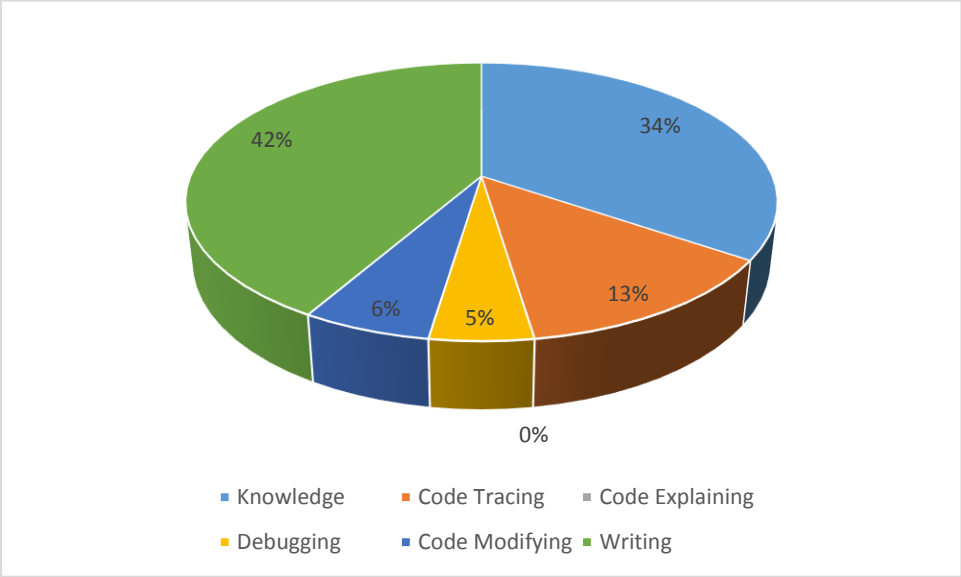


Figure 3.1. Different computer programming tasks found in Introductory CS books.

Another significant source of information, which could elucidate the expected skills in a typical CS1 curriculum, is the CS2013 curriculum framework, which was detailed at the beginning of this section. It is helpful because the framework not only provides the core topics the students are expected learn in a typical CS1 course, but also provides descriptive learning outcomes for each topic of the KU. Although the descriptors of the learning outcomes may not have an exact match with the outcome descriptors of each individual institute, the ACM curriculum guidelines expect the institutes to encapsulate the same level of mastery and intent for a given concept when individual institutes' design their own curriculum. Each learning outcome is also associated with one of the three mastery levels – familiarity, usage or

assessment – where familiarity is the lowest and assessment is the highest level. Table 3.4 shows the descriptors of the learning outcomes and their associated mastery levels. The last column, added by the researcher, shows the type of programming tasks predominant in the descriptor. The skill type was determined by the keywords used in the learning outcomes descriptors and the typical tasks students are expected to engage with in each of these descriptors. For example, the first descriptor expects students to analyse (read with understanding suggests two skills – tracing and reading skills) a given code segment and then explain the behaviour (function) of the code.

Table 3.4

Descriptors of the Learning Outcomes and Their Associated Mastery Levels

Learning Outcomes	Mastery Level	Skills
Analyse and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, conditional and iterative control constructs, functions, parameter passing and recursion.	Assessment	Analyse (Trace+ Read) Explain (Read)
Identify and describe uses of primitive data types	Familiarity	Knowledge
Write programs that use primitive data types	Usage	Write
Modify and expand short programs that use standard conditional and iterative control structures and functions.	Usage	Modify (Read +Write) Expand (Read +Write)
Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, arrays, the definition of functions, and parameter passing	Usage	Implement (Write) Test (Trace+ Read +Write) Debug (Trace Read + Write)
Choose appropriate conditional and iteration constructs for a given programming task	Usage	Knowledge

Similar to the skills identified from other sources presented here, the keywords provided by the CS2013 curriculum framework descriptors also suggest that the students are expected to acquire a similar range of skills in curriculums that accord with the framework. For example, the second outcome emphasised writing, modifying and extending, where modifying and extending basically are combination of code reading and writing skills, because one cannot modify or extend (both involves writing) without being able to read or comprehend the

behaviour of the code segment. Additionally, students are expected to extend their understanding of each of these concepts from familiarity to assessment level at the conclusion of the course according to CS2013 curriculum framework.

The CS1 literature review also concedes the importance of the programming skills advanced in the previous discussion to achieve CS1 curriculum goals (Corney et al., 2014; Lister et al., 2004; Lopez et al., 2008; McCracken et al., 2001). However, a particular focus was placed on three of the essential programming skills: code-tracing– checking the steps in a program's execution; code-explaining (reading) – stating the overall purpose of a piece of code; and, code-writing – translating the problem solution into actual programming language code similar to the CS2013 curriculum framework. These skills have been recognised as fundamental in both current (Corney et al., 2014; Harrington & Cheng, 2018; Lister et al., 2004; McCracken et al., 2001) and older literature (Perkins & Martin, 1986; Soloway, 1986), and in the Building Research in Australasian Computing Education (BRACE) examination framework. A substantial body of empirical research, particularly BRACElet publications (a series of multi-institutional Computer Science education research studies of novice programmers initiated by BRACE), explores the learning process of these skills utilising the Structure of the Observed Learning Outcome taxonomy (SOLO) (Collis & Biggs, 1982), and produces empirical evidence that conceptually links the skills of tracing, explaining, and writing code.

Although there is no conclusive evidence of a strict hierarchy between tracing, explaining, and writing code, a casual hierarchical relationship exists between these skills as reported in the CS1 literature. Empirical evidence does confirm the combined effects of tracing and explaining in accounting for substantial variation in writing ability (Lister, Fidge, & Teague, 2009; Lopez et al., 2008; Venables, Tan, & Lister, 2009). Lopez et al. (2008) used linear stepwise regression to understand the learning hierarchy of these programming skills. Their results demonstrated the evidence of a learning path between the skills of computer code tracing, reading, and writing. The study found that the students' knowledge of programming constructs form the bottom of the learning hierarchy, whilst code tracing and explaining form the intermediate skills, with code writing skills at the top of the learning hierarchy. Similarly, two main follow-up BRACElet studies (Lister et al., 2009; Venables et al., 2009) correspondingly corroborated evidence of skills hierarchy demonstrated by Lopez et al. (2008). In sum, the BRACElet picture of the early development of programming postulates that firstly the novice acquires the ability to trace code. The ability to explain code develops

when the ability to trace becomes stable. Finally, the systematically writing code emerges after a reasonable development of both tracing and explaining. Accordingly, the construct model of early programming skills development embodied in the current study is made explicit by depicting the progression of skills as shown in Figure 3.2.

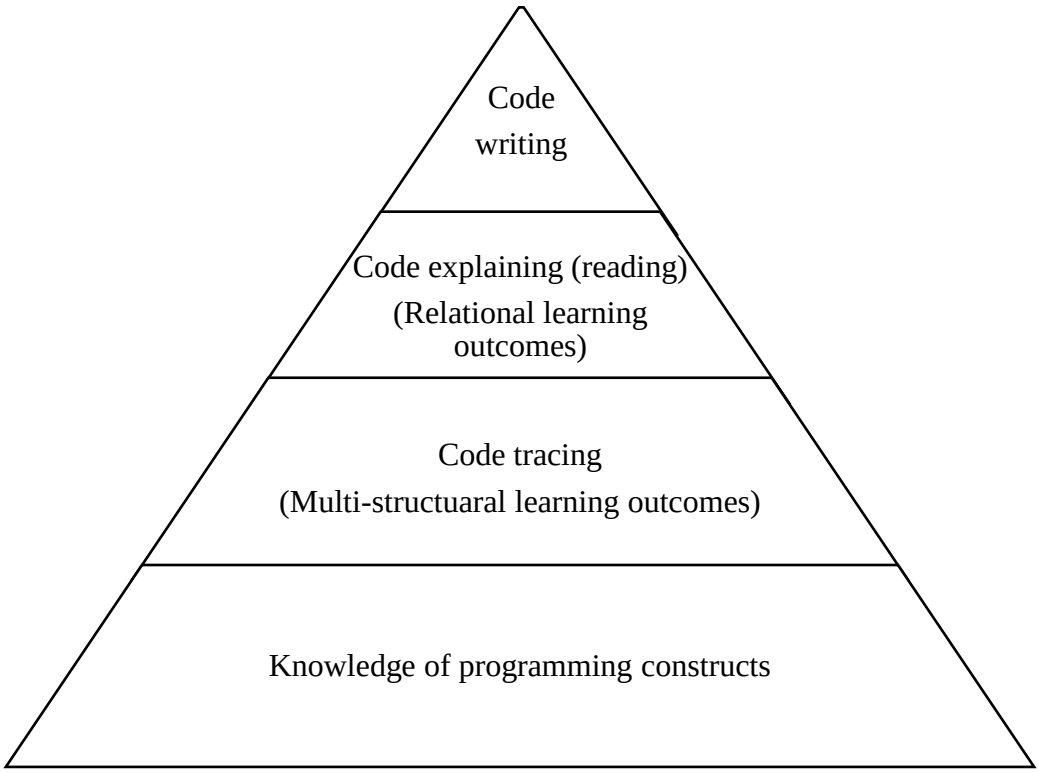


Figure 3.2. Difficulty hierarchy of the three main programming skills.

3.2.4. Models of human conceptualisation

A construct model provides motivation and structure for the construct to be measured in the form of hierarchical statements (Wilson, 2004). Computer Science educators have tried to apply models and taxonomies of human conceptualisation in evaluating student understanding of programming concepts and how their performance grows in complexity when mastering the knowledge (Fuller et al., 2007). These models serve multiple roles such as assessing the students' achievement against the intended learning outcomes (Thompson, Luxton-Reilly, Whalley, Hu, & Robbins, 2008), as a framework to design curriculum and assessment tasks (Johnson & Fuller, 2006; Scott, 2003). They are also employed as a tool for communicating student achievement and student progression of curricula (Boulton-Lewis, 1994). The two most prevalent models applied to date for these purposes are the SOLO

taxonomy (Biggs & Collis, 1982) and Bloom's Taxonomies of Education Objectives (Bloom, 1956).

Bloom's taxonomy has a history of over fifty years and is a familiar tool for educators (Pickard, 2007). Bloom's original model consisted of six levels, with the first three levels (knowledge, comprehension, and application) being hierarchical in nature, whereas, the upper three levels (analysis, synthesis, and evaluation) are more parallel according to the opinions of Anderson and Krathwohl (2001). Some opined that Blooms taxonomy classification was not a properly constructed taxonomy, as it lacked a systemic rationale of construction (Morshead, 1965). This was subsequently acknowledged and the model was later re-established by Anderson and Krathwohl (2001) along more systematic lines. Bloom's model has been applied for a variety purposes in the CS1 domain including course design and evaluation (Lister, 2001); evaluating difficulty ratings of courses offered in CS degrees (Oliver, Dobele, Greber, & Roberts, 2004); design assessment tasks for different ability levels (Lister & Leaney, 2003); and, improving instruction (Alaoutinen & Smolander, 2010; Whalley et al., 2006). While this model was a useful framework for a variety of CS1 educational purposes, there has also been several precedents in which the framework was found to be problematic to apply in designing and evaluating computer programming tasks (Fuller et al., 2007; Shuhidan, Hamilton, & D'Souza, 2009; Thompson et al., 2008). One issue raised by Fuller et al. (2007) was that they found it challenging in mapping the levels of Bloom's with CS1 assessment tasks. Another issue contended by Gluga, Kay, Lister, Kleitman, and Lever (2012a) was that it requires a deeper understanding of the learning context for accurate classification and that it is often difficult to reach consensus on the interpretation of the levels. The classification issue is not only prevalent among computer science educators. There have been studies demonstrating that even educators of other disciplines who were well trained in the use of Bloom's taxonomy were not able to match the assessment tasks with Bloom's taxonomy levels consistently (See Fairbrother, 1975; Stanley & Bolton, 1957).

Some critiques argue that the framework is difficult to apply in some educational contexts because the posited existence of a cumulative hierarchy separating one cognitive level from the other levels does not hold together as assumed in the model (Paul & Binker, 1993; Sugrue, 2002). The assumption of the cumulative hierarchy of cognitive levels has been investigated in earlier literature; however, these studies found no empirical evidence supporting the premise (see Kreitzer & Madaus, 1994; Kropp, 1966). A long array of CS1 literature reports (Gluga, Kay, Lister, Kleitman, & Lever, 2012b; Starr, Manaris, & Stalvey, 2008; Whalley et

al., 2006) efforts to adopt Bloom's taxonomy to design and evaluate computer programming assessments tasks. However, according to a systematic review of the use of Bloom's taxonomy in Computer Science Education, several studies reported difficulty in applying the taxonomy in assessment tasks consistently (Masapanta-Carrión & Velázquez-Iturbide, 2018).

Unlike Bloom's taxonomy, Biggs' SOLO (Collis & Biggs, 1982) model is grounded on a theory of learning and teaching supported by research on the student learning process (Biggs & Tang, 2011). There are several intriguing characteristics that make SOLO more appealing than Bloom's taxonomy. Firstly SOLO is a theory about teaching and learning based on research on student learning unlike Bloom's taxonomy which is a theory about knowledge based on the judgements of educational administrators (Biggs & Collis, 1982). Bloom is more suitable for setting learning objectives, whereas SOLO is suitable for setting learning objectives as well as assessing the learning process of students, which could be cognitive, performative, effective or a combination of these (Potter & Kustra, 2012). Unlike the expectation of the Bloom's model, where there must be an exact correspondence between the task and the outcome, SOLO taxonomy does not impose such a relationship, consequently, a task can be designed to assess students of varying abilities or just to elicit a particular SOLO level (Whalley & Kasto, 2013). Most importantly, unlike Bloom's taxonomy SOLO does not see knowledge and the intellectual process as separate entities, rather the model sees them as an integrated system of elements which focuses on the learning process where knowledge is inferred across all levels (Whalley & Kasto, 2013).

Recently there has been a growing interest by CS educators to study the SOLO taxonomy with respect to assessing fundamental programming skills (Ginat & Menashe, 2015; Izu, Weerasinghe, & Pope, 2016; Lister, Simon, Thompson, Whalley, & Prasad, 2006; Sheard et al., 2008; Whalley, Clear, Robbins, & Thompson, 2011). Perhaps there might be a commonality between the taxonomy and the programming skills hierarchy as postulated in Figure 3.2 above. Of the five levels – prestructural, unistructural, multistructural, relational and extended abstract – the lowest three levels are regarded as quantitative (with respect to the element details) and the two higher levels are more qualitative (with an emphasis on integration and notion of the coherent whole) (Potter & Kustra, 2012). This shows that learning passes through various stages from quantitative to more qualitative approaches; however each level above manifests an increased understanding of knowledge and a higher level of abstraction (Whalley et al., 2011). A similar pattern of increased understanding can be seen within the core computer programming skills. For example, skills such as tracing are more quantitative in

nature and are lower in the hierarchy of knowledge acquisition than explaining and writing (See Figure 3.2). The latter two skills are qualitative levels that are of a higher order than tracing. These two skills require students to derive programming solutions or show understanding of written code, which integrates multiple concepts in an interleaved manner forming a logical whole. There have been several studies demonstrating the reliability of SOLO taxonomy in assessing various programming skills (Clear et al., 2008; Ginat & Menashe, 2015; Izu et al., 2016). Table 3.5 shows Clear et al.'s (2008) definition of these levels for code writing tasks. Similarly, there has been research adopting the model to assess other skills such as tracing and explaining code (Lister et al., 2006; Sheard et al., 2008). The majority of the research on code explaining is shown to be assessed at the Relational level (See Lister et al., 2006; Sheard et al., 2008; Whalley et al., 2006), and writing assessed at the extended abstract level (Clear et al., 2008).

Table 3.5

Clear et al.'s (2008) SOLO Categories for Code Writing

Phase	SOLO Category	Description
Qualitative	Extended Abstract-Extending	Used constructs and concepts beyond those required in the exercise to provide an improved solution
	Relational - Encompassing	Provides a valid well structure program that removes all redundancy and has a clear logical structure. The specifications have been integrated to form a logical whole
Quantitative	Multistructural - Refinement	Represents a translation that is close to direct translation. The code may have been re-ordered to make a valid solution
	Unistructural – Direct Translation	Represents a direct translation of the specifications The code will be in the sequence of the specification
	Prestructural	Substantially lacks knowledge of programming constructs is unrelated to the question

More recently, a model developed specifically to assess computer programming code called "Block Model" was proposed by Schulte (2008). Although it was a substantial contribution to the CS body of knowledge, the model did not gain momentum within the research community. A keyword search of the ACM digital database showed only one paper

(Whalley & Kasto, 2013) that attempted to apply the model to assess computer programming codes.

In summary, there are two established models of human conception generally applied in pedagogical contexts. Bloom's taxonomy is seen as a set of principles rather than a theoretical model of human conception like SOLO. Bloom's taxonomy was found to be difficult to apply and interpret in some educational contexts while SOLO was more highly regarded for use in objective settings as well as evaluating the assessment tasks. Specifically, there has been substantial CS1 research literature demonstrating SOLO taxonomy can be reliably used to assess the assessment tasks of CS1 in general.

3.2.5. A construct model of CS1 student competence

A construct model provides motivation and structure for the construct to be measured and usually is in the form of hierarchical statements (Wilson, 2004). Based on the equivalence between the SOLO levels and established hierarchy of programming skills, a generic construct model for assessing each topic of the CS1 student competence construct is presented in Figure 3.3. The vertical axis represents the hierarchy of programming skills whereas the horizontal axis represents SOLO levels. The diagonal line suggests a relation between the programming skills variable (four skills), and the SOLO levels variable (four levels). The diagonal line also provides the coordinates (programming skills, SOLO level) for plotting each of the five overarching CS1 concepts on the Cartesian plane. The model portrays postulated associations between three hierarchically structured variables.

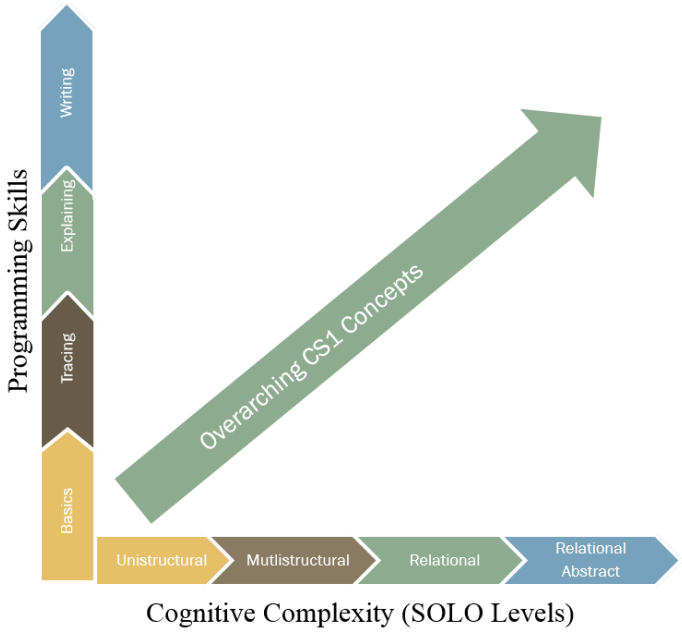


Figure 3.3. Proposed generic construct model of CS1 student competence

In summary, the review of the CS1 literature suggests that CS1 student competence can be operationalised to constitute five fundamental topics irrespective of programming language or the paradigm used for CS1 instruction. The learning objectives defining the expected goal of a CS1 curriculum embodied three important precursor skills – tracing, reading (explain) and writing – for learning to program, which manifests to form a hierarchy in that order. Models of human conception have long been used to guide the assessment tasks of other human science domains. Consequently, the literature shows that both Bloom's and SOLO taxonomy have been applied widely in the design and assessment of student tasks in CS1. However, as suggested by the CSI literature, the SOLO taxonomy learning hierarchy is found to align well with the expected skill progression of CS1 students. Therefore, SOLO taxonomy levels were used in this study to assess the progression of these key skills as the students learn the CS1 topics.

3.3. Factors Associated with Student CS1 Competence

The previous section presented the internal model of the CS1 student competence variable illustrating the constituent elements and their relationships. Similarly, there are several external variables relevant to CS1 student competence. The two major categories of variables shown to be of influence are: (a) individual or student factors, or characteristics of an individual that might play a role in CS1 student competence; and (b) environmental factors (internal and external), or characteristics of the learning environment. These factors contribute to the study in several ways: (a) the factors will be used to examine item bias (Differential Item Functioning) – a test performed in Rasch analysis to support invariance of the instrument across different demographics of the sample; (b) support some aspects of the validity argument of the investigation; and, (c) to test the association between CS1 student competence. Therefore, this section discusses empirical literature on the associations between these factors and CS1 student competence.

3.3.1. Student factors

A central focus of computer science education research is improving CS1 instructional practices. Research into novice programmers spans more than thirty years and has examined a wide range of factors associated with CS1 student competence (Robins, 2010). The CS1 literature suggests an assortment of factors that account for variation in CS1 student competence including prior mathematics performance, previous programming experience, gender, and previous study of science.

The relationship between previous mathematics performance and achievement in CS1 has been confirmed in earlier studies (Bergin & Reilly, 2006; Evans & Simkin, 1989; Jerkins, Stenger, Stovall, & Jenkins, 2013; Lambert, 2015; Leeper & Silver, 1982). For example, White (2003) investigated the correlation between mathematical proficiency and competence in visual programming, and procedural programming (Ott, 1988). White (2003) correlated freshmen mathematics scores and American College Testing and Scholastic Aptitude Test (ACT/SAT) scores with introductory visual programming course scores. The study revealed that both freshmen mathematics scores and ACT/SAT mathematics scores were positively correlated with student competence in CS1. A similar conclusion was drawn by Lambert (2015) in a study of multiple factors tested for correlation with CS1 student competence, which had shown only programming experience and mathematics scores were the predictors of CS1 success. Research also shows that student failure in CS1 is normally attributable to poor mathematical skills. For example, Gomes, Carmo, Bigotte, and Mendes (2006) investigated the link between students who failed in CS1 and their mathematical ability. Many of the students who failed also lacked mathematical competency in one or more areas of the curriculum.

Mathematical problem-solving ability is often advanced as influential on CS1 student competence. Computer programming requires more domain-specific mathematical problem-solving skills than general problem-solving skills (Gomes et al., 2006). According to (Shneiderman & Mayer, 1979), computer programming requires semantic and syntactic knowledge. The semantic knowledge is needed to comprehend, design and formulate a solution that is translatable to a computer program. Syntactic knowledge is necessary to translate the semantic solution into programming code (Nowaczyk, 1984); semantic knowledge, which rests upon the ability to problem-solve, is a pre-requisite for syntactic knowledge. Nowaczyk (1984) examined the relationship between mathematical problem-solving ability and student competence, focusing on areas of logical operations, algebraic solutions, transformations and mathematical relationships. He concluded that individual differences in semantic knowledge were related to CS1 student competence. This finding is similar to that of a previous study by Kurtz (1980). Pillay and Jugoo (2005) also studied the effect of problem-solving ability on the competence of novice programmers, reporting a positive correlation between competence and problem-solving ability.

Previous programming experience is one of the commonly tested variables for correlation with CS1 student competence (Bergin & Reilly, 2006; Hagan & Markham, 2000; Strnad, Šerbec, & Rugelj, 2009; Wiedenbeck, 2005). Studies have revealed either directly or

indirectly that previous programming experience is associated with CS1 student competence. Hagan and Markham (2000) investigated the direct association between previous programming experience and student competence. They reported that students with prior programming experience performed better than students without prior programming experience. They also found that competence was consistently related to the number of programming languages students were familiar with. Similarly, Lambert (2015) tested several factors for correlation with student competence, and found that the mathematics ability and programming experience were the only factors having a significant correlation. In terms of indirect associations, Wiedenbeck (2005) established that student self-efficacy was influenced by previous programming experience. Similarly, a corpus of studies has shown that problem-solving ability is influenced by previous programming experience (Tu & Johnson, 1990).

Although student self-efficacy has been associated with academic achievement (Brosnan, 1998; Eachus & Cassidy, 1997; Parker, Marsh, Ciarrochi, Marshall, & Abduljabbar, 2013), within the domain of CS, very few studies have investigated this important relationship. An investigation carried out by Wiedenbeck (2005) on 75 undergraduate students enrolled in CS1 from different academic disciplines hypothesised a model of factors affecting student competence in CS1. Students' mental model of programming (a mental representation of real-world objects or systems) was influenced by self-efficacy, and both mental model and self-efficacy were positively correlated with student competence. Similarly, Wiedenbeck (2005) examined the combined effects of previous programming experience, perceived self-efficacy, and knowledge organisation on CS1 student competence. The study revealed that the previous programming impacted perceived self-efficacy, and this, in turn, was associated with CS1 student competence.

A significant body of CS1 research indicates underrepresentation of women in CS and related fields (Chao & Henderson, 2012; Wang, Hong, Ravitz, & Ivory, 2015). Wang et al. (2015) reported that approximately 20% of students admitted to the computer science courses were females. Nonetheless, evidence concerning gender differences in CS1 student competence remains equivocal. For instance, some studies showed that males perform better than females (Goold & Rimmer, 2000; Nowaczyk, 1983; Owolabi et al., 2014), while other studies (Byrne & Lyons, 2001; Pillay & Jugoo, 2005) revealed that student competence in CS1 was not gender-dependent. Alternatively, Byrne and Lyons (2001) reported that although it was not significant, overall the females in their study performed better than the males. They

further noted that in regard to previous programming experience the females outperformed the males.

There are not usually prerequisites for admission into CS and its related bachelor's degree programs, unlike other disciplines such as engineering (Boyle, Carter, & Clark, 2002). While it is debatable whether CS falls into the domains of science or engineering (Loui, 1995), CS admission requirements are not comparable to either of these disciplines. As a result, students from a variety of backgrounds including science, business, and the arts streams enroll in CS programs. Significantly, student competence in science subjects has shown correlation with student competence in CS1. Bergin and Reilly (2006), demonstrated this relationship in a study that tested fifteen factors for correlation with CS1 student competence. The findings of the study revealed that science subjects, in general, had a significant influence on competence. Another study carried out by Rountree, Rountree, and Robins (2002) tested students from different disciplines of study for correlation with student competence. They reported that students with a humanities background scored lower than other backgrounds. However, there have not been many empirical studies linking these two variables.

3.3.2. Learning environment factors

Besides student characteristics, some researchers have also shown that the learning environment can be related to poor student competence in CS1(Moons & De Backer, 2013). An extensive literature search of teaching practices in CS1 instruction (See Pears et al., 2007) revealed a rich corpus of studies weighing up the learning environment factors of student competence. The factors include the programming paradigm, programming language, programming environment and assessment and feedback strategies. There is a huge array of research and debate around these variables to reform CS1 classrooms and improve instructional practice.

One approach to improving student learning is to focus on a particular programming education methodology or approach (Moons & De Backer, 2013). The main approaches used for CS1 curriculum instruction are Functional-First (PF) and Objects-First (OF). ACM/IEEE-CS Computer Science Curricula 2001 defines the PF approach as a course that "introduces algorithmic concepts in a language with a simple functional syntax, such as Scheme", whereas, the OF approach as a course that "emphasises the principles of object-oriented programming and design from the very beginning" and that "begins immediately with the notions of objects and inheritance" (Roberts & Engel, 2001). Bailie, Courtney, Murray, Schiaffino, and Tuohy

(2003) shared their experiences in switching the CS1 instructional paradigm to an OF approach. While many reported positive effects, some authors also expressed their dissatisfaction and their students struggle to learn the OF approach material. One of the authors recounted that the OF approach helps with teaching the important elements of the computer programming process, including problem analysis and the structuring of solutions in a natural manner. Furthermore, studies have shown that students with a PF background have difficulty in transferring their programming knowledge in the advanced modules that are based upon Object Oriented (OO) principles (Sajaniemi & Hu, 2006). Consequently, many universities choose to adopt the OF approach despite initial challenges (Sajaniemi & Hu, 2006). However, this might result in causing detrimental effects on student self-efficacy and competence. Moskal, Lurie, and Cooper (2004) reported frustrations faced by students when the OF approach was used for CS1 instruction, whereby students were overburdened by having to learn PF and OF concepts in one module. Thus students were obstructed from developing fundamental programming concepts. Burton and Bruhn (2003) reported similar challenges experienced by their students. However, despite these difficulties, some educators believe that early exposure to OO software design principles helps to develop programming competencies required for the future course modules.

A second approach is the use of pedagogic programming languages developed for teaching purposes. The most widely used programming languages in universities for CS1 curriculum instruction are C++ and Java (Moons & De Backer, 2013). Similarly, C++ and Java are ranked as the top two most popular and influential programming languages of 2018 (Putano, 2018). However, there is a growing belief that these languages are not pedagogically suitable (Close, Kopec, & Aman, 2000; Hadjerrouit, 1998; Mannila & de Raadt, 2006). Research shows that programming languages like Python (Python, 2015) would be more appropriate for CS1 instruction (Agarwal & Agarwal, 2005; Agarwal, Agarwal, & Celebi, 2008; Leping et al., 2009). There are several studies documenting the suitability and positive outcomes of using Python for CS1 instruction, among them, but few are empirical in nature. Several of these studies' findings are of some concern with respect to generalisability. For example, the study findings of Koulouri, Lauria, and Macredie (2015) revealed that using Python instead of Java facilitated students' learning of programming concepts, which was quantified by a number of indicators such as frequent use of important programming constructs. However, the authors explicitly noted that one possible explanation for the students' increased and more elegant use of difficult programming constructs such as Loops compared with students who were instructed

in Java could be that Python hides a lot of details from students. For example, the concept of control variables in loops is generally a difficult concept for novice programmers. In Python these details are updated “behind the scenes”, thus, students are not required to update the control variables explicitly, unlike the case for programming languages such as Java and C/++. This suggests that loose syntax programming languages such as Python make programming easy. However, the consequence of its use is that students fail to achieve a good foundation of the fundamental programming concepts when they move to more advanced courses.

While some instructors strongly believe the language of CS1 instruction matters, others hold the view that programming language choice does not influence CS1 student competence. For example, a very recent study suggests that switching the programming language of CS1 instruction from VBA to MATLAB had no impact on CS1 student competence, leading to the conclusion that the choice of programming language does not influence understanding of CS1 concepts (McPheron, Gratiano, & Palm, 2015). A similar conclusion was drawn by another study of a similar line of inquiry (Farag, Ali, & Deb, 2013). Likewise, a study conducted by Alzahrani, Vahid, Edgcomb, Nguyen, and Lysecky (2018) showed that programming was not made easier for students by learning CS1 in Python. In this study, students were presented with 11 programming tasks in either Python or C, which were almost identical. The study shows that the students who were instructed in Python had a higher struggle rate (measured by time or attempts on an exercise) than those who were C++ instructed. Therefore, some educators surmise all the common languages such as Java, C/C++, and Python are equally suited to instruct CS1; however, programming language choice should essentially depend on such factors as market need, ease of use and its impact on more advanced computer programming courses, and the availability of pedagogical resources.

The third approach is to use a program development environment tailored for educational purposes that align with specific programming language choices. In the past decade, teaching programming with basic editors and command line tools have been replaced with more sophisticated Integrated Development Environments (IDEs) that are specially designed to support the teaching of CS1. Among such IDEs, BlueJ (Kölling, Quig, Patterson, & Rosenberg, 2003) has been widely promoted as a tool that delivers considerable positive effects on CS1 instruction (Kölling, 2015; Patterson, Kölling, & Rosenberg, 2003). Similarly, IDLE (Python, 2015) is an educational IDE for Python program development. While the use of pedagogical IDEs is accepted as good practice, some universities still choose to use industry level IDEs like Eclipse, Borland JBuilder, and Microsoft Visual J++. Among the industry

standard IDEs, Eclipse is one of the well-accepted choices. Some of the reasons for its popularity are because of its open-source technologies, its plug-in architecture, and extensibility for other programming languages with a uniform look and feel (des Rivières & Wiegand, 2004). Additionally, it allows the option to build Java plug-ins to suit various pedagogical needs (Bergin, 2000).

Conversely, there is also a view that a simpler program development environment such as command line tools with a simple editor increases students' understanding of concepts and the process of computer program development. For instance, Chen and Marx (2005) cited that novices can potentially acquire useful mental models by learning programming from a command line. Dillon, Anderson, and Brown (2012) explored the visual IDE's and command line programming environments and their effects on students' understanding of programming concepts. Their study shows that visual IDE's could provide a lower learning curve for students, however, command lines tools were found to be more effective to broaden students' understanding of programming and related concepts. This is because the command line environments are more restricted and the use cannot bypass the fundamental steps of learning to computer program thus enabling students to develop a more concrete understanding of program behaviour (Dillon, Anderson-Herzog, & Brown, 2012). In contrast, visual IDEs can combine multiple behaviours into a single button click. This may cause students to develop a false perception about programming preventing students from developing the required fundamentals, concepts and procedures (Dillon, Anderson-Herzog, & Brown, 2012).

A fourth approach is the use of effective assessment and feedback methods that assess and foster the attainment of learning outcomes consistently. Although few studies have been carried out in the field of computer programming, the role of assessment and feedback is one of the oldest concerns of the discipline of education (Atlas, Taggart, & Goodell, 2004; Bandura, 1986; Kinnunen & Simon, 2010). Studies have demonstrated that individual student attitude and sensitivity towards feedback on their assessment can positively influence competence and self-efficacy (Atlas et al., 2004; Linderbaum & Levy, 2010). For example, a study conducted by Lee and Ko (2011), reported improvements in CS1 student competence when the quality of feedback was improved by personalising the feedback to meet students' individual needs. Similarly, there is evidence of improving student competence when students were given feedback on their performance at the early stages of a CS1 course (Alemán, 2011; Traynor, Bergin, & Gibson, 2006). Furthermore, Ebrahimi (2012) demonstrated in a study that consistent feedback of potential errors made by students in the early stages of learning to

programme encourage students to follow the correct approach to solving programming tasks. This was found to be useful specifically for students who were struggling with the foundational programming concepts at the early stage of the course. However, according to the authors this approach prevents students developing creative problem solving skills for their programming tasks.

3.3.3. A model of factors influencing CS1 student competence

While the above discussion is not an exhaustive list of factors associated with student competence, the main aim of the succeeding section is to generate a list of factors that have been frequently discussed in the CS1 literature. These factors can be organised into two broad categories as shown in Figure 3.4. As with other scholastic achievements, there are more environmental and ecological factors. However, the aim here is to identify those factors that are open to intervention at the institute level to improve overall student competence, and lead support to some aspects of the validity argument of the investigation.

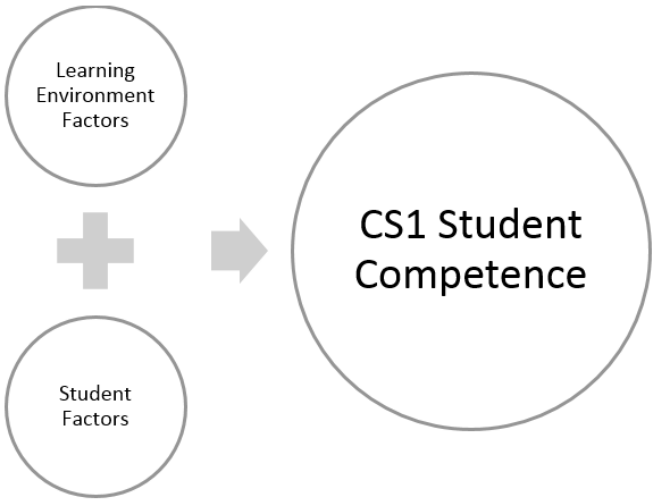


Figure 3.4. Two broad categories of factors associated with CS1 student competence

3.4. Summary

This chapter stressed the primacy of a theoretical framework to establish the validity of the measure. A critical analysis of the literature was presented with the focus of uncovering the elements embodying the variable CS1 student competence pertaining to the construct of CS1 student competence. Then, a theoretical model of CS1 student competence variable was defined. The second section presented external factors related to CS1 student competence. The next chapter discusses the evolvement of the older conception of validity into contemporary

unified validity. This will be followed by a comparative analysis of the characteristics of the two most widely applied measurement theories for psychological measurement construction.

Chapter 4 –Measurement Development and Validity Evaluation Models

4.1. Introduction

This chapter is divided into three sections. The first section provides a brief history of validity with an overview of how traditional forms of validity evolved into a unified form. It covers issues pertaining to older conceptions of validity as a driving force into the concept of unified validity. Then, a comparison of two of the widely known measurement models for instrument construction is presented, leading to a discussion on the application of the Rasch Measurement Model (RMT) to construct interval-level measures.

4.2. Conceptions of Validity and the Unified View

Validity is an important tenet of psychometric instrument development. This is because the credibility of instrument outcomes depends on the validity evidence collected to support the appropriateness of the interpretations, uses, decisions based on assessment results, and critical appraisal of instrument flaws. Validity evidence is also useful for providing information when making decisions about the choice of an instrument (Cook & Hatala, 2016). Careful integration of validity principles and procedures into the instrument development process is a way to ensure that all aspects of the development process are attended to, consequently helping to assay the validity argument for the investigative process. This section presents a brief history of the development of older conceptions of validity into the modern unified view.

4.2.1. Review of the evolution of the concept of validity

Over the last half-century, a major conceptual and definitional shift with a broader interpretation of the term “validity” within psychometric testing has been observed (Goodwin & Leech, 2003). The conception of validity has progressed from relatively simpler limited criterion-related models to a unifying, more sophisticated series of models with emphasis on the construct of interest (Kane, 2001). Similarly, the definition of the term validity has evolved in parallel with the changing epistemological views of validity.

In the early 19th century, validity was viewed as a static property of the measure manifested in the test itself leading to a judgment about whether the test was valid or not (Goodwin & Leech, 2003). In this notion, validity is an empirical index attained by a single correlation (predictive correlation coefficient) of the test with a criterion measure; a test is valid for anything with which the test correlates (Guilford, 1946). One of the fundamental issues with this early model is the validity of a measure depends on the validity of the criterion

measure. Consequently, a criterion measure also requires validation with another criterion measure and so on, resulting in a paradox of potential infinite circularity (Messick, 1993).

After the publication of the seminal article by Cronbach and Meehl (1955), validity was conceptualised as three specific types – content, criterion-related and construct. This persisted until the 1980's when the meaning of validity became more dependent on the use; validity was defined as the extent to which the test fulfilled its intended purpose (American Psychological Association, American Educational Research Association, National Council on Measurement in Education, American Educational Research Association, & Committee on Test Standards, 1966) However, during the 1980s through to the 1990s, theorists (Cronbach, 1980, 1988; Messick, 1988, 1989) espoused other aspects of validity which focused on accuracy and the consequences of inferences drawn from test scores. With this conception of validity, construct validity became more widespread in instrument validation and this leads to a push towards a more unified view of validity.

In 1989, Messick (1989) presented his new conceptualisation of construct validity as a unified and multi-faceted concept. He acknowledged that a unified theory of validity was not his own idea, but rather the culmination of debate and discussion within the scientific community over the preceding decades. Under this framework, all forms of validity are related to and are dependent on the quality of the construct. Validity, he said, was "an overall evaluative judgment of the degree to which empirical evidence and theoretical rationales support the adequacy and appropriateness of interpretations and actions on the basis of test scores or other models of assessment" (p. 741). In cognisance of this definition, Messick (1995) also proposed a process framework for collecting validity evidence to answer the specific points highlighted earlier. Under this framework, all forms of validity are interrelated to each other and are dependent on the overall quality of the construct. The framework provides six different aspects of construct validity for determining the quality of a test as presented in Figure 4.1. This notion was exemplified by the Standard for Educational and Psychological Testing (American Psychological Association, American Educational Research Association, National Council on Measurement in Education, American Educational Research Association, & Committee on Test Standards, 1999) and the later versions. Standards for Educational and Psychological Testing (Standards) also endorsed that the process of validity testing of an investigation involves "accumulating evidence to provide a sound scientific basis for the proposed score interpretations" (p. 9). Wolfe and Smith (2007a, 2007b) later supplemented validity framework of Messick (1989, 1995) with an additional criterion – the interpretability

aspect – resulting in a seven aspect framework. Wolfe and Smith (2007a, 2007b) used RMT based methods to demonstrate the validity evidence, thus, this framework is popular among the researchers who employ the Rasch approach to measurement development.

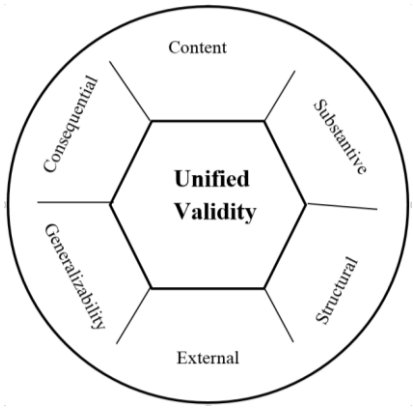


Figure 4.1. The six aspects of evidence in Messick's (1995b) unitary validity framework

Three important conclusions can be drawn from the new definition and the conception of validity. Firstly, unlike previous conceptions, validity is not a property or a numeric number attached to the instrument. Secondly, validation is not examined at the end stage of the instrument development process; rather, validity evidence is gathered throughout the instrument development process. Finally, validity refers to a specific interpretation or use of the measure and decisions grounded in the interpretation (Cook & Hatala, 2016). Furthermore, although validity is seen as unitary concept, it does not imply that validity cannot be usefully differentiated into distinct aspects to meaningfully address functional aspects of validity. In fact, as Messick (1995) himself highlighted, the importance of this distinction is to disentangle some of the complexities inherent in appraising the appropriateness, meaningfulness, and usefulness of score interpretation.

Given the unified view of construct validity; it places a significant weight onto the construct or the meaning of the variable under investigation. Some authors suggest that the claim of a latent variable existing as a quantitative property is in itself a scientific theory which makes predications (Popper, 2014; Salzberger, 2013). Therefore to substantiate the ontological claim of the scientific theory about the latent variable under investigation, a measurement model is required to provide evidence relating to its existence (Wilson, 2004). To provide the statistical evidence of a quantitative variable within the data, two broad categories of statistical

models exists – The Classical Test Theory (CTT) based models and Item Response Theory (IRT) based models.

With respect to construct validation, the CTT based models cannot infer anything beyond the factorial structure of the variable (Lissitz, 2009; Salzberger, 2013). This means, none of its procedures allow for theory-based predications to support the ontological claim entailed by the theory of the construct. The common methods available within the CTT based models such as convergent, discriminant or factorial validity merely provide evidence in supporting the structural theories linking various constructs, which all fail to address the true notion of construct validity (Lissitz, 2009; Salzberger, 2013). The essence of construct validity is testing for an existence of a quantitative structure within the data as theorised in the theoretical model of the latent variable. Contrary to the CTT based methods which fail to test for this structure, Rasch model (also known as 1-PL IRT model) advanced by Danish Mathematician Georg Rasch (Rasch, 1960) specifies the methods and the requirements data have to meet to infer measurement of a quantitative variable. The Rasch model is a unified approach to measurement issues postulated by Wright and Masters (1982) – unidimensionality, quantification, qualification and linearity all of which are required to support construct validity. The link between the Rasch practice of measurement development and Messick's unified validity have been demonstrated in the literature. For example, Smith (2001) illustrated evidences that inferred directly from the theory and practice of Rasch measurement to show all the facets of Messick's unified validity. Similarly Bond and Fox (2001) elucidated in detail issues of construct validity in psychological and educational assessments from the perspective provided by the Rasch approach to measurement construction.

In conclusion, over the last half-century, validity has undergone both definitional and conceptual changes. Validity was initially seen as a property of the instrument but the contemporary view relates to the interpretation of data obtained through the use of the instrument. History shows three major developmental stages of validity modes – criterion based models, construct based models and unified-construct based models. The major drawbacks of historical models were the driving force for Messick's unified view of the validity, which integrated the validity types as a construct based model. The modern view of validity places signification on the construct under investigation, which in itself is a scientific theory that entails evidence whether it exists as a quantitative property or not. The two common models for providing evidence of a quantitative property are CTT based methods and IRT based methods. The IRT based methods, particularly the Rasch approach to measurement

construction practices, is amenable to the validity aspects of Mesick. The next section provides a comparative analysis of IRT and CTT based models with respect to their characteristics.

4.3. Comparison of Measurement Models

This section provides a critical comparison of the distinguishing characteristics of two of the most widely used measurement development models, the Item Response Theory (IRT) models, specifically the RMT, and Classical Test Theory (CTT) models. The focus is on each of these measurement models' ability to address contemporary measurement concerns, and consequently any validity issues. Here, an exposition of the general features of the two models, including an overview of theoretical assumptions, is presented. The section concludes by highlighting the types of data produced by the scaling process for each model and the implications for data use in subsequent testing and mathematical operations.

4.3.1. Overview

The overarching goal of any educational or psychological assessment is to determine some aspects of human cognition as accurately and reliably as possible by assigning a numerical score (Erguven, 2013). However, as the literature on common assessment practices recounts, the majority of these assessments are not guided by stringent measurement theories. Measurement models and related theories are important in the measurement construction process because they provide a framework for guiding the measurement development process, considering issues, and addressing technical problems (Hambleton & Jones, 1993).

Basically, there are two categories of models available to assess, design, analyse, and score assessments – CTT and IRT. A comparative analysis of these models in relation to the basic premise, assumptions, and methods, would help assess their strengths and weaknesses and determine which might be the most suitable for this inquiry. Under the umbrella of each of the theoretical frameworks, a number of different variations exist. However, this section will explore the exposition of the general features of the two models, including an overview, theoretical assumptions, characteristics of resulting measures in each of the models, and the practical importance of scale levels.

The original CTT model was first published in the late 1960's (Novick, 1966). Since then, it has been the most widely used general framework in the 20th century for psychometric measurement development and assessment, particularly before the birth of IRT (Kline, 2005). On the other hand, IRT is a contemporary alternative to CTT, gaining momentum in the late 1970's despite its history being dated parallel to CTT (Furr & Bacharach, 2008). Although

CTT is still widely used for instrument development, IRT based models have become popular for an assortment of instrument development tasks. Essentially, the aim of both CTT and IRT involves establishing the position of the individual along some latent dimensions similar to the measurements of the physical sciences. However, each of these models is fundamentally different in their underlying mathematical theories.

Looking at the basic epistemology of CTT, it theorises that every measurement is an additive composite of two components (Alagumalai & Curtis, 2005). Hence, the observed score (denoted by X) of a candidate in an examination is composed of a True score (denoted by T) and a random Error (denoted by E) component; thus the observed score (X) is a linear function of the true score plus the random error represented by $X = T + E$ according to the founders of the theory (Novick, 1966; Spearman, 1904). To make this equation solvable, three assumptions are made: the average error score in the population of examinees is zero, true scores and error scores are uncorrelated, and error scores on parallel tests are uncorrelated (Alagumalai & Curtis, 2005; Hambleton & Jones, 1993). The first assumption is that if the same test is being repeated an infinite number of times with a candidate (assuming no learning occurred in between), the mean of all the scores is equal to the true score, as the random error fluctuation to both sides will nullify each other (Kline, 2005). However, standard deviation around the mean is due to the Standard Error of Measurement (SEM). Furthermore, it is assumed that these random errors are uncorrelated to each other and they are uncorrelated to the true score (Alagumalai & Curtis, 2005; De Champlain, 2010).

By contrast, the fundamental assumption of IRT differs in both the modeling process as well as the assumptions made. IRT is based on the simple assumption that the more able person has a greater chance of success in items of an assessment than a less able person; likewise, the easier the item, the better the chance of a person being successful in that item (Bond & Fox, 2015). (Bond & Fox, 2015). IRT is a probabilistic mathematical model based on strong assumptions that are testable using statistical procedures resulting in far superior statistics than those produced by CTT (Kline, 2005). Firstly, IRT based measures connect students and item interchangeably through an equal interval logit scale by a mathematical model (Sumintono, 2017). Particularly the Rasch model often considered to be 1-PL IRT model uses a model-driven approach to measurement development resulting in a logit ruler, which satisfies the five principles of measurement for human sciences postulated by Wright (2004). These are: a) produce a linear measure; b) overcome missing data; c) giving an estimate of precision; d) detecting misfits or outliers; and e) being replicable. The IRT family of models

differ from each other in two main ways: the item characteristics and the measurement models in terms of the response option format (Cappelleri, Lundy, & Hays, 2014). The differences among these models are depicted in Table 4.1.

Table 4.1

Comparison of IRT Models

Model	Item response Format	Model Characteristics
1-Parameter (Rasch) logistic	Dichotomous	Discrimination power equal across all items. The threshold varies across items.
2-Parameter logistic	Dichotomous	Discrimination and threshold parameters vary across items.
Graded response	Polytomous	Ordered responses. Discrimination varies across items.
Nominal	Polytomous	No respecified item order. Discrimination varies across items.
Partial credit (Rasch model)	Polytomous	Discrimination and power constrained to be equal across items.
Rating (Rasch model)	Polytomous	Discrimination equal across items. Item-threshold steps equal across items.
Generalised partial credit	Polytomous	Variation of the partial-credit model with discrimination varying across items.

Note: Reprinted from Overview of Classical Test Theory and Item Response Theory for the Quantitative Assessment of Items in Developing Patient-Reported Outcomes Measures, by Cappelleri, J. C., Jason Lundy, J., & Hays, R. D. (2014), *Clinical therapeutics, 36*(5), 648-662.

4.3.2. Item, test and person statistics

As a psychometric approach, IRT provides item, test and person information in statistical as well as graphical form. In essence, IRT is a set of mathematical models that describe the relationship between an individual's 'ability' or 'trait' and how they respond to items on a unidimensional scale (Nguyen, Han, Kim, & Chan, 2014). This relationship for a dichotomous item is depicted with an Item Characteristic Curve (ICC) as shown in Figure 4.2. The *x*-axis represents the different ability levels (in logits) in increasing order and the *y*-axis represents the probability of success in answering each of the items given the ability levels of the respondents. When items have polytomous response options, the interpretation of ICCs is slightly different in that the ICC plots the expected item score over the range of the trait. To

depict the probability of endorsing each response category for a polytomous item, the category probability curves (CRCs) can be plotted, with one curve corresponding to each response category (Cappelleri et al., 2014) as in Figure 4.3.

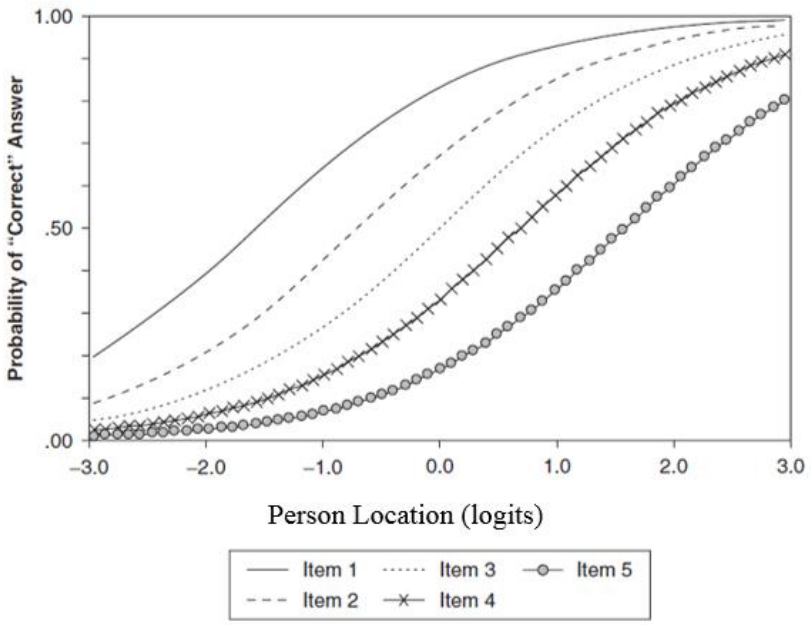


Figure 4.2. Sample Item Characteristic Curve (ICC) for dichotomous items

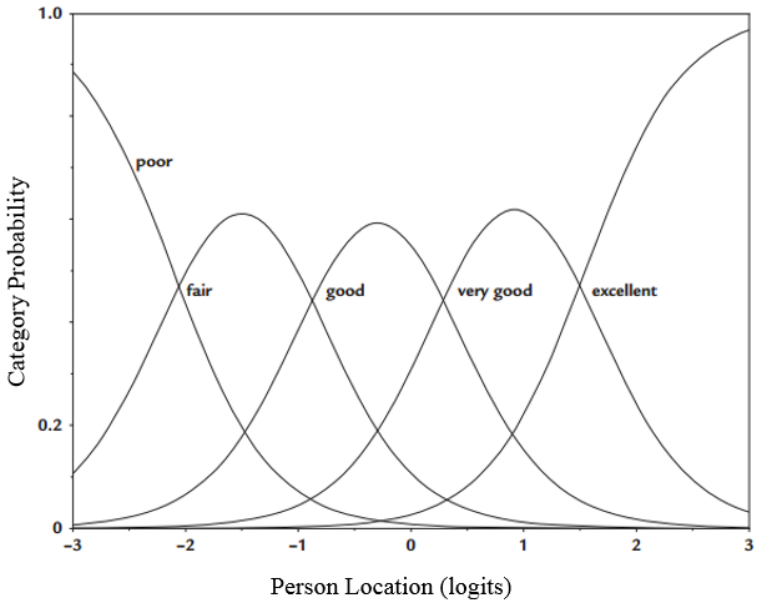


Figure 4.3. A sample CPC for a polytomous item

Item Information Function (IIF) is another useful set of statistics which provides information about ability (θ in relation to the item parameter, particularly at two extreme poles

that are otherwise difficult to get, in addition to items that are answered by all or none). It actually tells how well a test performed in estimating the ability levels of the students (Baker, 2001). IIF also provides information on the psychometric quality of an item in that the higher the information value at the given θ level, the better discriminatory power it has and vice versa, as shown in the Item Information Curve in Figure 4.4 (Furr & Bacharach, 2008). For example, Item 1 of Figure 4.4 can discriminate better the student abilities that lies between (-3.0 logits -2.0 logits), whereas, item 5 can discriminate the students at the higher levels (1.5 logits – 3.0 logits). Test Information Function (TIF) or scale information can be generated using item functions (sum of all IIFs) and the combined Test Information Curve would appear as shown in Figure 4.5 (Bond & Fox, 2015). Item functions provide useful information and precision of a particular item parameter, conversely, TIF can provide useful information at the test level (Furr & Bacharach, 2008). TIF is convenient to draw a general interpretation of the test. For instance, a TIF that is peaked at some point on the ability scale measures ability with unequal precision along the ability scale (Baker, 2001). Such tests would measure the student abilities that fall near the peak of the TIF more accurately, whereas, the tests that have flat TIF's over some region of the ability scale can estimate the ability scores on that range almost with equal precision and outside the range with less precision (Baker, 2001).

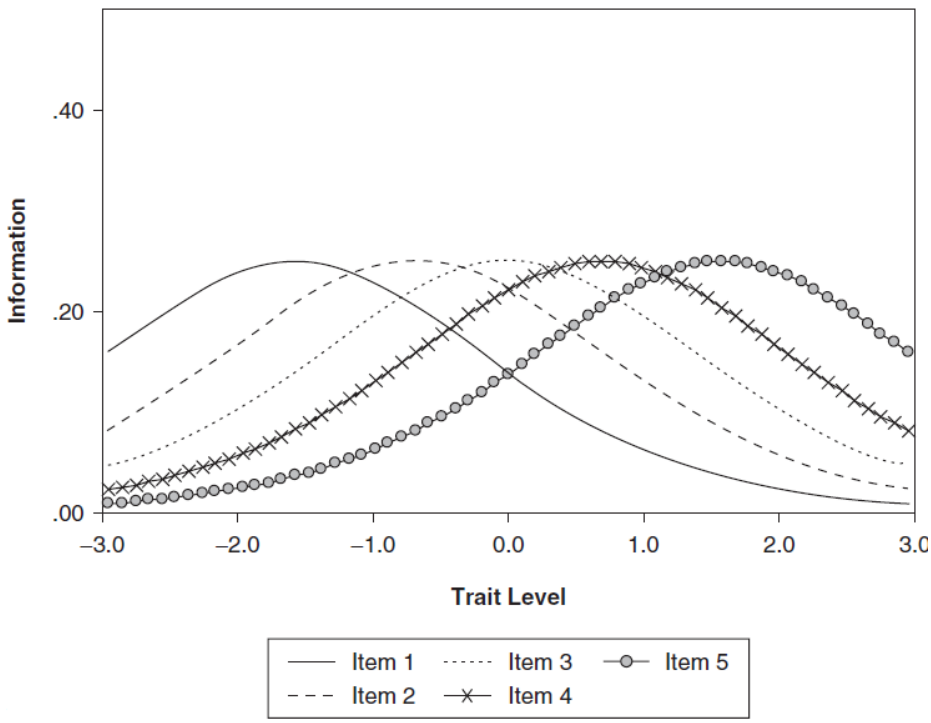


Figure 4.4. Item Information Curves

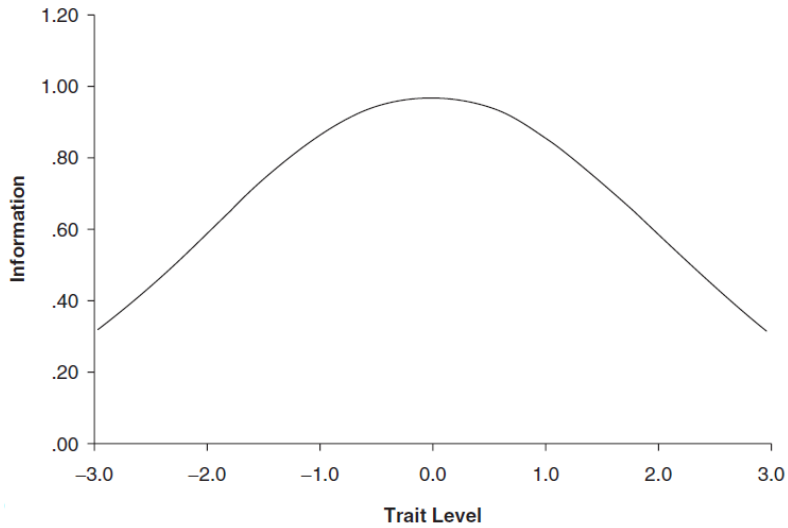


Figure 4.5. Test Information Curve

One of the strengths of the IRT models is its ability to present item and examinee attributes on the same logit scale (an interval level scale adopting the logit as its iterative unit) as shown in Figure 4.6. Logits is the unit of measurement (log odds unit) that results when IRT software such as the Rasch model is used to transform raw scores obtained from ordinal data to log odds ratios on a common interval scale (Bond & Fox, 2015). The benefits of the item-person map includes showing the relationship between item difficulty and person ability, demonstrating the extent of item coverage or comprehensiveness (item targeting), and the amount of redundancy and the range of the attributes in the sample. Most importantly, the logit scale is an interval scale in which the unit intervals between the locations on that item-person map have a consistent value or meaning (Bond & Fox, 2015). Therefore, it is possible to obtain a relatively concrete picture of response pattern probabilities for an individual given the trait score (Hays, Morales, & Reise, 2000). For example, the probability of success for any person on an item located at the same point on the item‑person logit scale is 50%. Similarly, when the person's ability exceeds the item difficulty level, then there is more than a 50% chance of that individual successfully completing the task being assessed. Conversely, if the person trait level sits below the item difficulty level, then there is less than a 50% chance of successful completion. Furthermore, it is possible to estimate the probability of success for each person on each item from this scale.

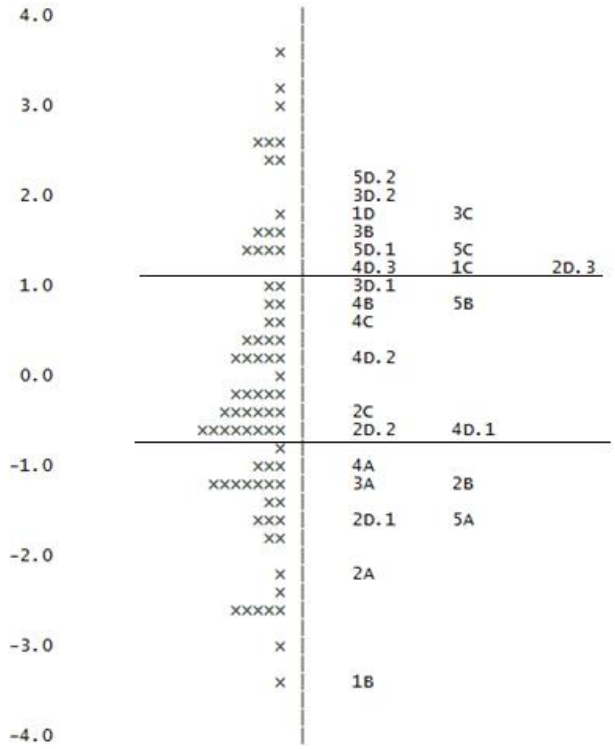


Figure 4.6. Item-person map

In comparison to IRT, CTT has very limited information, especially at the item level, basically depending on P-value and Item-Correlations (point-biserial correlation coefficient) (De Champlain, 2010; Fan, 1998). Item curves as shown in Figure 4.7, could be drawn using P values (proportion of individual respondents in a sample that pass/endorse an item) to gain more fine-grained information of an item similar to ICC. Item curves are very useful for finding out the overall performance in terms of how well the individual performed in the item levels (Kline, 2005). Analogous to the IIF of an IRT, CTT item curves could provide information on the items that are discriminating at different θ levels. Item-to-total correlations provide an index of the differentiating power of the item, and is typically referred to as item discrimination (Kline, 2005). Unlike IRT, the scale score is not characteristically informative about the item response pattern (Hays et al., 2000), especially those individuals whose score range is close to the middle. Another point of interest is unlike IRT, in CTT there is technically no absolute item difficulty or discrimination that generalises across samples or populations of examinees (Albano, 2017). The same goes with ability estimates in that they fluctuate with the overall ability of the sample (Albano, 2017; Fan, 1998).

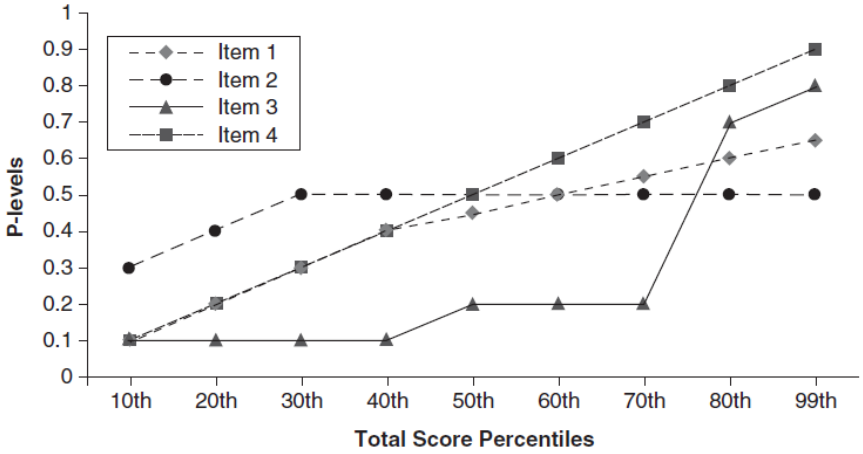


Figure 4.7. Item curves based on p-levels.

One of the major drawbacks of CTT is that the examinee's ability is defined by the characteristics of the particular test: the difficulty of the test is determined by the sample of examinees who take the test, contrary to the item level information in IRT (Fan, 1998). In CTT, the p-value (item difficulty level) is derived from the content of the subject matter as well as the ability of the test samples; hence the p-value of an item difficulty will vary across different samples. Therefore, as the characteristics and sample pool changes, the validity and reliability vary. Consequently, the estimates of psychometric features derived from instruments founded on CTT are not generalisable beyond the sample it was tested on (Hambleton, Swaminathan, & Rogers, 1991). Contrary to CTT, IRT has the property of parameter invariance (up to a set of linear transformations) in that the set of item parameters and the set of examinee parameters are not tied to the model (Rupp & Zumbo, 2006). This implies that parameter values are identical in separate examinee populations or across separate measurement conditions, which are commonly investigated through estimated parameter values from different calibration samples (Rupp & Zumbo, 2006). Therefore, the calibration, item parameters and sample parameters derived from the original samples and items could be used to measure the same trait across many other similar populations, unlike CTT.

4.3.3. Model Fit

IRT requires the Goodness of Model Data Fit to be verified before applying IRT models for the given set of data (Fan, 1998). These include evaluation of both in-fit and outfit statistics to determine how data-to-model fit occurs for each item and person (Royal, 2010). If only these assumptions are tenable, the IRT models exhibit model-data-fit; hence IRT models can be applied to generate measures from data. A limitation of IRT is that common IRT models cannot

be applied to estimate the ability and item difficulty of the test forms involving different domains (multidimensionality): compromising any of the requirements of the model could result in erroneous information on estimation of ability and item difficulty. However, this weakness is also the strength of these models because unidimensionality is one of the fundamental properties laid down by Thurstone in the theoretical requirements for measurement in the social sciences (Andrich, 1978, 1989). Similarly, the unidimensionality characteristic is among the four measurement criteria advanced by Wright and Masters (1982) for fundamental measurement. Contrary to IRT, the "major advantage" of CTT is that its assumptions can be easily met by most of the data sets, making it easy to apply in a variety of testing situations (Abedalaziz & Leng, 2013). However, it should be noted that not all CTT models are considered weak; in fact models such as binomial test models and Generalisability Theory are based on much firmer assumptions of error distribution and differentiation (Nodoushan, 2009).

4.3.4. Reliability, internal consistency, and measurement error

With respect to error measurement, the fundamental approaches and theories of each framework are different. CTT assumes and produces a standard error of measurement that spans the entire spectrum of samples (De Champlain, 2010; Royal, 2010); essentially assigning the same amount of error to every individual representing the data set (Alasuutari, Bickman, & Brannen, 2008; Embretson, 1996b). However, studies confirm that reliability coefficients calculated with repetitive measurements are found to vary, specifically for individuals with higher levels of the property that is measured (Feldt, Steffen, & Gupta, 1985), but this violates the fundamental error measurement formula of CTT. This clearly explains practically that the traditional CTT formula does not adequately represent the error propensity of most of the examinees. This has been identified as one of the fundamental drawbacks of CTT (Saltstone, Skinner, & Tremblay, 2001).

In addition to unaccounted undifferentiated gross error (Shavelson & Webb, 1991), CTT's concept of reliability is also related to the principle of correlation. In CTT models, it is assumed that the items showing high factor loadings in factor analysis procedures contribute to high reliability through high item intercorrelation (Ganglmair & Lawson, 2003). However, both factor analytical procedures and internal consistency indices like Cronbach's alpha can be misleading and may artificially inflate in some conditions. For example, the high correlation among the subsets of items could mean redundancy of items rather than a relationship with the construct in question (Steinberg & Thissen, 1996), therefore, the alpha is increased (Tavakol

& Dennick, 2011). (Tavakol & Dennick, 2011). It is also worth highlighting that not only the length of a test or the correlation between items influence the Cronbach's Alpha coefficient, but also the influence of the empirical distribution of the measure is influential, which is often ignored in CTT based traditions. Similarly, it is incorrect to assume a high coefficient alpha always means a high degree of internal consistency. As Streiner (2003) explains, the coefficient alpha is also affected by the length of the test. More precisely the alpha increases as the number of items in the test increases (Streiner, 2003), suggesting reliability based on a coefficient alpha can be misleading. Another issue as recounted by Embretson (1996b) is, just like other measurement properties of CTT, the standard error of measurement is derived from population-specific data resulting in a changing measurement error as the population changes. This is due to the two main variables of the measurement error formula; variance and reliability vary when the sample changes.

CTT's concept of reliability and measurement error is a characteristic of the measurement, which depends on the degree to which the measurement is free from measurement error, and is based on the single index measured from test and sample characteristics (Alasuutari et al., 2008). While this may be helpful in some situations, like parallel testing, IRT takes a totally different approach that produces a standard error for each person and item. In fact, the most significant difference between CTT and IRT is the way measurement error is conceptualised (Alasuutari et al., 2008). Unlike CTT, where reliability is represented by a single value (e.g. coefficient alpha), IRT's approach is more fine-grained in that measurement precision is different for different levels of ability (Antony & Barlow, 2002) but generalises across populations (Embretson, 1996b). In IRT, as one would naturally expect, the standard error is lowest for moderate trait levels (i.e.: z scores near zero) while the error is highest at the extreme trait levels. This is because measurement precision depends on the number of appropriate items to test the trait level of the individuals. The extreme ends usually contain very few items compared to the middle. In addition, just as CTT calculates a single measurement error value for the entire population, IRT too can calculate a single value to describe the entire population (Embretson, 1996b). This is calculated by averaging the IRT measurement error estimates of all the individuals of the sample population.

Unlike CTT's theory in which larger (more items) tests are more reliable, IRT asserts that in some testing situations such as in computer adaptive testing, shorter tests are more reliable than the longer tests (Embretson, 1996b). The backbone of most computer adaptive testing is IRT models, where items are individually selected to match the ability: the items that

are too extreme are avoided, resulting in more uniform measurement errors at all trait levels (Smits, Cuijpers, & van Straten, 2011; Wainer, Dorans, Flaugher, Green, & Mislevy, 2000). Furthermore, modeling plots of item information resulting from a test could serve as a way for improving the items in an item bank of a test; therefore increasing reliability precision (Lai, Cella, Chang, Bode, & Heinemann, 2003). This information is particularly useful from the researcher's perspective as it helps to determine the extent to which each item or person measure is stable and useful (Royal, 2010). Another advantage of IRT is that measurement error is expressed in the same unit of measurement, hence the direct comparison of the estimate of ability and error is possible, in addition to the possibility of using it to build a confidence interval around the estimate (Partchev, 2004). The variance of ability is inverse to that of the Test Information Function (TIF) and can be demonstrated as in Figure 4.8.

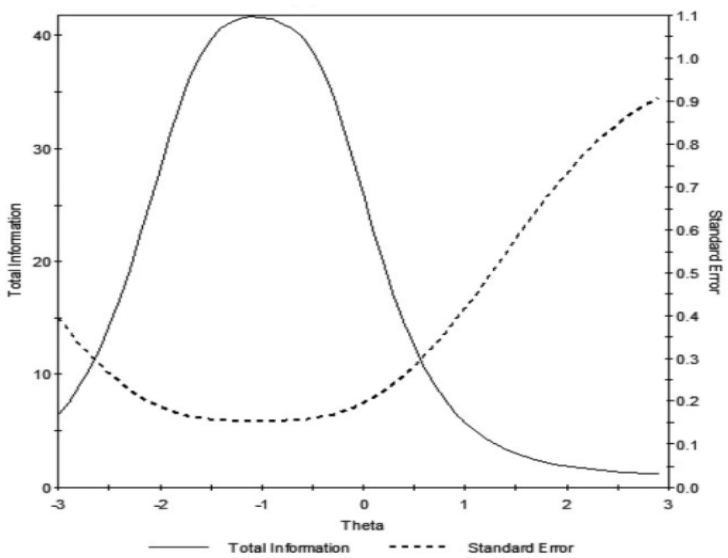


Figure 4.8. TIF and standard error of measurement

4.3.5. Missing data

Missing data is a prevalent issue in education and psychological research that is hard to avoid in a data set in real practice (Hohensinn & Kubinger, 2011). Missing data, as described by Rubin (1976), differs between missing completely at random (MCAR), missing data not at random (MAR), and systematic or non-ignorable missing data stemming from a variety of reasons. MCAR is defined as when the probability that the data are missing is not related to either the specific value which is supposed to be obtained or the set of observed responses, whereas MAR is set of observed responses, but is not related to the specific missing values which is expected to be obtained (Kang, 2013). Regardless of the reason or type of missing

data, if data is not handled appropriately, it could result in two main problems: negative impact on statistical power (Verma & Goodale, 1995) and biased parameter estimates leading to invalid conclusions (Hohensinn & Kubinger, 2011). Furthermore, missing data could also potentially influence the construct validity of the instrument used to measure the latent trait (McKnight, McKnight, Sidani, & Figueredo, 2007). Most of psychological variables we try to measure are not directly observable, rather inferred based on the responses to items measuring the latent construct. Therefore, if responses to some items are not captured, validity can be affected because the information captured may not reflect full range of the construct under investigation (McKnight et al., 2007)

The most common approach to handle the missing data is listwise and pairwise deletion (Peugh & Enders, 2004). The listwise deletion simply omits those cases with the missing data and analyses the remaining data (Kang, 2013; Peugh & Enders, 2004). Although listwise deletion is a common approach to handle missing data when MCAR is satisfied (Kang, 2013), it is known to affect the statistical power of the tests conducted because the statistical power relies in part on a high sample size. This is because every deletion results in an exclusion of data which leads to a deduction of the sample which is being statistically analysed (Allison, 2001; Roth, 1994). This approach is particularly problematic and imposes bias if MCAR is not satisfied (Olinsky, Chen, & Harlow, 2003). Another common approach to handling missing data is pairwise deletion (Olinsky et al., 2003; Peugh & Enders, 2004) which attempts to minimise the loss that occurs in listwise deletion. Unlike listwise deletion, pairwise deletion eliminates information only when the particular data-point needed to test a particular assumption is missing. If there is missing data elsewhere in the data set, the existing values are used in the statistical testing (Kang, 2013). This technique has higher statistical power than listwise deletion as it preserves more information. Although this technique is typically preferred over listwise deletion because it increases the power in the analysis, it also requires MCAR being met in the data. The two most well-known problems with this technique are: (1) the standard of errors computed by most software packages uses the average sample size across analyses, which tends to produce a standard of errors that are underestimated or overestimated; and (2) it can produce an intercorrelation matrix that is not positive definite, which is likely to prevent further analysis (Kim & Curry, 1977; Wothke, 1993).

Other missing data handling methods such as mean substitution, where the mean value of a variable is used in place of the missing data value for that same variable are also applied in some instances. In this approach, if the missing values are not completely random, especially

in the presence of a great inequality in the number of missing values for the different variables, the mean substitution method may lead to inconsistent bias (Kang, 2013). Furthermore, an increase in sample size with no new information also results in an underestimation of the errors (Malhotra, 1987). Regression imputation is another approach to missing data. In this approach, instead of deleting any case that has missing values, it replaces the data values with a probable value estimate based on other information available from other variables (Kang, 2013). Then the data set is analysed with the standard set of procedures for complete data. The major advantage of this approach is that it retains a great deal of data and minimises the bias significantly that may be imposed from listwise or pairwise deletion, as this approach minimises the distortion to the distribution of the data and altering the standard deviation.

More contemporary approaches to handling the missing data are multiple imputation (MI) and maximum likelihood (ML) estimation methods. Multiple imputation methods are more popular than the maximum likelihood methods, although the latter are generally more preferable (Allison, 2012). The MI method was an idea proposed by Rubin (1977) to deal with the problem of increased noise due to imputation. The MI follows a series of steps: (1) Imputation – unlike single imputation, the imputed values are drawn m times from a distribution; (2) Analysis – each of the m datasets is analysed resulting m analyses; and (3) Pooling – combines the *m* results into one by calculating the variation in parameter estimates. There are different approaches to multiple imputations and the choice depends on the type of "missingness" in the data matrix. The ML uses a totally different approach to missing data. Contrary to the data imputation used by MI, the ML does not impute any missing data, but rather it estimates parameters directly using all the information that is already contained in the incomplete data set (Dong & Peng, 2013). The ML estimate of a parameter is the value of the parameter that is most likely to have resulted in the observed data. The optimal parameter is estimated by maximising the likelihood function of the incomplete data (Dong & Peng, 2013).

Although none of these methods is inherently better than the other, Allison (2012) argues that ML has optimal statistical properties (if assumptions are met), and it has several advantages over multiple imputation. The most significant advantage of ML over MI is that there is no potential conflict between an imputation model and an analysis model (Allison, 2012). The specification of the model – the imputation model – for producing the imputed values is often a challenging task in MI and if the imputation model is poorly specified, there is a risk of creating invalid estimates of the target parameters (Nguyen, Carlin, & Lee, 2017). Similarly the ML algorithm makes use of all the information in the observed data, in the presence of an

unlimited number of missing-data patterns. However, the major advantage of MI over ML is its wider applicability on different statistical models unlike ML methods that are limited mostly to linear models (Allison, 2012).

With respect to the approaches mainly adopted to handle missing data by each measurement development approach is different. CTT's approach to handling missing data has been a predominant issue as recounted in several studies (Montiel-Overall, 2006; Peugh & Enders, 2004). The CTT models does not tolerate missing data; consequently, some technique must be applied to fill the missing data. This forces the model to apply an imputation technique (Sébille et al., 2010) as discussed before which most of the techniques literally fill the missing data gaps (Graham, 2009; Rubin, 1976). Another issue with missing data in CTT based models is that the missing data can significantly decrease the reliability of the measure. In CTT based models the reliability is directly related to the total number of items used to capture the construct; more items increase reliability. Therefore, missing data or being unable to capture data to an item reduces the total number of items to capture the construct and consequently reducing the reliability of the measure (McKnight et al., 2007).

In general, IRT models are very robust in dealing with missing data. IRT models estimate the person and item parameters using all the available information (sufficient statistics of person, item or Rasch-Andrich threshold) based on likelihood algorithms such as Joint Maximum Likelihood (JML), Constrained Maximum Likelihood (CML) or pairwise conditional estimation methods (Royal, 2010; Sébille et al., 2010). For example, a study which investigated how missing values influences the bias and precision of patient disability measurements reported that missing data proportions as high as 50 % also had a negligible bias for person ability estimation (Erdogan et al., 2013). This indicates the level of tolerance to missing data by IRT models. Although missing item responses do not cause bias on item and person parameter estimates, studies suggest that the condition lowers the precision of the estimates and lessen the sensitivity of the fit statistics of the model. For example, a study conducted by Zhang and Walker (2008), concluded that as the degree of missing data increases, the person wrongly diagnosed for both fitting and misfitting to the IRT models also increases. The results of this study were consistent with other similar studies such as De Ayala (2003) and Furlow, Fouladi, Gagne, and Whittaker (2007). To address the precision issue with missing data in IRT models, some studies suggested that precision could be improved using item response function imputation methods (for more details on this method see Erdogan et al., 2013; Sijtsma & Van der Ark, 2003).

4.3.6. Score meaning

Test score meaning, as noted by Embretson and Reise (2000), requires specifying a standard and a numerical basis for comparison. In the case of CTT, the standard for score comparison is norm-referenced and the numerical basis is based on the rank order. The score is an estimate of the relative position of the tested individual with respect to a norm group who have taken the same test (Embretson, 1996b). One of the ramifications of norm-referenced score meaning is that without the context of normative information being specified, the score is meaningless (Alasuutari et al., 2008; Kline, 2005). Similarly, the norm-referenced score has no meaning with respect to what the person is capable of, thus, the score meaning cannot be used to determine whether the test taker has achieved the required competency for a given purpose (Wu & Adams, 2007). For example, Figure 4.9 shows two scales adapted from "Applying the Rasch model to Psycho-Social Measurement: A Practical Approach" by Wu and Adams (2007). The one on the left is the item difficulty scale, which shows the percentage of correct answers by the students for each category of questions, that is, 25% of students were able to solve the questions on most difficult topic (Arrays), while 90% of students were able to get correct answers for items on the easiest topic (conditional structure). As can be observed, there is no easy way to link these student abilities with item difficulties shown for various student scores – 25%, 50%, and 70% and 90% – on the left. One might intuitively infer that the abilities of those at the top, meaning for those who scored 90%, could answer questions of all difficulties; conversely, those at the bottom could most likely answer the few easiest. However, it is particularly problematic inferring corresponding abilities and scores of the students in the middle range. For example, a student who has scored 70% does not necessarily have any item on the functions correct, as it is not apparent what proportion of items constitutes each level of difficulty. Therefore, norm-referenced scores lack one of the fundamental characteristics of an ideal measurement – "linking scores to tasks so that a substantive meaning can be given to scores in terms of underlying proficiencies or skills" (Wu & Adams, 2007, p. 12).

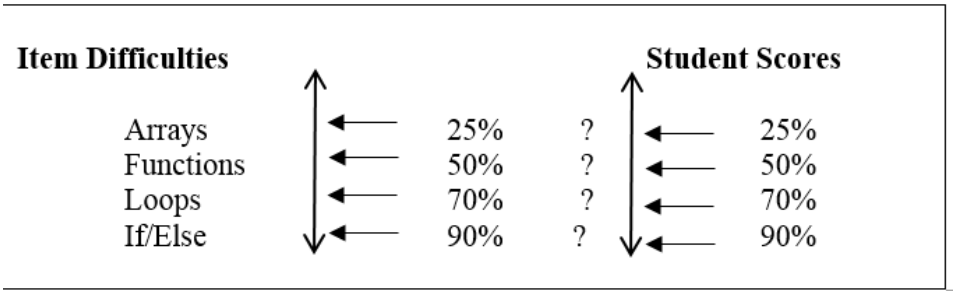


Figure 4.9. Linking scores to tasks. Adapted from "Applying the Rasch model to Psycho-Social Measurement: A Practical Approach" by Wu and Adams (2007), Melbourne: Educational Measurement Solutions, p. 12.

Contrastingly, in IRT, items and persons are calibrated on a common scale, and the score is compared with respect to items, that is, the relative position of the item and trait level of the person has a direct meaning for the expected item performance (Embretson, 1996b). By calibrating both item and person on the same scale, it is possible to construct interpretations for person ability scores in terms of task demands (Wu & Adams, 2007) as shown in Figure 4.10. The two scales shown in the left and right of Figure 4.10 – person ability and item difficulty – are linked by the mathematical function of the probability of success shown in Figure 4.11. For any given student ability score, it is possible to compute the probability of success on any item of the scale. The probability of success (answering) on an item of the scale depends on how far apart an item is from the relative position of the person on the common scale (Bond & Fox, 2013; Embretson, 1996b). For example, if the trait level of the person exactly matches with the item difficulty, then there is a 50% chance of answering that question and more than a 50% chance of answering every item below the person trait level. As shown in Figure 4.11, when the person's trait level is very low (represented by the horizontal axis), then there is very little chance (close to 0) of the student being successful in that item. Conversely, for a high achiever, there is a very high chance (close to 1) of being successful, and if the ability of the student exactly matches with the item difficulty, then there is a 0.5 (50%) chance of being successful. Finally, another defining characteristic is that if the items are structured by a construct model of a hypothetical learning path, substantive trait level meaning can also be inferred (Embretson, 1996b).

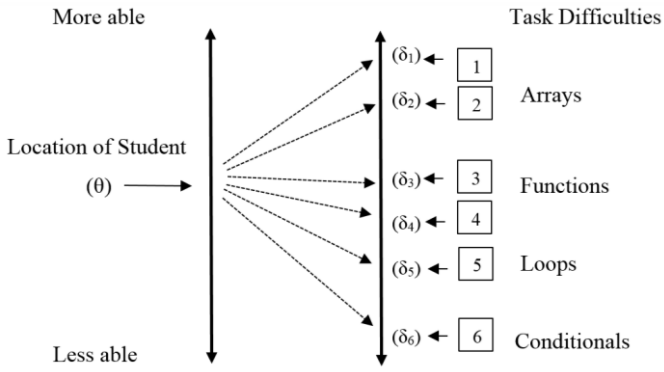


Figure 4.10. Linking student abilities to tasks in IRT models. Adapted from "Applying the Rasch model to Psycho-Social Measurement: A Practical Approach" by Wu and Adams (2007), Melbourne: Educational Measurement Solutions, p. 15.

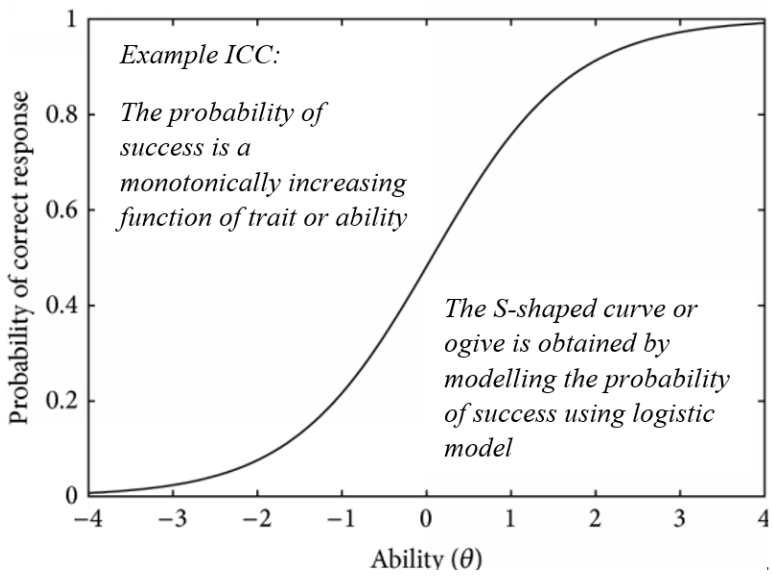


Figure 4.11. ICC showing probability of success on an item as the ability increases

4.3.7. Scale properties

One of the inadequacies of CTT is that the relative distance between each pair of scores is not maintained. This is because the relative distance between a pair of scores is directly influenced by the difficulty of the items involved (Embretson & Reise, 2000). This can be illustrated with a simple example of an easy test and a difficult test administered to the same group of students. In the easy test, the high ability students will differ very little in the total score; they will answer most of the items. Contrastingly, if a difficult test is administered, the performance difference will emerge because the persons with higher ability answer more items correctly than those with lower abilities. This relationship is shown in Figure 4.12, where four students (A, B, C, and D) were administered an easy and a difficult test. It can be seen that the

distance between the student A and C are quite close in the easy test (horizontal axis). However, in the difficult test, they are almost twice the distance apart (vertical axis). From this simple example, it is clear that the meaning of score differences depend on the item properties constituting the test, and does not provide a stable frame of reference of invariance in terms of distance between the students on the ability scale (Wu & Adams, 2007). Furthermore, as Figure 4.12 suggests the relationship between the two tests is not linear, thus, there is no way to map the two tests with a linear transformation using a scaling factor.

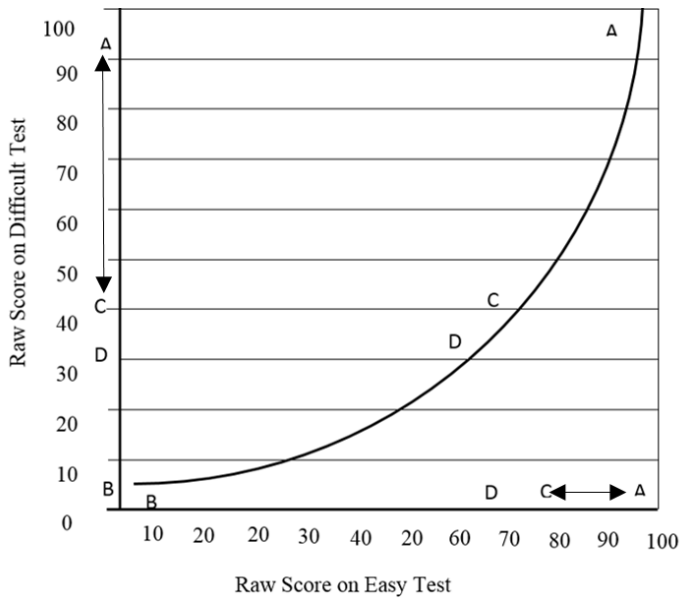


Figure 4.12. Student score distance variance in easy and difficult test in CTT. Adapted from "Applying the Rasch Model to Psycho-Social Measurement: A Practical Approach" by Wu and Adams (2007), Melbourne: Educational Measurement Solutions, p.11.

Some literature reports that the interval-level scale can be justified in CTT based raw scores given that two conditions will hold true (Jones, 1971). These specific conditions are that (a) the true trait level, measured on an interval scale, is normally distributed, and (b) observed scores have a normal distribution. As argued by Embretson and Reise (2000), the second condition can be met in two ways (a) selecting items that result in normal distributions by matching the item difficulty with the norm group ability or (b) normalising non-normally distributed observed scores by transforming to z-scores or percentile ranks. However, they argued that the first condition simply is an assumption, and if that assumption is not reasonably assumed for the trait, then the interval-level scale cannot be justified. Even in the cases where the conditions are being met, only one norm group can define the intervals and multiple norm groups can create paradoxes in justifying interval-level scales (Embretson, 1996b).

Luce and Tukey (1964), showed that physical concatenation can be obtained from responses produced by the interaction of two kinds of objects, namely persons and test items, if certain axioms are satisfied (Wright, 1985). The idea dates back to the work of Thurstone (1927) who provided rough examples of the idea of additivity of psychological constructs. Finally, Rasch (1960) concretised this idea and made fundamental measurement available by application of the Rasch model – a special case of IRT – to social scientists (Wright, 1997). Several authors of the time explored and confirmed the additive conjoint property of Rasch's measurement model. For example, Perline, Wright, and Wainer (1979) provided examples of the extent to which Rasch analysis can structure the data to achieve conjoint measurement requirements, specifically monotonicity and double cancellation. Similarly, Wright and Stone (1979) and later Wright and Masters (1982) explored these properties by showing how to obtain additivity from mental tests and the construction of additivity from the rating scale and partial credit data respectively.

In the Rasch model, the additive decomposition – when two parameters are additively related to third variable – is achieved with the formula, $LogOdd_{ni}(x = 1) = f(\theta_n - b_i)$ (Bond & Fox, 2015). The formula states that the "log odds" (probability of success) that a person (n) attempting any item (i) is simply a function of the difference between the persons' ability (θ_n) and the item difficulty (b_i) (Bond & Fox, 2015). According to this theory, interval scale properties hold true if the laws of numbers apply (Embretson, 1996b), meaning to say that the same performance difference must be observed when the trait scores have the same interscore distances, irrespective of their overall positions on their trait score continuum (Embretson, 1996b). Figure 4.13 shows that Rasch transformed scores maintain the property of invariance between the people irrespective of a change of measurement conditions, whether the test is easy or difficult, unlike the case for CTT. Note that the relative distance between A and C on the easy test (horizontal axis) and hard test (vertical axis) remains the same unlike the case for CTT based scores as illustrated in Figure 4.13. For example, the distance between person A and person C is almost 2 logits difference in both the easy and the difficult test. However, this does not imply that the absolute values of the Rasch scores for an individual are the same for the easy and hard tests, but the relative distances between people are constant (Embretson, 1996b; Wu & Adams, 2007), just as it is the same whether a Kelvin temperature scale or a Celsius scale is used for measuring the temperature. Maintaining this invariance of the distance regardless of the item difficulties justifies that a quality of interval-level scale has

been achieved (Embretson, 1996b), which is another fundamental property of an objective measurement (Wright & Masters, 1982).

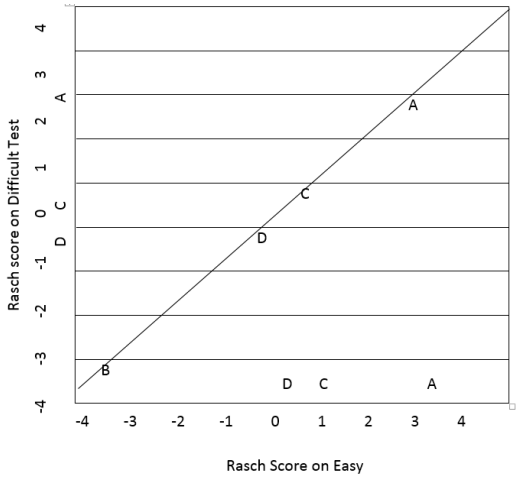


Figure 4.13. Student score distance variance in easy and difficult test in IRT models. Adapted from "Applying the Rasch model to Psycho-Social Measurement: A Practical Approach" by Wu and Adams (2007), Melbourne: Educational Measurement Solutions, p.16.

4.3.8. Practical implication of scale levels on statistical analysis

The four classifications of measurement defined by Stevens (1946) are ordered according to how many numbers of properties are applicable for number comparisons (Embretson & Reise, 2000). This means the number of properties or the level of measurement of data has important implications for what operations can be performed on them.

One of the frequently discussed issues within psychometrics, but as yet has no consensus, is the determination of the scale level one has achieved by CTT based on the measurement development process (Knapp, 1990). This is because CTT's underlying theory is not based on a justifiable measurement model; rather it is based on some assumptions. These assumptions are that the true score of the population are assumed to be (a) measured at the interval level and (b) normally distributed (Kline, 2005). Since there is no realistic way to prove the first assumption, the researchers focus on achieving the second assumption as justification to apply a parametric analysis. Furthermore, some authors hold the view (See Borgatta & Bohrnstedt, 1980; Gaito, 1980) that when the summed score distribution of CTT based instruments follow a certain shape (normality) the interval-level scale could be assumed. In fact, many popular statistical books and software used in educational research teaches that the normal distribution of data is required for the dependent variable, implying an interval scale of measurement allowing arithmetic operations. There has also been a notion that if scores are not

normally distributed, they can be normalised by an arbitrary monotonic transformation of scores. However, Michell (1997) refutes this idea by asserting that the raw scores require an additive structure in the data which can be determined by a statistical model such as the Rasch model. Furthermore, Harwell, Gatti, and Linacre (2002) elucidated the fallacy concerning the link between the normality of data and linearity by stating that linearity itself is independent of any particular sample distribution. Hence, a normal distribution is not a characteristic to achieve linearity.

There have been several studies demonstrating biased comparison results when the observed data did not meet the interval-level properties. For example, Maxwell and Delaney (1985) demonstrated how group comparisons using t-tests can be misleading if the data does not manifest the interval-level properties. Similarly, on several occasions, Embretson (1996a) demonstrated interaction effects are significant when Factorial ANOVA designs were applied to the data, while there was no such effect with interval-level data. All of the authors (Embretson, 1996a; Embretson & Reise, 2000; Maxwell & Delaney, 1985) elucidated that this effect was due to inappropriate test difficulty levels and the skewed population distributions of raw scores. The validity of other comparison results such as repeated measures and regression coefficients are also shown to depend on the scale level achieved (Embretson & Reise, 2000; Merbitz, Morris, & Grip, 1989). Therefore, an implication of these results is that an outcome of a research hypothesis based on CTT scores and IRT scores can lead to different statistical outcomes despite being calculated on the same set of observed data.

In summary, the comparisons have revealed that there are many measurement challenges in human science measurement development, which cannot be appropriately addressed in CTT, unlike IRT models. CTT and IRT differ in many respects including their theoretical grounding. The assumptions of CTT are easy to meet for a variety of testing situations, whereas, IRT is based on stricter criteria that require the data set to fit into the chosen IRT model. However, the advantage of IRT models is that it enables construction of interval level measures, whereas, the scale level achieved by CTT is at best the ordinal level. The next section presents the defining features of Rasch Measurement Theory which is known as One-parameter Logistic Model (1-PL) in some IRT literature.

.

4.4. Rasch Measurement Theory

The previous section highlighted that there are fundamental issues in CTT-based models with respect to achieving ideal measurement criteria as presented by Wright and Masters (1982). Rasch Measurement Theory (Rasch, 1960) is a modern approach to measurement development and is responsive to the measurement shortcomings manifested in classical theories. Most importantly, it reflects the basic criterion of invariance – a crucial feature of measurement – whereby the instrument should work for all the individuals in the sample irrespective of other factors (Andrich, 1988) just as the measures of physical sciences work. Therefore, the purpose of this section is to provide an overview of RMT and how the model's requirements lead to the interval-level scaling of data.

There was a view that the measurement of psychological properties cannot progress beyond ordinal scoring because it does not seem that the attributes of interest to psychologists can be concatenated (Perline et al., 1979). However, research into modern measurement theory has confirmed that empirical concatenation is not always required to construct interval-level measures. The measurement theorists of the past have demonstrated several models which yield interval-level scales (see Coombs, Dawes, & Tversky, 1970). The simultaneous conjoint measurement model of Luce and Tukey (1964) is generally recognised as an important theoretical contribution to the development of interval-level scale models (Perline et al., 1979). RMT is a practical realisation of this model, although references about the link between these two models are less acknowledged (Perline et al., 1979). According to Bond and Fox (2015), the Rasch model for measurement is the closest generally acceptable approximation of fundamental measurement principles in the human sciences that provides the same sort of rigorous measurement to the human sciences as those in the physical sciences. In fact, proponents of RMT argue that it is the only objective measurement model, which encapsulates the rules of sound scientific measurement (Bond & Fox, 2013; Embretson & Reise, 2000; Royal, 2010; Royal & Eli, 2013). For example, like many authors (Chien, Hsu, Tai, Guo, & Su, 2008; Tennant & Conaghan, 2007), Stenner (2001) advocated RMT as the only model that transforms raw scores into interval-level measures with sufficient invariance and objectivity:

> Measurement is the process of converting observations (e.g. counts) into measures (quantities) via a construct theory. The Rasch Model states a requirement for the way observations and construct theory combine in a probability model to make measures.

> There is no other combination of observation and theory that produces sufficiency invariance, and objectivity in the resultant measures. (p. 804)

Similarly, McAllister (2008, p. 490) praised RMT by describing it as "a statistical model for validating assessment tools that are particularly suited to quantifying human performances on assessment items".

RMT is a probabilistic model based on the simple idea that the probability of affirming or successfully completing a task depends on the ability of the person and the difficulty of the task. In other words, the more able person has a greater chance of success in difficult items than the less able person (Kline, 2005). For dichotomous data, this is expressed as a logistic function of the discrepancy between the person's ability (θ) and the difficulty expressed by the item (b) as shown in Formula 1 (Bond & Fox, 2015). This can be expressed as a logit function as in Formula 2 (Rasch, 1960). The other two choices of parameterisation of RMT are represented by Formulae 3 and 4; Formula 3 is the Rating Scale model where items share the same response structure (Andrich, 1978), and Formula 4 is the Partial Credit Model (PCM) where each item has its own response structure (Masters, 1982) respectively.

$$p_{ni} = \frac{e^{(\theta_n - b_i)}}{1 + e^{(\theta_n - b_i)}} \qquad (1)$$

$$ln\left(\frac{p_{ni}}{1 - p_{ni}}\right) = \theta_n - b_i \qquad (2)$$

$$ln\left(\frac{p_{nij}}{1 - p_{nij}}\right) = \theta_n - b_i - \tau_j \qquad (3)$$

$$ln\left(\frac{p_{nij}}{1 - p_{nij}}\right) = \theta_n - b_i - \tau_{ij} = \theta_n - b_{ij} \qquad (4)$$

Figure 4.14. Formulae for the three parameterisations of the Rasch model.

Where ln is the normal log; p_{ni} is the probability that person n affirming item i; θ_n is the "ability" measure of person n; b_i is the "difficulty" measure of item i the point where the highest and lowest categories of the item are equally probable; τ_j is the "calibration" measure of category j relative to category $j-1$, the point where categories $j-1$, and j, are equally probable relative to the measure of the item. No constraints are placed on the possible values of τ_j.

RMT uses an efficient, consistent and unbiased method for approximating item and person parameter estimates (person-free and item-free) that is easy to apply in a variety of practical contexts (Wright, 1977). However, this requires the data to fit the model instead of the model being modified to fit the data, inductively constructing measures from data. The fit

of the data to the RMT can be determined by calculating how much remains after data is used for calibration of item and person parameters (Wright, 1977). The degree of measurement achieved from an observed set of data depends on how closely the response data approximates the Rasch prescription of criterion for successful measurement (Cano et al., 2014). When the data is considered to fit the model, the resulting measure manifests fundamental measurement criteria; thus, invariant comparisons of items and persons can be made in terms of a constant unit (Andrich, 1988) by plotting both items and persons on a common continuum called a logit ruler. To achieve the fit of the data to the model, the response data structure of the instrument must closely approach RMT's criteria of rigorous fundamental measurement (Bond & Fox, 2015).

The requirements underlying RMT are (1) uni-dimensionality, (2) local stochastic independence, (3) monotonicity, and (4) sufficiency of simple sum statistics (De Jong & Kamphuls, 1985). Rasch models are robust to minor violations in that it is capable of calibrating data containing substantial variations to the item discrimination parameters (Linacre, 2000) and slight departures from other assumptions (Fisher, 1993). However, it is critical that these assumptions be carefully evaluated against the standard procedures reported in the literature (Cantrell, 1997), and take necessary actions to achieve acceptable fit of the data to the model to construct measures from data.

The requirement that the items should be located on a continuum or scale is among the theoretical requirements of measurement that were laid out by Thurstone in the 1920's. This means that the items must reflect an underlying latent trait that is unidimensional at some level of scale. In other words, the notion of uni-dimensionality posits that all the items constituting the test function in unison to form a single latent trait or dimension (Bond & Fox, 2015; Sick, 2010). However, unlike physical measures, the requirement of uni-dimensionality in latent traits cannot be satisfied fully (Hambleton & Jones, 1993).

To test this requirement, the Rasch approach to measurement development provides several procedures, statistics, and visual displays. Fit indices are the first point of reference to detect unidimensionality (Bond & Fox, 2015), followed by more complex procedures such as evaluating dimensionality by common factor analytical procedures or conducting a Principal Component Analysis of Rasch residuals (Smith, 2002; Tennant & Conaghan, 2007; Tennant & Pallant, 2006). Violation of this assumption will result in confounding effects that will preclude constructing the linear scale of the latent variable (Bond & Fox, 2015).

Local stochastic independence postulates that an examinee's responses to any pair of items are statistically independent when abilities influencing the test are controlled (Embretson & Reise, 2000; Verguts & Boeck, 2001). More specifically, the score on an item does not contain a clue to the score on other items, and should not correlate with each other (McCamey, 2014; Verguts & Boeck, 2001). This is a fundamental feature of the Rasch model, which states that probability of success on an item is totally determined by two factors – the item difficulty (δ) and the person ability (θ) as shown in Formula 1 of Figure 4.14. If there are factors other than these two influencing the probability of success for a person on an item, then the assumption of the Rasch model is violated (Wu, Tam, & Jen, 2016). Such violations of Rasch model requirements could result in a biased parameter estimation, which consequently artificially inflates reliability estimates.

There are several factors that may violate the stochastic independence. Local dependence is one of the common causes. It can be examined via residual correlations matrices after the extraction of the first factor (Linacre, 1998; Marais, 2009). Some of the Rasch literature suggests that noticeably higher items correlations in the Rasch residual correlation matrix could be an indication of local dependency between items. However, despite the popular use of this method, there is currently no well-documented range of critical values to flag this condition, and for this reason, a variety of cut-off values are reported (Christensen, Makransky, & Horton, 2017). For example, Linacre (1998) advised that an inter-item residual correlation > 0.3 above the average residual correlation as a cut-off to flag local dependency, whereas, Andrich (1988) suggested an inter-item residual correlation ≥ 0.3 as an indication of possible local dependency.

Another cut-off value often reported in the literature is the critical value of 0.2 proposed by Chen and Thissen (1997). However, Marais (2013) holds the view that the critical value will always be relative to the parameters of the specific data sets, thus, the local dependence critical value should be considered relative to the average item correlation, concluding that no single uniform critical value exists. A standard way of accounting for local dependence is examining the recalibration effects on person estimates by combining the dependent items into a single polytomous item (Marais, 2009). The presence of response-dependence typically results in artificially inflating the reliability, Person Separation Index (PSI), thus, recalibration should result in decreased PSI value if local dependence exists (Marais, 2009).

The third requirement, monotonicity, requires that the ICC or the logistic function to accord to certain characteristics. The ICC must be monotonically increasing, such that higher

ability results in a higher probability of success in the item (Bond & Fox, 2015; Hagquist, Bruce, & Gustavsson, 2009). This means that any increase in person ability is always accompanied by an increase in the probability of a correct response on any item. In the case of polytomous items, on average, those with higher scores on a latent variable must also endorse higher categories (Cavanagh, 2009). A very basic way of examining this requirement for a polytomous item is evaluating the category selection statistics for each response category generated by the sample (Bond & Fox, 2015). These category frequencies summarise the distribution of all responses across all categories. Low frequencies are also an indication of insufficient information to calculate category thresholds. Similarly, threshold (or step) calibrations (50:50 point difficulties estimated for choosing one response category over the adjacent category) and category fit statistics could also serve to verify the monotonicity requirement (Wright & Masters, 1982). Guidelines recommend that the thresholds of the categories must be at least 1.4 logits apart to show the empirical distinction between categories, but not more than 5 logits so as to avoid large gaps in the variable (Linacre, 1999). Finally, the item probability curve is a visual display to examine whether polytomous items accord to the monotonicity requirement. Each response category should have a distinct peak in the probability curve display, illustrating that each is indeed the most probable response category for some portion of the measured variable (Bond & Fox, 2015).

The fourth requirement is sufficiency, which states simple sum statistics for an item or person is the sufficient statistic for the item or person parameter (Magno, 2009). This means that the number of items correct contains enough information to estimate person ability, and the item total score contains enough information to estimate the item difficulty. This idea is related to the principle of invariance which is crucial to fundamental measurement (Andrich, 1988). This principle requires that the measurement of persons is not dependent on the items being used for the measurement and the calibration of the items is not dependent on the persons being used for the calibration (Rasch, 1961). Lack of invariance across a priori specified sample groups (e.g. gender) could be evaluated by statistical procedures such as Differential Item Functioning (DIF) analysis. DIF essentially is a procedure which checks whether sub-groups of a norm group score differently on a specific item, given the same location value on the latent trait (Hagquist & Andrich, 2004).

Finally, the main goal of Rasch analysis is to find the non-statistical difference between the observed data and model expectation (Sampaio, Goetz, & Schrag, 2012). The difference is evaluated through goodness-fit chi-square with a non-significant p-value. Additionally, both

item and person fit residual statistics are expected to be normally distributed with mean and standard deviation close to 0 and 1 respectively (Sampaio et al., 2012). However, these statistics are just an approximation of the overall fit, the determination of fit of the data should follow a holistic approach relying on several of the Rasch fit statistics and displays. They include, but are not limited to, item and person fit statistics analysis, threshold functioning, DIF analysis, dimensionality and testing for local dependence. These fit estimation procedures will be explained in more detail in the following chapters.

The Rasch measurement model enables development of psychological measures that are parallel to the physical measures. RMT is believed to be a practical realisation of simultaneous conjoint measurement theory, which some believe is the closest generally accessible approximation of the fundamental measurement requirements for the human sciences. The Rasch approach to measurement provides several statistics and procedures to assess the data and stepwise improvement of the data to achieve fit of the data for interval scaling.

4.5. Summary

This chapter began with a brief review of the progression of older forms of validity into unified construct validity and ultimately its endorsement by popular validity standards organisations. Then, the chapter provided a critical comparison of the two most widely applied instrument development models focusing on their main characteristics. Finally, a brief overview of the Rasch model was provided highlighting the model's requirements to achieve the fit of the data to the Rasch model requirements. The next chapter provides the methodology and methods used in this study.

Chapter 5 - Methodology

5.1. Introduction

This chapter explains the research methodology guiding the study and methods employed. It begins by reiterating the aim and research questions, then providing justification for the use of a quantitative methodology. Next, it explains the research design and the three phases of the empirical research. The instrument development phase is explained first, then the details of the methods and procedures for providing validity evidence, and third, the last phase explains the methodology, procedures, and methods of the correlational analysis. The chapter concludes by describing ethical issues.

5.2. Aims and Research Questions

As described in Chapter 1, the main aim of this research was to develop an objective measure of CS1 student competence commensurate with the principles of contemporary measurement validity theories. The research questions were:

1. Can a measure of student competency in CS1 be constructed?
2. What evidence is available to support an argument for the validity of the project?
3. Are there statistically significant associations between student competency in CS1 and student and classroom learning environment characteristics?
4. What are the consequences of the research for the design and delivery of CS1 instruction?

5.3. Methodology

The most appropriate methodology or paradigmatic view for investigating a research problem rests upon the essence of the phenomena being investigated (Chilisa & Kawulich, 2012). The main part of this study involved testing objective theories or hypothesis as suggested by the research questions. Statistical analysis and scientific methods characterised by a positivistic way of knowledge creation were applied. Therefore, the research was framed in a positivistic epistemology, supported by a quantitative framework of methods, techniques, and procedures similar to the natural sciences (Chilisa & Kawulich, 2012; Fraenkel & Wallen, 2003). However, some constructivist methods were used in some aspects of the instrument development process to provide data on the qualities of CS1.

5.4. Research Approach

This investigation was guided by the construct modeling approach of Wilson (2005), which synchronises well with the validity aspects of Messick (1995), thus, through which the validity concerns can be addressed. Similarly, this model expects to employ a Rasch model centered around analysis of the data, where the model relates students' abilities to items difficulties by placing items and persons on a common scale known as the item-person map – an interval-level scale measured in logits. Therefore, through this approach, the two main research questions could be directly addressed and the eventuating interval-level scale from these two research questions are used in the correlational design to answer the last two research questions.

5.5. Research Design

The development effort of the CS1 student competence measure (CS1 measure) is divided into three phases: (1) CS1 measure development; (2) evaluation of the validity of the CS1 measure development process; and, (3) correlational analysis of relations between student and learning environmental factors with CS1 student competence. Figure 5.1 is a visual illustration of the main activities in each of these phases and their relationships. The connecting arrow between the first two phases was to show that the instrument development phase was informed by concurrent consideration of validity aspects of Messick (1995) as interpreted by Wolfe and Smith (2007a, 2007b), although *post hoc* evaluation of validity was undertaken in a separate phase. Similarly, as the arrow suggests, some of the correlations revealed in Phase three were used to support the validity evidence of the measure.

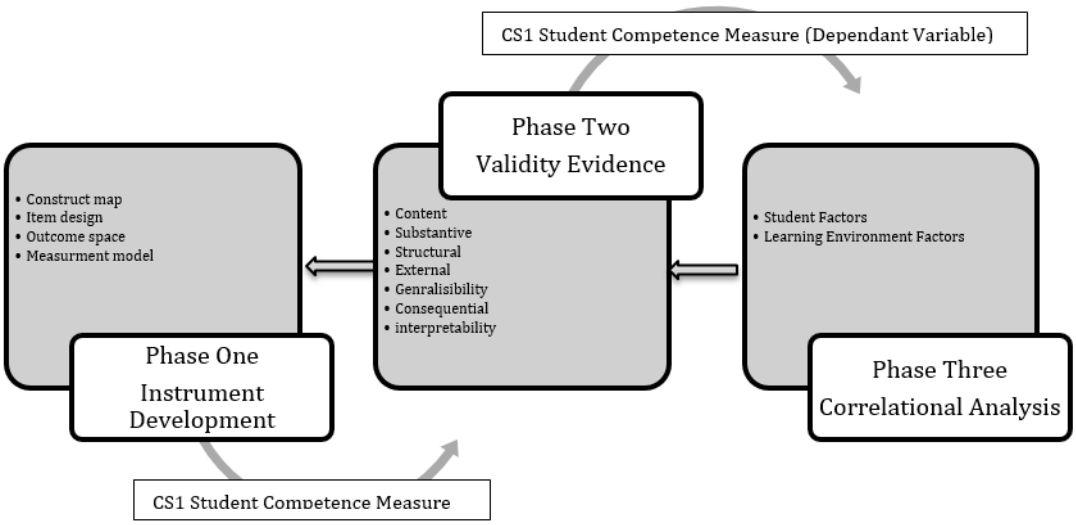


Figure 5.1. Research design

Wilson's (2005) Construct modelling approach is a coherent and integrated framework incorporating four building blocks: (a) construct map; (b) item design; (c) outcome space; and, (d) measurement model. This is an established model developed by the Berkeley Evaluation and Assessment Research (BEAR) group (Wilson, 2005; Wilson & Sloane, 2000). The construct modelling approach takes a developmental perspective on learning, that is, learning is seen to progress along the dimensions of the latent construct of the learner. This means that the approach expects the instrument development to be grounded in a construct theory as deliberated in the construct model of the CS1 student competence construct (Figure 3.3, Section 3.2.5). Similarly, the Rasch approach to measurement development also requires the items to be constructed upon a construct theory of a continuum of increasing difficulty order for creating an interval level, unidimensional measure (Wolfe & Smith, 2007a, 2007b). Finally, validity is related to the inferences made from the test scores of a measure and the use of test scores should reflect the definition of the construct (Wu & Adams, 2007). Thus, there is a clear link between the validity of the measure and definition of the construct in terms of its properties. In other words, defensibility of the validity argument depends on the connection established between the measure's score interpretation and its underpinning construct theory. Therefore, the instrument development approach, the instrument construction model, and the validity model have a common foundation and the outcomes are complementary to each other.

5.5.1. Phase one: CS1 measure development

The activities of an investigation into the development of CS1 competency measure was based on the four building blocks of as construct modelling approach as exemplified by Wilson (2005). It begins by developing a construct map which is a hypothetical depiction of students' increasingly sophisticated conceptions of knowledge building over time (Wilson, 2005). The construct map guides the design of the assessment items as well as postulating the range and sophistication of responses to the items. This is termed the outcome space for an item. A statistical measurement model such as the Rasch model is used to empirically verify consistency between what is observed (outcome space data) with what is measured (construct map) (Wilson, 2005). The link between these building blocks, as illustrated in Figure 5.2, suggests that the instrument development process is an iterative process, meaning, it is a cycle that may be repeated several times until the desired outcome is achieved. Validity has been incorporated in the model to illustrate how development activities are informed by concurrent consideration of aspects of validity evidence.

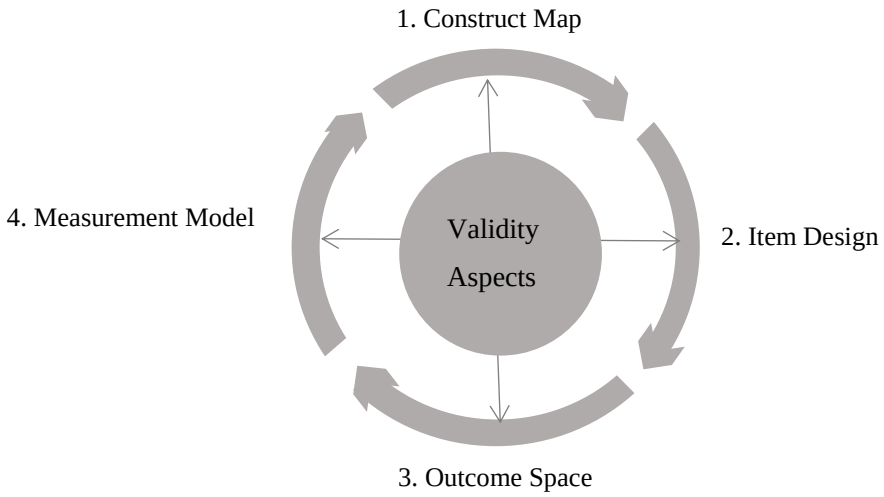


*Figure 5.2.*Wilson's construct modelling approach incorporating validity aspects

5.5.2. Building block 1: Construct map

Every new instrument development effort must begin with specifying the intended purpose, followed by an intense literature review to establish the background knowledge about the study's content and intended inferences (Wilson, 2005; Wolfe & Smith, 2007a). This helps to establish underlying theories behind the construct of interest and related constructs and their characteristics already established in the literature. Therefore, Chapter 3 was dedicated to establishing the construct of CS1 student competency and its characteristics. A construct model illustrating the key elements and their characteristics was proposed in Figure 3.3 of Section 3.2.5. The model was used as the kernel to develop the construct map (See Appendix I), which describes the hypothesised developmental trajectory or the expected learning outcomes for each topic. Figure 5.3 provides the construct map developed for the third construct – loop structure. The response to items was based on common tasks found in common CS1 evaluative tools (See Gluga et al., 2012a; McCracken et al., 2001) and introductory programming books (Deitel & Deitel, 2010; Hubbard, 1999; Johnson, 2012; Kochan, 2015; Streib & Soma, 2014).

Students who are able to write a code segment implementing multiple nested loop structure integrating the concept of IF/Else structure to achieve a given task	Level 4 Extended Abstract	High	Write a code segment implementing multiple nested loop structure integrating the concept of If/Else structure to achieve a given task
Students who are able to explain by summarising the purpose of a given code segment implementing one-level nested loop structure	Level 3 Relational		Explain by summarising the purpose of a given code segment implementing one-level nested loop structure
Students who are able to trace the output of a given code segment implementing one-level nested loop structure	Level 2 Multistructural		Trace the output of a given code segment implementing one-level nested loop structure
Students who are able to trace the value of loop control variable in a given code segment implementing non-nested loop structure	Level 1 Unistructural	Low	Trace the value of loop control variable in a given code segment implementing non-nested loop structure

*Figure 5.3.*The construct map for loop structure (construct 3)

5.5.3. Building block 2: Item design

Item development requires decisions about the format of questions, e.g. multiple choice, short answer or essay type. Item development was informed by standard educational test development guidelines (American Educational Research Association, American Psychological Association, and National Council on Measurement in Education and Joint Committee on Standards for Educational Psychological Testing, 2015).

One of the purposes of the construct map is to guide the item development, which must align and reflect the construct irrespective of item formats (Wilson, 2005). The construct map represents the hypothetical responses expected in relation to an item given a specific level of understanding. Thus, an item can be designed to measure a single or multiple levels of a construct. The items for each construct in this study followed a uniform format where all (A) parts are the basics, the (B) parts are tracing, all (C) parts are explain questions and all (D) parts are writing questions. All (A), (B) and (C) questions were designed to measure a single level of the construct, whereas the writing questions were designed to measure at multiple levels. The expected general hierarchy of responses for each of the writing items with their corresponding SOLO levels is shown in Appendix II. The fundamental principle applied in structuring the rest of the questions is drawn from Collis, Romberg, and Jurdak (1986) in their mathematical problem-solving instrument, in which the questions began with a stem item followed by a series of questions in hierarchical order of difficulty. The (A) questions of the

stem were tracing questions (easiest level) established at unistructural level complexity as described in the construct map (Appendix I), whereas, the (B) questions were slightly more difficult than the (A) questions, and then the (C) questions were the most difficult of the stem. The main idea of using a stem format was to save time; students need not start with a new problem that requires reading to create the mental model for each part, rather they build upon previously accomplished tasks. This is important when the time factor is a constraint as in this investigation. As the CS1 measure is a cognitive test, it is important to make sure that students do not leave questions unattempted because there was too much to read and process. Therefore, the required answers for each question were short, requiring a single value or few words except for the code writing questions (all (D) parts), in which students were required to write a maximum of eight lines of code. The writing Questions were matched up with the highest of the SOLO levels. This format used in the overall designing of the test and the question types can better assess the range of skills required and represents typical question formats found in many of the BRACElet publications.

The advantage of having a variety of question formats is that no student would be disadvantaged or advantaged by homogeneity and guessing that occurs in some formats such as Multiple Choice Questions (MCQ). Additionally, it is virtually impossible to make MCQ choices truly random without a pattern of right or wrong (Poundstone, 2014), and guessing may result in an inflation of the scores in the less proficient students (Marais, 2014). A sample set of questions developed for the topic loop structure written in Java programming language is shown in Appendix IV. The same uniform format was followed for the rest of the topics. To enhance readability, all program writing questions were illustrated either with diagrams or tables. This is a common practice found in many programming exercises of introductory programming textbooks and BRACElet project publications. The item development process resulted in a 20-item (5 constructs x 4 questions for each) test, which was then reviewed and pilot tested to assess whether the items captured the competencies characterised in the construct map.

Review: After item development, the questions were reviewed with four students. Two of them were high school students then studying ATAR (Australian Tertiary Admissions Rank) Computer Science, and two second year CS students studying Computer Science. The questions were reviewed with them on a one-to-one basis; based on their responses, some items were reworded. Then, the Expert Review Group (ERG) screening of items on the CS1 instrument was obtained from four CS1 lecturers, each of whom had over five years of

experience in teaching CS1. As the expert group members were in different geographical locations, the construct model and the questions developed were emailed. The experts were asked to review the questions against the construct map to examine the match between the competencies defined and the questions written to achieve each competency defined in each cell of the construct map. The review questions and responses by one of the experts are in Appendix III . The main goal of the review was to determine the quality of items and to collect evidence on the hypothesised hierarchical ordering of items. The reviewers were asked to comment on three aspects: (a) whether the task or the questions match with the competencies defined in the construct map; (b) whether there was a learning path in the items written for each sub-construct as hypothesised in the construct map; and, (c) the appropriateness of accompanying diagrams. Specifically, they were required to make comments if they could not agree on an item with respect to these aspects.

The assumed difficulty hierarchy matched with the experts' ratings except for Question 2C, which was subsequently replaced. Furthermore, experts also suggested some changes to the visual illustration of Question 1D and 5D. Based on their advice, the visual illustrations of some questions were improved, some questions were re-phrased, and Question 2B was replaced with another question based on the suggestions of ERG, which was subsequently reviewed by ERG. After the amendments, the instrument originally written in Java was translated to C++ and Python to accommodate the different programming languages taught in the targeted institutes for data collection. One of the expert group members who had extensive experience teaching computer programming in Python and application development, evaluated the Python version for any possible issues and disparities between the translations. Similarly, the C version was checked by one of the experts from the panel who had taught CS1 in C language for more than 10 years. There was little difference between the Java and C version as the researcher tried to avoid using language dependant concepts such as Input/output procedures. In languages like Java, I/O procedures are more difficult to implement as opposed to C and Python, therefore, I/O was intentionally avoided to minimise bias. The item set designed for the topic loop structure written in Java programming language is attached in Appendix IV.

Pilot Testing: The item technical quality involves aspects such as unambiguous phrasing, accurate answer keys and suitable reading levels for the target population (Messick, 1996). The main goal was to empirically test the quality of the items constituting the measure, appropriateness of the test format, and adequacy of the construct coverage (Wolfe & Smith,

2007a). Therefore, the 20-item test was piloted to gauge the interpretation of the questions by the participants (Creswell, 2012) with 10 students selected from the Asia Pacific University of Technology (APU), Malaysia. These students were at the time studying in the second semester of their CS degree. They were selected using convenience sampling from which some volunteered to do the test. Use of a convenience sample is generally acceptable for pilot testing (Wolfe & Smith, 2007a). The test was administered in a test setting by a colleague of the researcher. The students were asked to provide a reason if they left a question unattempted. This was to evaluate the content representativeness.

The main goal of the pilot testing was to evaluate whether the responses of items suggested the same interpretation and hierarchy as hypothesised in the construct map. Given responses accord this hierarchy, students with relatively higher abilities in the observed latent trait should have answered more items correctly than those with a lower ability, and consistent with that, more difficult items should be answered correctly less often than easier ones. The questions for each construct were hypothesised to form a hierarchy in that the first question (example Question 1A) is easier than the question next (example Question 1B) and so forth.

The responses received for each question were first entered into an excel sheet and assessed as to whether the responses for each construct formed a structure somewhat similar to the Guttman scale (Guttman, 1950). The Guttman scale is used to measure an increasing amount of "attitude" or "learning" towards a latent trait similar to what has been hypothesised in the construct map. While the majority of the question responses were shown to follow the postulated hierarchy, Questions 1A and 1B were shown to be disordered. Similarly, disordering was also observed between 2A and 2B of question 2. Question 3B, 4A, and 5A were assumed to be badly worded because very few students answered these questions, and some provided feedback that they did not understand these question very well. To avoid confusing the participants, Questions 3B and 5A were reworded, and the stem of Question 4 was completely replaced by a similar, but more concise question after consultation with the expert panel members. Then the instrument was translated into other programming languages; the programming languages of the targeted institutes. A noteworthy point here is that this translation is not same as translating an instrument from one human communication language to the other. The segments that had been translated were only the code segments resulting only in a syntax change. It is also important to emphasise, unlike human communication languages where a sentence or word is subject to different interpretations, the different programming language constructs are objective that conveys the same meaning although the syntax may be

different. For example, the following two code segments written in Java and python as shown in Table 5.1) would result in the same output regardless of the person who interprets it (Output is 5 ,2, 1, 0). Therefore, the issues prevalent in other psychological instrument translations are not relevant in this situation.

Table 5.1

Comparability between Java and Python Code for the Same Task

Java	Python

`int x = 11;`
`while (x != 0) {`
`  x /= 2;`
`  System.out.println(x);`
 	
`}`
	`x = 11`
	`while x != 0:`
	`  x /= 2`
	`  print x`

5.5.4. Building block 3: Outcome space

The outcome space is where the researcher makes inferences from the responses to items by categorising and attaching scores to the range of responses received for each item. The construct map defines the properties of the different levels of knowledge of the construct qualitatively, including what students know and are capable of doing with said knowledge (Wilson, 2005). Thus, it helps in categorising the different levels of sophistication in the responses. In short, it is the scoring model for an item that maps the student responses onto levels in the construct map (Wilson, 2005), thus it can be viewed as a specialised version of the construct map.

To score responses for basics, tracing and explaining questions (that is, all parts [A] [B] & [C]), a dichotomous scoring model (0 for incorrect or 1 for correct) was applied. In other words, these questions were designed to measure a single level of the construct, whereas, the writing questions were designed to receive multiple responses to be measured at multiple levels. All part (C) tasks required explanation, so for these questions, if the students were able to explain the main idea, the response was marked as correct (1), otherwise was marked as incorrect (0). Writing questions, all (D) parts, were scored at multiple levels using a scoring rubric of four SOLO levels as detailed in Appendix II. The rubric was developed by drawing ideas from the CS1 research literature, where (see Clear et al., 2008; Ginat & Menashe, 2015; Izu et al., 2016) SOLO taxonomy had been applied to evaluate and categorise student responses to program writing tasks. Clear et al's., 2008's SOLO categories were found to be particularly

useful, and accordingly, was used as the main reference for developing the different response categories for program writing questions. Outcome space is also related to assigning numerical scores to the response categories. Therefore, each category of the rubric was assigned a numeric value suggesting the level of competency each student had achieved as defined in the construct map.

5.5.5. Building block 4: The measurement model

The fourth building block is the measurement model, and its objective is to link the scored data with the construct map (Wilson, 2005). A statistical model such as a Rasch model could be applied to the scored data to empirically verify the learning trajectory postulated in the construct map. There are two main types of measurement models: Classical Test Theory (CTT) and Item Response Theory (IRT). The merits of IRT based models to construct measures have been presented in Chapter 4, with special attention paid to the qualities of the Rasch model for instrument construction. Coincidentally, the construct modeling approach also employs a Rasch model-based analysis at the item level (Embretson & Reise, 2000; Hambleton & Jones, 1993; Wilson, 2005). Rasch modeling relates the students' abilities to item difficulties by placing items and persons on the same scale called an item-person map measured in logits. This scale is an aggregation of all the students' proficiency levels in relation to all the item difficulties so that visual comparisons can be made directly. Therefore, to verify the relationships assumed in the construct map, data was collected by administering the 20-item CS1 measure to a sample of 85 students as delineated below.

Participants: The sample comprised of 85 students (25 [Maldives National University (MNU)], 31 [Asia Pacific University of Malaysia (APU) and 29 [Villa College, Maldives]). Attempts were also made to recruit participants from two universities in Srilanka and another university from Malaysia. However, due to a lack of cooperation, these universities had to be excluded. Similarly, two other institutes from Maldives had also been contacted, but none of these institutes replied to the correspondence. Therefore, due to the time and budget constraint the participant's selection had to be restricted to the above-mentioned institutes only. The students selected had completed the CS1 and had just begun the second semester of the first year of university study. Each of these groups was instructed with a different choice of programming language. Of the total invited (all the students then were enrolled into CS1), only three students declined to participate. The characteristics of these participants are shown in Table 5.2. The age factor was not considered as the sample consisted mainly of students aged between 18 – 20 years. Total population sampling was used because the sampling population

was relatively small. An ideal sample for a Rasch analysis parameter estimation is according to some studies above 100 (Chen et al., 2014). Due to the limited number of students studying for CS degrees, the sample was constrained to the maximum achievable from the three institutes. However, a significant advantage of this sample is that it was not a subset of the sample; thus, the possibility of bias occurring was reduced due to the sampling technique. Sample selection and the sample size was influenced by several factors including the difficulty in gaining access to conduct the research in other venues, time and budget constraints.

Table 5.2

Characteristics of the Subjects (N=84)

Characteristics	Categories	No
Gender	Male	70
	Female	6
Institute	APU (Instructed with Python)	31
	MNU (Instructed with C++)	25
	Villa (Instructed with Java)	28
Year 12 or 10 Mathematics Completed	Yes	35
	No	50
Studied CS1 in High School	Yes	25
	No	59
Stream of Study	Commerce	25
	Science	57
	Arts	2
Computer Prog Exp of at least 6 months	Yes	35
	No	50

Data collection: Prior to the test administration, approval was sought from each institute in writing (see Appendix VIII). Each institute arranged a date and a test venue and informed the students in advance the nature, the main topics, and purpose of the test. The students were asked to be present if they wished to take part in the study.

First, the final 20-item CS1 measure (C version) was administered to 25 students of Maldives National University which was the first venue of data collection. This included all the students studying CS1 at the time, none declined. The test was conducted under similar protocols as university exams in a designated place under the supervision of the researcher and an administrative staff from each institute. Before the test administration, informed consent was sought and it was reiterated that participation was voluntary and participants had the option to decline at any point during or before the test administration. The participants were given one

hour to complete the test. Second, the Java version of the instrument was administered to 28 students of Villa College; no one declined to do the test. Then the final round of data was collected from Asia Pacific University of Technology (APU) by administering the Python version of the instrument to 31 students. Three students from APU chose not to participate in the test. The same administrative protocol was followed in each of these institutes. After completion of the test, the students were also asked to complete a 15-minute survey developed to collect student and learning environment factors (Table 5.3).

The student responses to each question of the 20-item CS1 instrument were marked and scored in accordance with the predefined scoring models described in the outcome space phase. Several randomly selected questions were checked by one expert panel member – a Ph.D. student at Curtin University with experience in teaching computer programming for first-year Computer Science (CS) students. This was to ensure the marking was consistent with the scoring model. In cases where there were discrepancies between the researcher's marking and the expert member marking, the responses to that particular question by every student were again reviewed and re-scored accordingly. As expected in any survey or test administration, some blank responses were received which were subsequently coded as 9 (missing data) and the rest were coded with the categories described previously. The highest level of hypothesised writing proficiency level (Extended Abstract) was not used in scoring writing questions as none seemed to reach that level. Data were then entered into an excel spreadsheet and then converted into a non-delimited text file. In reporting the results the following procedures of the Rasch Analysis suggested by Sampaio et al. (2012) have been used for evaluating the fit of the data to the Rasch Model requirements.

The software: Data were analysed using the Rasch Unidimensional Measurement Model (RUMM2030) computer program (Andrich, Sheridan, Lyne & Luo, 2011). Additionally, IBM's SPPSS version 24 was used to conduct ANOVA and other related analysis presented in phase three of the study.

Rasch model: The Rasch Partial Credit Model, also known as an unconstrained polytomous model, was applied as the response structure of the CS1 measure differs across the items.

Sample size: To a certain extent, the sample size depends on its intended uses, as different levels of precision may be acceptable for different applications. Currently, there is little agreement about the size of the sample (Nguyen et al., 2014), but the literature suggests

general guidelines to obtain a robust parameter estimation (Chen et al., 2014; Linacre, 1994). The literature warns that for some tests, such as the goodness-of-fit Chi-square test, too small a sample could result in unstable results and may jeopardise the generalisation of findings (Sampaio et al., 2012). Likewise, with a larger sample, a slight deviation from the Rasch model may result in significant Chi-square values (Sampaio et al., 2012). In principle, the sample distribution plays the key role in the sample size. It has been shown that a sample size as small as 100, but well distributed evenly across the trait levels of interest achieves 99% confidence of the person being ± 0.5 logits (Linacre, 1994). Therefore, a well-distributed small sample achieves more robust parameter estimation than a larger sample that is off-target.

Global or overall fit: Global fit statistics give an overall view of how well the observed data from the instrument fits to the Rasch model expectations. The chi-square (X^2) statistics (the item-trait interaction) is the main global fit statistic available when Rasch analysis is performed using RUMM2030 software. When the data fit the model, the chi-square has a probability value greater than 0.05. However, this value should not be taken at face value instead the chi-square statistic is sensitive to the sample size. That is, when the sample size is too large almost any small difference will appear to be statistically significant. Additionally, both the individual item and the person fit statistics should also not significantly deviate from the model; this is explored via items and person fit residuals. Fit residuals are expected to be between ±2.5, and M ±SD item and person approaching 0±1 (Pallant & Tennant, 2007; Tennant & Conaghan, 2007).

Internal consistency reliability: The purpose of this test is to assess the extent to which the items distinguish between the different levels of groups – different levels of student competencies. The test is measured by the Person Separation Index (PS1). The PSI is equivalent to the test reliability of person separation, also called the reliability case of estimates according to Wright and Masters (1982). In RUMM2030, the test is based on estimated person locations, in some ways similar to traditional Cronbach's alpha reliability. PSI greater than 0.70 is satisfactory for group comparison and PS1 greater than 0.85 is required for individual comparison (Fisher, 1992; Pallant & Tennant, 2007; Tennant & Conaghan, 2007).

Response category ordering (thresholds ordering): The purpose of this test is to assess whether participants use categories consistent with the metric estimate of the underlying construct (Sampaio et al., 2012). Disordered threshold occurs when the level at which the likelihood of failure to agree with or endorse a given response category (below the threshold) turns to the likelihood of agreeing with or endorsing the category above the threshold (Bond &

Fox, 2013). Disordered thresholds indicate that participants were not able to discriminate between some of the scoring categories defined in the rubric. The ordering of thresholds is examined graphically via the item category probability curves. Collapsing adjacent categories with poor discrimination is a justified procedure to treat the condition, however, collapsing may result in losing the data structure if the data fits the model.

Local independence: Local independence refers to the variance explained in residuals after the Rasch factor has been removed; a very minimal association between the items is expected (Wright, 1996b). This is examined by estimating the correlation between residuals – after taking the Rasch factor – between the items, where correlation coefficients between residuals are higher than 0.30, or noticeably higher (Wright, 1996b). Some literature suggests that residual correlation between the items should be no more than 0.20 above the average correlation (Marais & Andrich, 2008). A general approach suggested by Marais and Andrich (2008) to account for local dependency violation, is to combine the items that reveal noticeably higher residual correlations into high order polytomous items and compare the reliability with that provided by the individual items. The presence of response dependence tends to increase the reliability, thus, a higher reliability in the former could mean the violation of local dependence, given no multidimensionality exists (Marais & Andrich, 2008).

Item bias: Item bias, commonly known as Differential Item Functioning (DIF), exists when the probability of endorsing an item differs for individuals who have the same level of ability but belong to different groups (Smith, 2002) such as female and male. This is examined by DIF analysis –a test of variance (ANOVA), which assesses whether there is a significant difference between the groups. Uniform DIF is indicated by a significant main effect for the person factor (example: gender), the direction of the difference is consistent for persons irrespective of ability scores (Pallant & Tennant, 2007). Where two or more items present uniform-DIF, it may be corrected by grouping items with DIF into one group and comparing them with the rest of the items if they cancel out (Tennant & Pallant, 2007; Wainer & Kiely, 1987). However, in the case of Non-uniform DIF, usually, the items are removed according to Sampaio et al. (2012). Sampaio et al. (2012). However, a more recent approach is splitting the item if this is meaningful to the construct is as common (Hagquist & Andrich, 2017). Person factors taken into consideration are shown in Table 5.2.

DIF items were checked after Bonferroni correction. Both Bonferroni and Benjamini-Hochberg procedure are classical approaches used to counteract the risk of Type I errors (the higher the chance for a false positive; rejecting the null hypothesis when it is not) on multiple

testing (Kim et al., 2017; McDonald, 2009). These classical approaches were found to have been used in many of the studies in the Rasch literature (Bland & Altman, 1995; Tennant & Pallant, 2007) when performing a large number of statistical tests as in DIF analysis. The main assumption in performing this procedure is that some will have *P* values less than 0.05 purely by chance even if all the hypotheses are really true. Literature suggests this procedure is most appropriate when: (a) there are a fairly small number of multiple comparisons and interest is to identify few cases that might be significant, and (b) the same test is repeated in many subsamples, such as when stratified analyses (by age group, sex, income status, etc) are conducted without an a priori hypothesis that the primary association should differ between these subgroups (Perneger, 1998). As the DIF analysis is a stratified analyses, some of the Rasch literature (Bland & Altman, 1995; Tennant & Pallant, 2007) shown to embrace Bonferroni adjustments in DIF analysis. However the downside of this approach is, although it protects from Type I Error, it is vulnerable to Type II errors (failing to reject the null hypothesis when in fact the null hypothesis should be rejected) resulting truly important differences are deemed non-significant (Perneger, 1998).

Unidimensionality: Just as for local dependency, unidimensionality is a basic RMT requirement. The purpose is to ascertain if each item of the test measures a single construct. Unidimensionality of a CS1 measure is tested by Principal Component Analysis (PCA) of the Rasch residual data (Smith, 2002). In PCA procedure, two sets of items are defined by taking the items with the highest positive and highest negatively loaded items on the first residual factor. Then a series of t-tests are conducted comparing the person locations of the two sets of items. Strict unidimensionality is achieved if less than 5% of the t-tests are significant, or the lower bound of the binomial confidence overlaps by 5% (Smith, 2002; Tennant & Pallant, 2006).

Targeting of items: Targeting examines the spread of CS1 scale items across the continuum (sample) by comparing the person locations with item locations. Items of a well-targeted measure cover the entire range of the sample across the construct being measured (Sampaio et al., 2012). Additionally, the M ± SD locations of the person approximate the item location (0±1 logits) (Sampaio et al., 2012; Soh, Barker, Morello, Dalton, & Brand, 2016). Targeting is important because the precision of estimates of an individual's location depends on the severity of items corresponding to the distribution of the student's ability in the targeted sample.

Measure calibration and refinement: The choice of measurement model depends on factors such as sample size and item format dimensionality (Duckor, Draney, & Wilson, 2009). A comprehensive review of the Rasch model and its benefits in educational measurement development has been explained in Chapter 4. The Rasch partial credit model (Wright & Masters, 1982) was applied to the data as it was developed for item formats with a non-uniform response structure similar to the CS1 measure format. Analysis followed an iterative process in which each iteration of the construct modeling building blocks was re-visited to examine the consistencies between these elements. For each iteration, some aspect of the measure was improved until the desired outcome was achieved.

To begin calibration, the delimited text file was transferred into the RUMM2030 computer program; the Rasch unconstrained partial credit model was applied to the initial data set to examine the fit between the response data and the Rasch model's expectation of the data. The first application of RUMM2030 revealed disordering thresholds in questions 3D and 5D, suggesting that the assumed hierarchy in the construct map had some flaws. These items were corrected by collapsing the middle categories of each question before further analysis. The refined measure was further analysed by a final iteration of the Rasch analysis to calibrate both items and persons on the same logit scale. Then, the psychometric properties of the measure were tested against Wright and Masters' (1982) measurement criteria to establish whether the measure manifested these criteria.

To consider whether observed data constitutes a measure, Wright and Masters (1982) deliberated a four criteria evaluative benchmark to test the instrument data. These manifestations have been described by Cavanagh, Waldrip, Romanoski, Dorman, and Fisher (2005) as follows:

- Unidimensionality – the reduction of experience to a one-dimensional abstraction (height, weight, intelligence);
- Qualification – more or fewer comparisons among persons, items, etc. (taller or smaller, heavier or lighter, brighter or duller);
- Quantification – a unit determined by a process which can be repeated without modification over the range of the variable (feet, inches, pounds, logits); and,
- Linearity – the idea of linear magnitude inherent in positioning objects along a line by some device or instrument (tape measure, scale).

Application of the Rasch measurement model for scale calibration attains all four criteria given the data fits the model expectation by positioning both persons and items on the same linear scale (Cavanagh & Romanoski, 2008). Analysing data with the Rasch model provides statistics and graphical displays to demonstrate each of these criteria. For example, to investigate the unidimensionality of the measure, the Principal Component Analysis (PCA) of the Rasch residual (the difference between the observed and predicted scores) was conducted to investigate whether the residual contains any meaningful structure beyond noise level data. To provide additional evidence for unidimensionality of the measure, a t-test protocol suggested by Smith (2002) and Tennant and Pallant (2006) was conducted. Furthermore, local dependency testing and DIF analysis were also performed to provide evidence of the stochastic independence of the items, which is considered another fundamental requirement of the Rasch model related to the unidimensionality of the measure. The qualification criteria require that both items and persons be able to be compared in a consistent way. Qualification requirement is met if the latent trait of interest leads to different responses to the items by the participants in accordance with their trait levels, which was demonstrated via item/person fit statistics and graphical displays such as item probability curves. The main aim of the Rasch measurement model is to develop measurement units similar to the physical sciences that is repeatable along a scale (Bond & Fox, 2013). Rasch outputs such as item-person threshold maps and item-person maps, where person abilities and item difficulties are plotted on the same scale calibrated in logits, were used to demonstrate the quantification criteria. Finally, data fit to the model statistics were generated to confirm an interval-level scale had been achieved.

5.6. Phase Two: Validity Evidence

The current notion of validity is an investigative process to evaluate the appropriateness of interpretation, uses and decisions based on the outcome of the measure (Wolfe & Smith, 2007a). The methodology was basically the application of a group of activities suggested by Wolfe and Smith (2007a) which was founded on the unified concept of validity from the Standards for Educational and Psychological Testing (AERA, APA & NCME, 1999) and the terminology and classification system proposed by Messick (1989, 1995) to evaluate the investigation process and outcomes constituted in the investigation. The activities are consistent with the guidelines for measurement specified by the American Educational Research Association, the American Psychological Association, and the National Council on Measurement in Education (2015). These aspects were:

1. Evidence of the content aspect;

2. Evidence of the substantive aspect;
3. Evidence of the structural aspect;
4. Evidence of the generalisability aspect;
5. Evidence of the external aspect;
6. Evidence of the consequential aspect; and,
7. Evidence of the interpretability aspect.

The methodology was basically an evaluation of the intentions, the processes and outcomes of the instrument that constitutes the instrument development phase against the seven indicators instrument development activities exemplified by Wolfe and Smith (2007a, 2007b). Rasch model statistics and graphical displays generated by the computer application RUMM2030 into data were selectively used to exemplify all six aspects of the framework, in addition to the relevant data from the development process.

The content aspect of the validity concerns the relevance and representativeness of the content upon which the items are developed and the technical quality of the items are established (Wolfe & Smith 2007). One way to provide content evidence is to provide clear statements of purpose by specifying the research questions (Wolfe & Smith 2007), which was made explicit by the research questions and aims presented in Section 5.2. Evidence of the content aspect is mainly concerned with the relevance and representativeness of the content upon which the items constituting the instrument are based upon and their technical quality (Wolfe & Smith 2007). This was demonstrated through the CS1 literature, and CS1 curriculum analysis during the construct and construct model development. Additionally, multiple reviews by the expert panel, pilot test data and item fit statistics of the final CS1 measure were used to demonstrate this aspect in addition to demonstrating the scoring and scaling model used.

The substantive aspect of validity refers to the theoretical rationales for the observed consistencies in the data (Wolfe & Smith 2007). This aspect of validity was substantiated, by examining the theoretical meaning of the item hierarchy against the construct model representing the CS1 student competence construct. Mainly Rasch Fit statistics and displays were used to substantiate this aspect. The literature on competence in CS1 confirms that mathematical ability and previous programming experience are the most consistent factors associated with competence in CS1. Therefore, item-person threshold distribution graphical displays of CS1 competency test data for these different groups of students were generated to demonstrate consistencies between the existing literature and CS1 measure data.

Evidence of the structural aspect of validity confirms the internal structure and dimensionality of the construct model (Wolfe & Smith 2007). One approach to establishing structural dimensionality is the application of the Rasch Model to test whether the Rasch extracted dimension adequately accounts for much of the non-random variance in the data. Therefore, a PCA of Rasch residuals and t-tests were used to confirm this aspect in addition to local dependency testing and item/person fit analysis.

The generalisability aspect of validity refers to the extent to which the performance of measure remains consistent across different measurement contexts (Wolfe & Smith 2007). This aspect can be partly explained by the specific objectivity requirement of the Rasch Model (Cavanagh, 2009). DIF analysis was performed for different demographic groups to demonstrate the items function was as expected, irrespective of the persons attempting them. Furthermore, internal reliability estimates for persons and items derived from the response data was undertaken by administering the instrument to students from three different institutes to reinforce the validity of this aspect.

The external aspect of validity relates to the extent to which the results are convergent when multiple methods are used to measure the same trait (Cavanagh, 2009). This has been traditionally known as convergent and discriminant validity, and also "includes criterion relevance and the applied utility of measures" (Wolfe & Smith, 2007a, p. 99). A list of procedures was suggested by Wolfe and Smith (2007b) to assay the external aspect of validity. However, many of these procedures require instruments of a similar nature to carry out the testing. Instruments of a similar nature are limited and difficult to access, as discussed in the literature review. Messick (1989) suggests that initial evidence of the external aspect of validity can be assessed by examining whether the measure developed was able to classify the person groups as postulated in the developmental models. The construct of CS1 student competence was operationalised to consist of four programming skills forming a learning trajectory as proposed in Figure 3.3 (Section 3.2.5). Separation statistics such as the Person Separation Index (PSI) of the measure can be used as an index to differentiate between person groups in Rasch measures (Linacre, 2014). Therefore, PSI was used to investigate whether the sample of the study can be differentiated into the number of competency levels as deliberated in the construct model.

The consequential aspect of validity addresses the consequences of score interpretation as a basis for action as well as the actual and potential consequences of using the test scores, particularly identifying sources of invalidity such as bias, fairness, and distributive justice

(Dimitrov, 2014). This is because there is no explicit way to determine the consequential aspect of validity in the Rasch model. Therefore, arguments were drawn from DIF analysis to support the argument of fairness that items are not biased in favor of a particular group. Similarly, the item-person map was used to reveal many sources of validity including whether items are distributed fairly and targeted to all ability levels of the participants.

The interpretability aspect of validity is "the degree to which the meaning of measures is clearly communicated to those who want to interpret the measures" (Wolfe & Smith, 2007b, p. 227). The Rasch item-person map and item-person threshold distribution graphical displays were used as the main source to communicate interpretability of the results. For example, the item-person map was used for comparison of both item difficulties and scores of individual students. Similarly, the item-person map was also used to draw empirical evidence regarding how well the observed data of the instrument matched with theory deliberated in the developmental models of the investigation.

5.7. Phase Three: Correlational Analysis

There are two purposes to this phrase: (1) to provide evidence to support some aspects in the *post hoc* evaluation of the validity of the investigation process and outcomes; and, (2) to examine the effects of the commonly reported student demographic variables and learning environment variables on CS1 student competence. In particular, the study aims to answer some of the long-held views and theories about programming language choice for CS1 instruction. Although several studies of the past have tried to answer this question, none had employed the interval-level scores of students as dependent variables. Furthermore, the data collected from this phase was also used to evaluate a DIF analysis.

To ascertain whether there are differences among various groups, a correlational analysis of the associations between student attributes and learning environment variables with CS1 student competence was conducted. The methodology was basically an explanatory correlational design, which explains and clarifies the degree of association between two variables one point in time. (Creswell, 2012). As identified in the literature review (Chapter 2), several factors were shown to associate with student competence. However, due to the unavailability of measures founded on stringent measurement theories, only those variables easily measured without the use of psychological instruments were selected to study.

5.7.1. Sample, instruments and data collection

Data was collected using a researcher-developed brief survey instrument administered to the same sample as detailed in Phase one. The independent variables captured are shown in Table 5.3. CS1 empirical research literature as discussed in Chapter 3 revealed several factors pertaining to student competence. However, only a few were selected for this study due to a lack of availability of instruments based on stringent measurement theories that would result in interval-level data. The main argument advanced in this investigation was the questionable validity of parametric analysis on the summed up scores of CTT based instruments and need for interval-scores. Therefore, employing raw scores contradict the whole notion of the current investigation, thus, the variables chosen for the study are those, which do not require special instruments. Some variables such as the programming paradigm and programming environment were unable to be quantified due to the confounding variables. A few years back students were usually tied up with Integrated Development Environments (IDE's) provided by the institution in the computer labs, whereas now students have access to a variety of IDE's to write and debug their programs in their personal computers and online websites. Therefore, the effect of the programming environment and, programming paradigm initially listed for testing was removed from the learning environment factors. Similarly, due to the limited number of female students studying CS courses in the selected institutes (about 4 students), the variable was dropped from the list of factors. Similarly, the age

Table 5.3

Independent Variables of the Study

Factor	Description	Variable Type
Programming language used for CS1 instruction	Java C Python	Categorical
Prior Computer Programming Experience > 6 months	Yes No	Categorical
Year 10 or year 12 Mathematics Background	12 10	Categorical
High School Stream of Study	Commerce Science	Categorical
Studied CS at High School	Yes No	Categorical

The application of Rasch model in phase two showed that the data fit to the Rasch model requirements, thus student competency scores could be assumed as interval. Additionally the scores showed manifestation of measurement criteria of Wright and Masters (1982) and evaluated for validity against Wolfe and Smith's (2007a, 2007b) validity framework. Therefore, these scores could be used for parametric analysis, such as analysis of variance (ANOVA) confidently without having to assume linearity as in raw scores (Boone & Scantlebury, 2006). In normal practice, an assessment of the normality of data is conducted to confirm the underlying assumption in parametric testing (Vickers, 2005). However, as argued in Chapter 2 and Chapter 3, there is no relationship between normal distribution and level of scores obtained from data, therefore this procedure was not conducted.

5.7.2. Data analysis

The characteristics of the sample and the total number of students in each category are shown in Table 5.3 (Section 5.7.1). A one-way ANOVA was conducted to examine whether there was a statistical significance between various student characteristics and learning environment variables with student competence scores. Since one-way ANOVA and t-tests are equivalent with two groups (produces same p-values and $p = t^2$), one-way ANOVA was chosen for significance testing as it had the advantage of avoiding type 1 error in cases where there were more than two groups. Apart from statistical significance through p-values, it was also important to quantify the strength or the effect of the difference between the two means (Creswell, 2012). Therefore, the effect size was calculated using partial eta squared (η^2) and, in cases of significant p-values with more than 3 groups, post-hoc comparisons using Tukey HSD test were conducted. The SPSS software was mainly used to run these tests; however, a few displays of RUMM2030 were used as well to demonstrate some aspects of significance.

5.8. Ethical Issues

First approval to conduct the study was sought from Curtin University Human Research Ethics Committee (See Appendix V) followed by the individual institutes involved. Approval from the institutes was sought by directly communicating with respective institutes via email.

Informed consent: All of the study's participants were students studying for CS related bachelor's degrees. Only those who volunteered to take part in the study were invited. All students who agreed to take part in the study were provided with detailed letters (see Appendix VII Appendix VIII) explaining: the purpose and nature of the research; how the research data would be utilised; the researcher's obligations and responsibilities; and, what was expected of

the participants by agreeing to participate. All of the participants gave written consent using a standard format provided by Curtin University adapted to fit the context of the study.

Confidentiality and privacy: To maintain confidentiality and privacy, none of the participant's identification details were recorded in the data collection. The participants were only identified with a number during the data entry.

Risks/Benefit Analysis: Basically, there was no significant foreseen risk for any of the participants of the study as the researcher herself was the sole person handling the data. The researcher was not affiliated with any of the institutes and the only contact with the participants was during data collection. Therefore, none of the participants directly or indirectly would be affected by the responses provided in the test. Additionally, participants were informed not to use any form of identification in the answer scripts to ensure the students would not be identified later. There was no other foreseen risk of a third party misusing the data as the researcher herself handled the data in all the research stages from scoring to data entry and analysis.

Adequacy of Method: The researcher and the main supervisor had the main responsibility to ensure the methods employed in all stages of the study were adequate for the study undertaken. Furthermore, they were also responsible to ensure that the methods and analytical procedures employed to derive research outcomes were ethical, adequate, and defensible.

5.9. Summary

The chapter began with a brief overview of the two main paradigms associated with the research methodologies and the justification for choosing a positivistic quantitative approach of methods for the current investigation. Next, the chapter explicated the three main models associated with the instrument development investigation and the confluence of these models to achieve the main goals of the research. Then, the chapter outlined the research design and explained the three main phases of the investigation. The chapter concluded by expounding ethical issues that may have arisen during the course of the study. The next chapter presents the results for each of the three phases of the study.

Chapter 6 – Results

6.1. Introduction

This chapter provides the results of each of the three phases of the investigation into the development of a measure to gauge CS1 student competence. A phase-based approach is used to present the results. Phase one presents the results of the measurement development process. Phase two presents the results of the validity evidence of the investigation. Finally, Phase three results a correlational study on factors associated with CS1 student competence are presented.

6.2. Phase 1: CS1 Measure Development

This section presents the results of the instrument development investigation. The section is organised according to the four building blocks of Wilson's (2005) construct modelling approach. These are the construct map, item development, and outcome space and measurement model.

6.2.1. Construct map

The outcome of the first building block was a construct map grounded on the construct model of CS1 student competence proposed in Chapter 3. The construct map shows the increasing anchor points of sophistication in learning to computer program by novice programmers as they progress through the topics of the curriculum. The construct map was reviewed by the study's Expert Review Group (ERG): they found the core skills – tracing, reading and writing – as pertinent skills to learning to program and the continuum of proficiencies defined in the construct map as all appropriate.

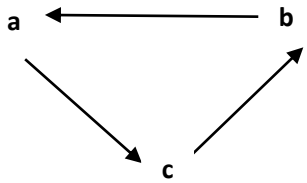
6.2.2. Item development

The second building block was concerned with item development to reveal the competencies or participant characteristics as they progress through each level defined in the construct map. The outcome of this building block was a test consisting of 20 questions. Four questions were designed for each construct or topic to capture the four levels defined in the construct map. This resulted in a 20-item test (4 questions for each topic x 5 topics). A sample item set developed for the loop structure is shown in Appendix IV. Two high school computer science students firstly reviewed the questions. Their purpose was to evaluate the clarity of the questions. These students highlighted some issues with item wordings that may have confounded understanding. For example, they suggested the word 'swap' as in the re-illustrated

version of Q1D (Figure 6.1) would be more appropriate than the word ‘shift’. After accommodating the suggested changes, the 20-item test was presented to two CS students studying in the second year of their computer science degree program to evaluate content representativeness. These students indicated that the topics constituting the test were studied in their CS1 course. The ERG review followed and this revealed a few further issues with the questions, especially the visual illustrations that may have precluded comprehension. For example, the writing task (Q1D) of the first topic (Basics) was re-phrased and re-illustrated as in Figure 6.1 based on ERG review feedback and student’s feedback.

Question 1(d) (Before)

There are three integer variables, a, b and c, which have been initialised. Write code to shift the values in these variables around so that a is given b’s original value, b is given c’s original value, and c is given a’s original value. The following diagram illustrates the direction of the shifts.



Question 1(d) (After)

There are three variables, a, b and c, which have been initialised to integer values. Write code to swap the values stored in these variables so that a is given the original value of b, b is given the original value of c, and c is given the original value of a. The following diagram illustrates the before and after effects of the swap

Figure 6.1. Question 1D before and after re-wording and re-illustration

The pilot test conducted with 10 students from the Asia Pacific University of Technology (APU), Malaysia results indicated some issues concerning the postulated ordering and wording of questions, though most were shown to follow the learning conjecture hypothesised in the construct map. The responses received from the pilot test were analysed to see whether the item responses for each topic formed a structure somewhat similar to the Guttman scale (Guttman, 1950). A Guttman scale is formed by a set of items if they can be ordered in a reproducible hierarchy similar to the item the hierarchy hypothesised in the construct map. Table 6.1 shows the answers for each part of question 1. For example, the difficulty order of question 1A and question 1B did not seem to follow this structure as

hypothesised. In other words, question 1A was found to be more difficult for the participants than question 1B. However, according to the total responses received for question 1C, it could be concluded that that 1C was more difficult than 1A and 1B. Exact same pattern of ordering was observed for Question 2A, 2B and 2C as well. Question 3B, 4A and 5A were assumed to be badly worded because many students did not attempt them unlike the tracing questions of other constructs. Few students wrote the reason for not attempting as instructed. Some students had difficulty in understanding the question while others said they had forgotten the concepts.

Table 6.1

Difficulty Rankings of Question 1 Based on Total Number of Correct Answers

	1A	1B	1C
S1	1	1	1
S6	1	1	0
S2	1	1	1
S3	1	1	1
S5	1	1	0
S7	1	1	0
S9	0	0	1
S4	0	1	0
S8	0	1	0
S10	0	0	0
Total	6	8	4

Based on these results, and after consultation with the ERG, question 1B and 2B were replaced by more challenging questions. To improve clarity, 3B and 5A were reworded, and the stem of each question was completely replaced by a similar, but more concise, question after consultation with the ERG. For example, the students found that question 3B (how many times will the while-loop execute?) confusing. However, the students did not specify why or which part was confusing. Therefore, the question was reworded and changed slightly after a one-to-one consultation with a participant of the pilot test.

Q3B *After the above code segment is executed, what is the value of z?*

The categories of writing questions, Question 2D and 4D, seemed to function well. For example, a good number of responses to Question 4D were received (1 student did not attempt, 1 student received score of 0, 3 students received score of 1, 2 students received score of 2, 2 student received score of 3 and 1 received score of 4) which was marked using the four point

scale shown in the Appendix II. The points or the scores received were consistent with student ability levels. This means the students who scored overall higher scores in the test also were the ones who scored in the higher levels of both Question 2D and 4D, and as expected, the students who scored overall low scores scored from the lower categories of 4D. Very few students attempted 1D, 3D and 5D. Due to the low response rate, the category functioning of these two questions was unable to be determined. However, the students indicated that the questions were not difficult; that they were unable to answer them because they had forgotten the concepts, as it had been a few months since they had completed CS1. The student feedback also indicated that they had studied the topics in CS1, thus, this helped to confirm the content representativeness of the measure. For example, one student who left Question 3D blank provided the following feedback:

"*All these topics I learned in CS1, but couldn't answer some because it has been like few months since we had done these stuff!"* (Student number 5)

6.2.3. Outcome space

The outcome space is related to how to categorise the variety of responses received for each item by the participants and assigned a numerical score to each of the response categories (Wilson, 2005). As explicated in the Methodology chapter (Chapter 5), two rubrics were developed to categorise and assign numerical values to the response categories. All A, B and C questions for each topic require fixed-answers. A correct answer suggests that the participant had reached the proficiency level defined in the construct map upon which the item was based. Categorising and scoring to these questions was found to be easy as the responses were fixed. However, scoring the writing questions with the 4 point scale was found to be difficult as none of the student's responses were shown to manifest the characteristics defined in category 4 or at the relational extended level (highest level) of the SOLO taxonomy. Consequently, the highest response category achieved by the participants was the relational level of the SOLO taxonomy. Figure 6.2 shows an example of what an expected extended abstract level solution for Question 2D might look like. However, the students' answers matched to the definition of the relational level because they reached answers with many unnecessary duplications of If/Else code as shown in Figure 6.3.

```
if (firstTime==0)
     ticketAmount = 300;
else
{
     if(actualSpeed >=61)
     ticketAmount = 150;
     else if(actualSpeed>=51)
     ticketAmount = 75;
     else if(actualSpeed>=41)
     ticketAmount = 50;
}
```

Figure 6.2. An expected level solution for Q2D at the 4th level (Extended abstract)

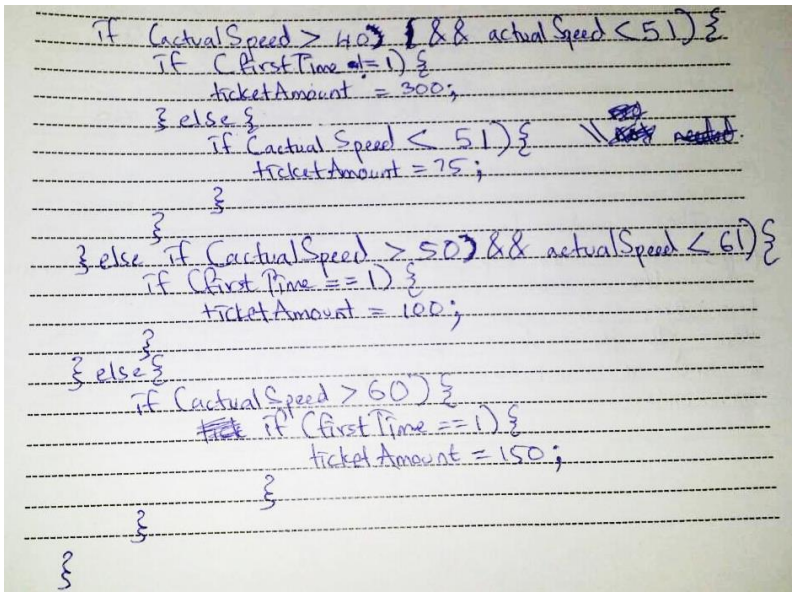


Figure 6.3. A sample student solution for Q2D scored at 3rd Level (Relational)

6.2.4. Rasch analysis of the data (Measurement model)

The aim of the final building block is the measurement model, and its objective is to relate the scored data back to the construct map. The construct modelling approach typically employs the Rasch Measurement Model to link these two elements by generating a hypothetical unidimensional line (a logit ruler) along which items and persons of the test are located according to their difficulty and ability measures. The following presents results of the Rasch analysis of response data using an unconstrained Rasch Unidimensional Measurement Model (RUMM2030) computer program (Andrich, Sherridan, & Luo, 2010).

A summary of the Rasch analysis of the data is shown in Table 6.2. Initial inspection of the 20-item CS1 measure revealed a non-significant item-trait interaction with a total Chi-square of (X^2 = 49.32, df = 40, p = 0.15), suggesting the response patterns fitted the Rasch model's expectations. The mean person location was -0.35, suggesting students overall found the test difficult, with a standard deviation of 1.69 due to the larger variance of student transformed scores. A mean person log residual test of fit of -0.31 (SD = 0.91) and item fit residual test of fit of -0.25 (SD = 0.53) indicated overall that both person and item fit residual were reasonably close to the ideal values of 0±1. The Person Separation index (PSI) is an estimate of the spread or separation of persons on the measured variable (Bond & Fox, 2015), which ranges between 0 - 1. Values for PSI of 0.8 are acceptable (Wright & Masters, 1982). The person separation index (PSI) of 0.87, suggests that the items can separate the participants into at least three statistically distinct competency groups (Linacre, 2014).

However, two items (3D and 5D) manifested reversed thresholds. Reversed thresholds in a Rasch analysis raises the issue of the scoring of the categories with successive integers. A threshold, defined by Bond and Fox (2015), is "the level at which the likelihood of failure to agree with or endorse a given response category (below the threshold) turns to the likelihood of agreeing with or endorsing the category above the threshold" (p. 314). If two successive thresholds are reversed, for example, the threshold between x-1 and x is greater than the threshold between x and x+1, it means that the person on the boundary of the former has a greater ability than the person on the boundary of the latter. This issue was dealt with, before proceeding with further analysis. The details of the procedure applied to resolve this issue are presented in the following section. The overall fit statistics of the initial analysis of data with RUMM2030 is shown in Table 6.2.

Table 6.2

Summary of Overall Fit between the Data and the Rasch Model

Analysis	Mean Item Location (SD)	Item fit residual mean (SD)	Mean Person Location(SD)	Person fit residual mean (SD)	X^2	df	P	PSI
Before Rescoring	0.0 (1.77)	-0.13 (0.91)	-0.35 (1.69)	-0.25 (0.53)	49.32	40	0.15	0.87
After Rescoring (3D,5D)	0.0 (1.84)	-0.11 (0.96)	-0.377 (1.84)	-0.32 (0.63)	39.95	40	0.47	0.87
Ideal Values	0.0 (1.0)	0.0 (1.0)	0.00 (1.0)	0.0 (1.0)			>0.05	

6.2.4.1. Disordered Thresholds and Re-scoring

Questions 3D and 5D were found to have disordered thresholds and did not accord to the levels defined in the outcome space, as defined (Appendix II). Table 6.3 shows the frequencies received for each category of the polytomous items. Question 2D and 4D received more than 10 responses for each category suggesting there was enough information to calculate the centralised thresholds. However, categories 2, 3 and 4 received unexpectedly low responses, particularly, category 3 in each case. The centralised thresholds for these items and the Category Probability Curves (CPC) (Figure 6.4) evidenced that the thresholds were reversed, suggesting the second category did not represent more of the trait than the first category (Linacre, 1999). For example, the threshold of the last category of Questions 3D (-1.27) and 5D (-1.08) is less than the category 2 thresholds of these questions. This suggests the thresholds are not in their natural order.

Table 6.3

Category Response Frequencies

Question	Cat 1 (Score 0)	Cat 2 (Score 1)	Cat 3 (Score 2)	Cat 4 (Score 3)
2D	16	21	25	15
3D	40	10	2	9
4D	19	15	12	11
5D	39	8	3	6

This condition can also be visually examined by analysing the pattern of the category probability curves generated for these questions by RUMM2030. Figure 6.4 displays the category probability curves of Question 5D before the condition was treated. The graph shows participant locations (logits) are plotted on the horizontal axis ranging left to right according to their ability locations on the CS1 measure. Similarly, the vertical axis plots the probability of observing each ordered category. Theoretically, each response category should peak at some point on the graph, that is, at some point, each category must become the most likely scored category by the participants as shown in Figure 6.5. For example, looking at the category functioning of Question 5D (Figure 6.5), the likelihood of a person scoring 0 (the blue curve) decreases as the participant's ability increases and scoring 1 (the red curve) becomes the most likely scored category. Similarly, the likelihood of a person scoring 1 decreases as the participant's ability increases and scoring 2 becomes the most likely scored category. When the categories function properly, this pattern repeats.

However, when the disordering of thresholds occurs, the pattern described before is disrupted as displayed in Figure 6.4. For example, response categories 1 and 2 (the red and the blue curve) did not peak and never become the most likely scored categories of the participants, suggesting the originally coded categories (0, 1, 2, 3) did not work as expected. The exact same pattern of category disordering of thresholds was also observed in Question 3D. Similarly, the Table 6.4 also evidences the threshold estimates are reversed. In other words, as assumed, the threshold estimates were not in increasing order (threshold 3 in each of these items is less than the adjacent threshold). Consequently, these items were dealt with by collapsing the middle two categories. In other words, the two adjacent categories (category 2 and 3, both of which has less than 10 responses) were re-scored as 1 and what was previously 3 was re-scored as 2 resulting in a 3 point scale (0, 1 and 2) for the two questions (Linacre, 1999). Collapsing category 2 and 3 seemed to be more appropriate than collapsing category 3 and 4 because it results in a natural decrease order of frequencies suggesting increased difficulty.

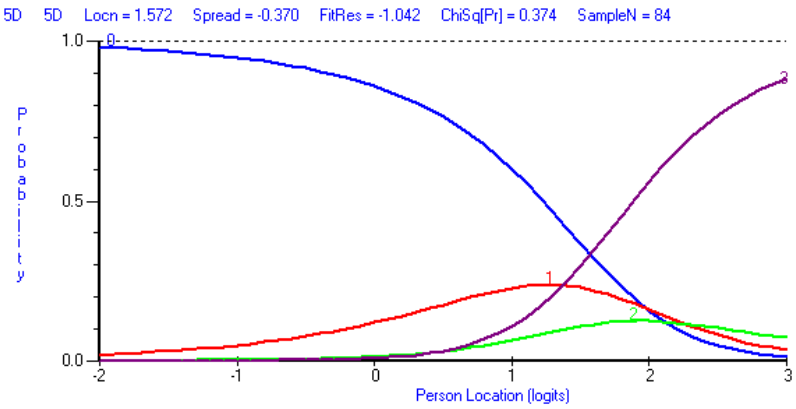


Figure 6.4. Category probability curves for Q5D before collapsing

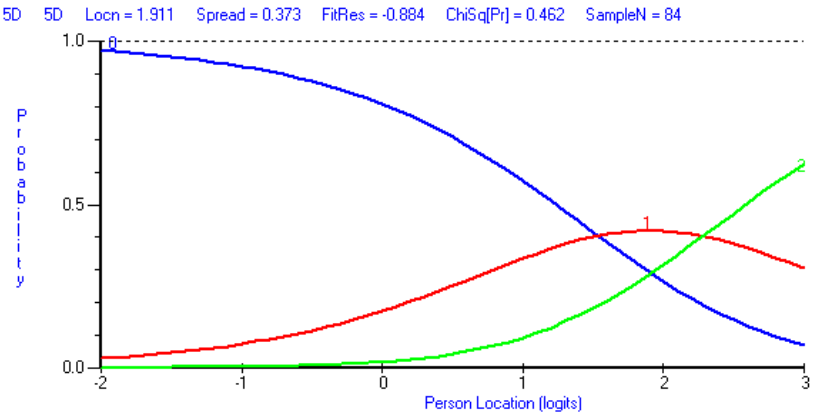


Figure 6.5. Category probability curves for Q5D after collapsing

Table 6.4

Items with Disordered Thresholds

Item Code	Location	Threshold 1	Threshold 2	Threshold 3
3D	1.27	0.26	1.01	-1.27
5D	1.57	0.40	0.68	-1.08

Figure 6.5 shows the result of regenerating the probability curves for Question 5D after collapsing the categories. Both items (3D and 5D) illustrated that three response categories fit better to these questions, which also results in improving fit statistics from -1.042 to 0.884. Similarly, the chi-squared probability (at the top of Figure 6.7) also suggested improved fit. Question 1D was found to be difficult to score on a polytomous scale like other counterpart writing questions, which successively re-scored dichotomously at the scoring stage. Other counterparts writing questions were left unchanged as a 3 point scale did not work well for them. Refitting the data to RUMM2030 with these changes indicated that the categories for all items were ordered correctly.

6.2.4.2. Overall fit to the Rasch model after re-scoring

The final statistics produced by RUMM2030 after re-scoring revealed none of the items or the persons had a significant misfit. The overall fit residual for the rescored measure is shown in Table 6.2. The overall performance of the measure marginally improved with the re-scoring of the items. The reliability was excellent with a PSI of 0.87, indicating good internal consistency (Tennant & Pallant, 2006). The high PSI also suggests that the measure has a good spread of items and these are sensitive enough to discriminate the sample into at least three programming competence groups (Linacre, 2014).

6.2.4.3. Individual item and person fit

Item fit can be analysed both by graphical displays generated by RUMM2030 as well as examining the fit residual statistics provided by RUMM2030. Fit residuals are the difference between the raw score and the score predicted by RUMM2030. All the item fit residuals were within the range of ±2.5, with the majority well below the range as shown in Table 6.5. An item with a good fit was attributed to the class intervals of participants represented by the black dot approaching the ogive (theoretical curve) predicted by the model. Q4D shown in Figure 6.6 is an item with a good fit, which shows the class intervals are almost touching the ogive, more specifically, the actual score and score predicted by the model were almost the same with a very small fit residual (-0.466).

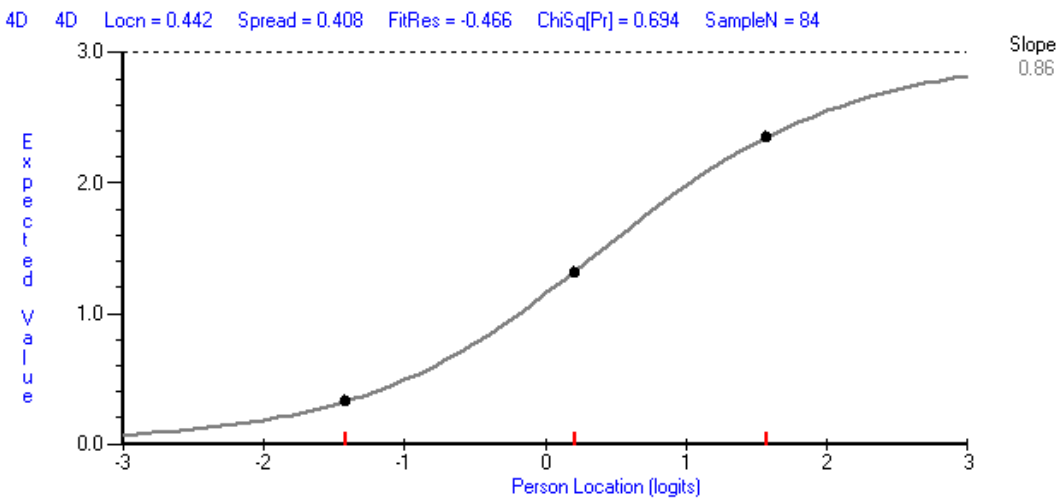


Figure 6.6. ICC for 4D showing a good fit to the model

Table 6.5

Item Fit Statistics after Rescoring (N=20)

Seq	Item	Type	Location	SE	FitResid	DF	ChiSq	DF	Prob
1	1A	Poly	-4.59	0.56	-0.48	77.69	0.47	2	0.79
2	1B	Poly	-3.18	0.40	-0.86	77.69	1.39	2	0.50
3	1C	Poly	1.45	0.30	0.08	71.21	0.62	2	0.73
4	1D	Poly	0.98	0.14	0.32	73.06	4.67	2	0.10
5	2A	Poly	-1.94	0.32	-0.57	76.76	3.13	2	0.21
6	2B	Poly	-0.95	0.28	1.62	77.69	7.95	2	0.02
7	2C	Poly	-0.22	0.28	-0.08	73.06	0.01	2	1.00
8	2D	Poly	-0.08	0.16	0.12	71.21	0.65	2	0.72
9	3A	Poly	-0.94	0.28	2.44	77.69	5.04	2	0.08
10	3B	Poly	1.82	0.32	-0.70	75.84	2.81	2	0.25
11	3C	Poly	1.97	0.36	-0.64	64.74	3.16	2	0.21
12	3D	Poly	1.27	0.17	-0.86	55.49	1.71	2	0.42
13	4A	Poly	-0.87	0.30	0.02	65.66	3.12	2	0.21
14	4B	Poly	0.88	0.29	-0.61	67.51	0.85	2	0.65
15	4C	Poly	0.91	0.32	-0.83	56.41	3.62	2	0.16
16	4D	Poly	0.44	0.17	-0.47	52.72	0.73	2	0.69
17	5A	Poly	-1.33	0.32	1.15	65.66	0.53	2	0.77
18	5B	Poly	1.11	0.31	-0.75	57.34	3.47	2	0.18
19	5C	Poly	1.69	0.35	-0.47	51.79	3.42	2	0.18
20	5D	Poly	1.57	0.19	-1.04	51.79	1.97	2	0.37

Similar to the item fit residual, none of the persons had the fit residuals outside the range of ±2.5, where values close to 0 indicated a good fit (Sjaastad, 2014). The fit residuals of persons ranged between -1.970 to 0.936.

6.2.4.4. Differential Item Functioning (DIF)

DIF investigates the items in a test, one at a time, for signs of interactions with sample characteristics such as gender (Badia, Prieto, & Linacre, 2002). To demonstrate unidimensionality, and the variable to be linear, the items of the scale have to work invariantly across individuals and groups (Hagquist & Andrich, 2017). In a DIF analysis, difficulty estimates of the items obtained for one subgroup within the sample are compared with those for another subgroup using analysis of variance (ANOVA). The results of the DIF analysis performed is presented in Table 6.6. It summarises the two types of DIF: (1) Uniform DIF occurs when the locations of the items are different but the slopes of the observed points are parallel; and, (2) Non-uniform DIF occurs when the locations are the same but the slopes are different (Andrich, Sheridan, & Luo, 2011). As presented in Table 6.6, based on F-ratios (number of class intervals = 3) shows none of the items have significant DIF. DIF was examined with all the factors shown in Table 5.2, with gender being excluded due to the few number of participants. None of the items showed DIF with these different demographic groups. DIF was also examined by graphical displays provided by RUMM2030 for each item. Figure 6.7 shows an ICC for Question 3D for those who had taken a High school CS course and those who had not. The blue curve denotes those who had taken a course and the red represents those who had not. It can be seen that, given a particular ability level, the probability of being successful on this item is not different for the two groups demonstrated by no inconsistent shift in item difficulty across the ability continuum. This suggests all the items work invariantly along the measurement continuum.

Table 6.6

Uniform and Non-uniform DIF Statistics (N=20)

Uniform DIF by Institute					Non-Uniform DIF by Institute				
Item	MS	F	DF	Prob	Item	MS	F	DF	Prob
1A	0.22	0.49	2	0.61	1A	0.14	0.32	4	0.86
1B	0.75	1.65	2	0.2	1B	0.42	0.93	4	0.45
1C	2.26	2.27	2	0.11	1C	0.42	0.42	4	0.79
1D	3.22	4.04	2	0.02	1D	1.86	2.33	4	0.06
2A	0.77	1.23	2	0.3	2A	0.62	0.99	4	0.42
2B	1.12	0.56	2	0.57	2B	1.00	0.50	4	0.73

2C	0.15	0.15	2	0.87	2C	0.79	0.76	4	0.56
2D	2.39	2.46	2	0.09	2D	0.99	1.02	4	0.40
3A	4.43	2.92	2	0.06	3A	3.94	2.59	4	0.04
3B	1.50	3.17	2	0.05	3B	0.28	0.59	4	0.67
3C	1.36	2.92	2	0.06	3C	0.28	0.60	4	0.67
3D	1.66	3.08	2	0.05	3D	0.05	0.09	4	0.99
4A	0.40	0.30	2	0.75	4A	0.33	0.25	4	0.91
4B	0.45	0.62	2	0.54	4B	0.57	0.77	4	0.55
4C	0.23	0.39	2	0.68	4C	0.56	0.96	4	0.44
4D	0.44	0.52	2	0.6	4D	0.40	0.46	4	0.76
5A	2.54	1.55	2	0.22	5A	3.10	1.89	4	0.12
5B	0.01	0.02	2	0.98	5B	0.47	0.85	4	0.50
5C	0.58	0.99	2	0.38	5C	0.82	1.39	4	0.25
5D	1.22	2.72	2	0.08	5D	0.71	1.58	4	0.20

Uniform DIF by Stream					Non-Uniform DIF by Stream				
Item	MS	F	DF	Prob	Item	MS	F	DF	Prob
1A	0.53	1.24	2	0.30	1A	0.27	0.63	2	0.54
1B	0.42	0.89	2	0.42	1B	0.34	0.73	2	0.49
1C	0.02	0.02	2	0.98	1C	0.45	0.43	2	0.65
1D	0.84	0.54	2	0.59	1D	0.70	0.45	2	0.64
2A	0.56	0.82	2	0.45	2A	2.23	3.27	2	0.04
2B	1.85	1.33	2	0.27	2B	1.31	0.94	2	0.39
2C	0.09	0.10	2	0.91	2C	1.15	1.21	2	0.30
2D	1.77	1.84	2	0.17	2D	0.51	0.53	2	0.59
3A	9.46	5.61	2	0.01	3A	-1.79	-1.06	2	1.00
3B	0.01	0.02	2	0.98	3B	0.36	0.65	2	0.52
3C	0.25	0.45	1	0.50	3C	0.08	0.14	2	0.87
3D	0.00	0.00	1	0.95	3D	0.04	0.09	2	0.92
4A	0.57	0.60	2	0.55	4A	-0.02	-0.02	2	1.00
4B	2.49	3.73	2	0.03	4B	0.57	0.85	2	0.43
4C	0.03	0.05	2	0.95	4C	0.71	1.20	2	0.31
4D	0.07	0.08	1	0.78	4D	0.04	0.04	2	0.96
5A	1.07	0.73	2	0.48	5A	1.41	0.96	2	0.39
5B	0.00	0.01	2	0.99	5B	0.72	1.17	2	0.32
5C	0.94	1.48	1	0.23	5C	0.29	0.46	2	0.63
5D	0.09	0.27	1	0.60	5D	1.05	3.19	2	0.05

Uniform DIF by Mathematics Background					Non-Uniform DIF Mathematics Background				
Item	MS	F	DF	Prob	Item	MS	F	DF	Prob
1A	0.45	1.05	1	0.31	1A	0.48	1.13	2	0.33
1B	0.04	0.08	1	0.78	1B	0.65	1.39	2	0.26
1C	0.23	0.23	1	0.63	1C	2.69	2.78	2	0.07
1D	1.06	0.67	1	0.42	1D	0.04	0.03	2	0.97
2A	3.01	4.28	1	0.04	2A	0.14	0.20	2	0.82

2B	2.82	2.05	1	0.16	2B	1.52	1.11	2	0.34
2C	0.46	0.48	1	0.49	2C	0.40	0.42	2	0.66
2D	0.53	0.54	1	0.46	2D	0.94	0.96	2	0.39
3A	1.48	0.85	1	0.36	3A	3.91	2.24	2	0.11
3B	3.10	6.26	1	0.01	3B	0.53	1.07	2	0.35
3C	0.72	1.42	1	0.24	3C	1.16	2.30	2	0.11
3D	0.00	0.00	1	0.95	3D	0.48	1.00	2	0.37
4A	1.62	1.74	1	0.19	4A	0.16	0.17	2	0.84
4B	4.22	6.47	1	0.01	4B	1.18	1.80	2	0.17
4C	0.22	0.37	1	0.54	4C	0.49	0.84	2	0.44
4D	0.46	0.53	1	0.47	4D	0.10	0.11	2	0.89
5A	4.52	3.29	1	0.07	5A	2.46	1.80	2	0.17
5B	0.22	0.36	1	0.55	5B	0.32	0.52	2	0.60
5C	1.29	2.12	1	0.15	5C	0.77	1.27	2	0.29
5D	0.09	0.26	1	0.61	5D	0.11	0.31	2	0.73

Uniform DIF by High school CS					Non-Uniform DIF by High School CS				
Item	MS	F	DF	Prob	Item	MS	F	DF	Prob
1A	0.06	0.14	1	0.71	1A	0.06	0.14	1	0.71
1B	0.08	0.16	1	0.69	1B	0.08	0.16	1	0.69
1C	0.45	0.45	1	0.50	1C	0.45	0.45	1	0.50
1D	0.01	0.01	1	0.93	1D	0.01	0.01	1	0.93
2A	0.41	0.59	1	0.44	2A	0.41	0.59	1	0.44
2B	0.26	0.18	1	0.67	2B	0.26	0.18	1	0.67
2C	0.49	0.54	1	0.46	2C	0.49	0.54	1	0.46
2D	4.30	4.66	1	0.03	2D	4.30	4.66	1	0.03
3A	0.15	0.08	1	0.77	3A	0.15	0.08	1	0.77
3B	0.24	0.45	1	0.51	3B	0.24	0.45	1	0.51
3C	0.00	0.00	1	0.98	3C	0.00	0.00	1	0.98
3D	0.03	0.07	1	0.79	3D	0.03	0.07	1	0.79
4A	0.21	0.23	1	0.64	4A	0.21	0.23	1	0.64
4B	0.09	0.13	1	0.72	4B	0.09	0.13	1	0.72
4C	0.62	1.09	1	0.30	4C	0.62	1.09	1	0.30
4D	0.11	0.13	1	0.72	4D	0.11	0.13	1	0.72
5A	0.38	0.27	1	0.61	5A	0.38	0.27	1	0.61
5B	0.13	0.22	1	0.64	5B	0.13	0.22	1	0.64
5C	0.50	0.80	1	0.38	5C	0.50	0.80	1	0.38
5D	1.30	3.82	1	0.06	5D	1.30	3.82	1	0.06

Probability values were based on F-ratios (number of class intervals = 3). Significant deviations are checked against $p < 0.00083$ (Bonferroni adjusted)

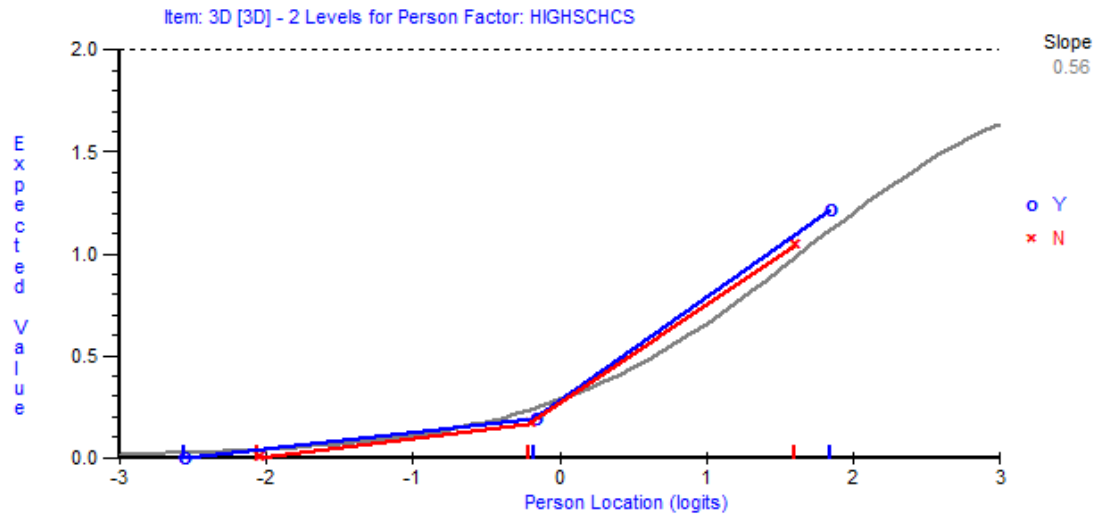


Figure 6.7. ICC for Question 3D showing no significant DIF

6.2.4.5. Local dependency and unidimensionality

To investigate the presence of possible local dependency, the residual items correlation matrix was analysed to identify unusually higher correlations. Table 6.7 shows the item residual correlation matrix of the 20-item measure. The highest correlation found was between items 5B and 5C (0.5). However, their correlation was not 0.30 above the average item correlation (0.315). Furthermore, Linacre (2014) reports that local dependence would be highly locally dependent items (Corr. > +0.7). The two suspected locally dependent items share only (0.5 x 0.5) 25% of the variance in their residuals in common, whereas, 75% of each of their residual variances differ.

Table 6.7

Residual Correlation Matrix of all Items after Taking the Rasch factor

Item	1A	1B	1C	1D	2A	2B	2C	2D	3A	3B	3C	3D	4A	4B	4C	4D	5A	5B	5C	5D
1A	1.0																			
1B	0.0	1.0																		
1C	0.0	0.1	1.0																	
1D	0.0	0.0	-0.2	1.0																
2A	-0.1	0.0	-0.3	-0.2	1.0															
2B	-0.1	0.1	-0.2	-0.1	0.1	1.0														
2C	0.1	-0.1	0.3	0.1	-0.1	-0.1	1.0													
2D	0.0	-0.2	0.0	0.0	0.0	0.0	0.3	1.0												
3A	-0.2	-0.2	0.0	-0.2	0.0	0.0	-0.1	-0.1	1.0											
3B	0.0	0.0	-0.1	-0.1	0.1	-0.2	-0.2	0.0	0.2	1.0										
3C	0.0	0.0	-0.2	-0.1	0.1	-0.2	0.0	-0.1	0.1	0.3	1.0									
3D	0.0	0.0	-0.2	0.0	0.1	0.0	-0.2	-0.2	0.0	0.0	-0.2	1.0								
4A	0.1	-0.1	-0.1	0.0	-0.1	0.0	-0.1	-0.1	-0.1	-0.2	-0.3	0.0	1.0							
4B	0.0	0.1	-0.1	-0.2	0.0	-0.1	-0.3	-0.1	0.0	-0.1	0.0	0.0	-0.2	1.0						
4C	0.0	0.1	0.1	0.1	-0.3	-0.1	-0.1	-0.1	-0.2	-0.2	-0.2	-0.1	-0.1	0.3	1.0					
4D	-0.2	0.2	0.2	0.0	0.0	0.0	-0.2	-0.4	-0.1	-0.2	-0.1	-0.1	-0.1	0.0	0.1	1.0				
5A	0.0	-0.1	-0.2	0.0	-0.1	-0.1	-0.1	-0.2	-0.2	0.0	-0.1	-0.1	0.0	-0.1	0.1	-0.1	1.0			
5B	0.0	0.1	-0.1	-0.2	-0.1	0.0	-0.1	-0.1	-0.1	0.1	0.1	-0.1	-0.1	0.0	0.1	-0.1	-0.2	1.0		
5C	0.0	0.0	0.0	-0.2	-0.2	-0.1	0.0	-0.3	0.0	-0.2	-0.1	-0.1	0.1	0.0	0.2	-0.2	0.1	0.5	1.0	
5D	0.0	0.0	-0.3	0.2	0.1	-0.4	-0.3	-0.3	-0.2	0.0	0.2	0.1	0.2	0.3	-0.3	-0.1	0.2	-0.1	0.1	1.0

Positive Residual Correlations > 0.3 are highlighted in green

Fit of the data to the Rasch model requires that the entire correlation between the items have to be captured by the latent trait. Correlation between any pair of items that are not accounted for by the Rasch factor is a symptom of either local dependency or multidimensionality, both of which are concerns that may violate Rasch model requirements (Hagquist et al., 2009). The PCA of Rasch residual (Chang, 1996; Linacre, 1998; Wright, 1996a) revealed that the strength of the first two residual contrasts as being little higher than 2 eigenvalues (See Table 6.8), with the strength of about 2 items (2.28 of 20 items). Therefore, a strict dimensionality testing was performed by taking two sets of items based on the correlations (positive and negative) between the items on the first residual factor as shown in Table 6.9 (Smith, 2002; Tennant & Conaghan, 2007). The items at the top were the highest three positively loaded items, whereas, the items at the bottom were the three highest negatively loaded items on the first residual factor. The person estimates of these two sets compared through a series of t-tests revealed that the proportion of person locations contrasted between the two item set represented in Figure 6.8 was only 3.5% significant. Since this was less than

5% significant, strict unidimensionality was supported (Smith, 2002; Tennant & Conaghan, 2007; Tennant & Pallant, 2006).

Table 6.8

Principal Components Summary

PC	Eigen	Percent	CPercent	StdErr
PC001	2.28	0.11	0.11	0.31
PC002	2.05	0.10	0.22	0.28
PC003	1.79	0.09	0.31	0.25
PC004	1.70	0.09	0.39	0.23
PC005	1.54	0.08	0.47	0.21
PC006	1.33	0.07	0.53	0.18
PC007	1.23	0.06	0.60	0.16
PC008	1.19	0.06	0.66	0.16
PC009	1.05	0.05	0.71	0.14
PC010	1.04	0.05	0.76	0.13
PC011	0.89	0.04	0.80	0.12
PC012	0.83	0.04	0.85	0.11
PC013	0.76	0.04	0.88	0.10
PC014	0.63	0.03	0.91	0.09
PC015	0.56	0.03	0.94	0.08
PC016	0.51	0.03	0.97	0.07
PC017	0.36	0.02	0.99	0.07
PC018	0.25	0.01	1.00	0.07
PC019	0.09	0.00	1.00	0.06
PC020	-0.05	0.00	1.00	0.05

Table 6.9

Principal Component Analysis of the Residuals Showing First Component Loadings

Item	Loading
5D	0.77
4B	0.39
3D	0.30
1C	-0.54
2D	-0.57
2C	-0.67

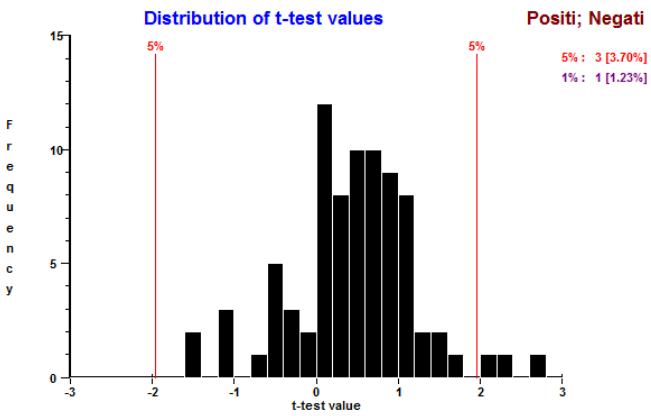


Figure 6.8. Summary of independent t-tests

6.2.4.6. Item-person distribution (targeting) and reliability

Figure 6.9 displays the distributions of items and persons on the unidimensional scale created. The persons taking the test are arrayed on the left side of the graph with each denoted by “x”; on the right side of the unidimensional line are the item thresholds. The average mean person location (-0.38 logits) demonstrates that on the whole, the measure was well targeted to the sample, although students on average found the test slightly difficult. Hence, more students were located at the lower level of the construct than the average of the items (set to 0 logits). The thresholds positioned at the bottom left (tracing questions; example 1A.1) were the easiest to score, whereas, the thresholds positioned at the top left were the most difficult to score (highest level of writing – 5D.2). Unlike as expected in the construct model, there was no clear separation between the highest levels of writing (highest levels are 1D.1, 2D.3, 3D.2, 4D.3 and 5D.2) and the code comprehensions levels (all [C]). The spread of the items was reasonable covering a sufficient range of ±4, which was deemed sufficient (Sampaio et al., 2012). The item-person threshold distribution map (Figure 6.10) shows there is no major floor or ceiling effects suggesting an adequate coverage of the construct. The PSI, which indicates a domain's ability to discriminate between the levels of an underlying trait, of the measure was 0.87. This suggests that the sample of students who completed the CS1 measure can be differentiated into at least three levels of competence; therefore the measure is suitable for use within the classroom for individual assessments and research (Fisher, 1992; Pallant & Tennant, 2007).

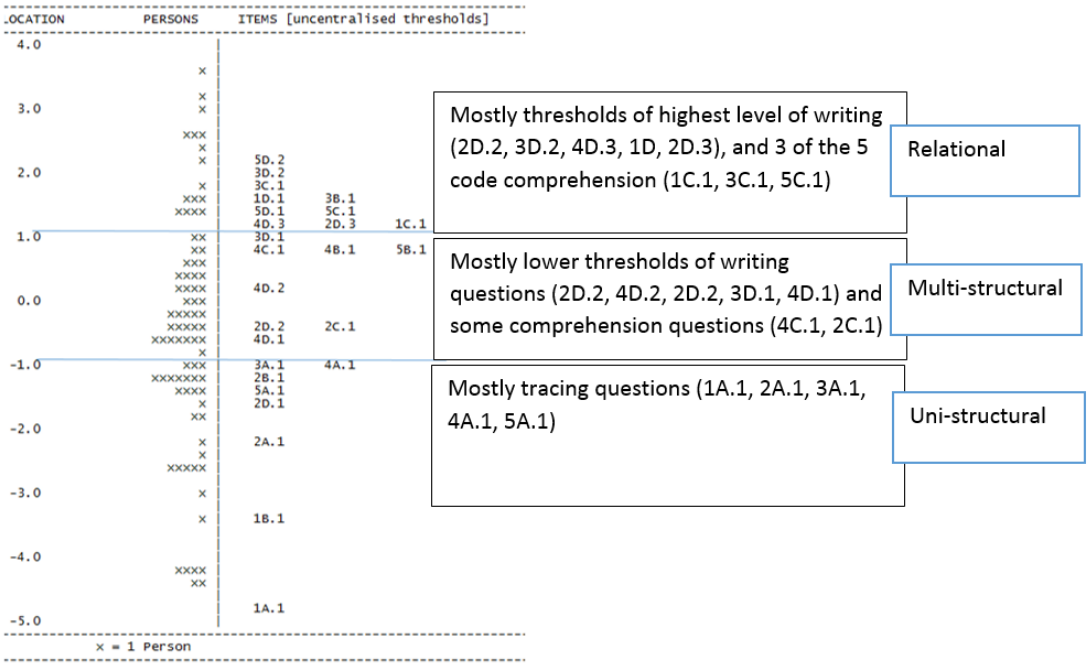


Figure 6.9. Item-person map

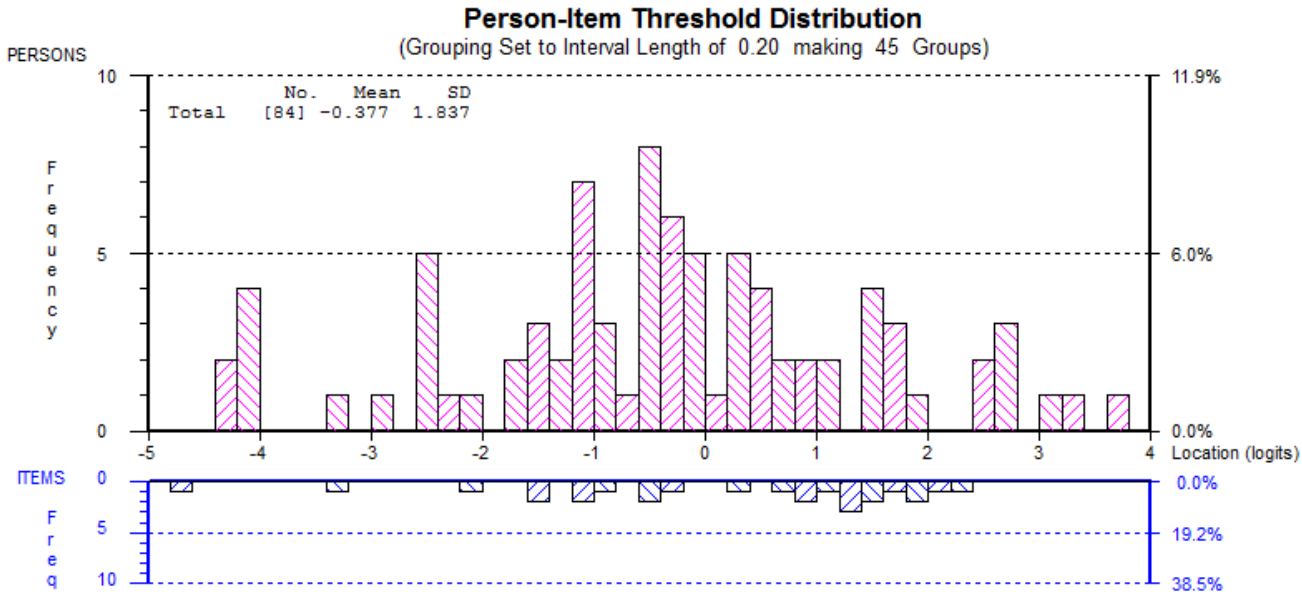


Figure 6.10. Item-person threshold distribution map for the CS1 measure

6.3. Phase 2: Validity Evidence

This section provides the *post hoc* evaluation of the validity of the research intentions, the investigative process, and outcomes of the study into the development of a CS1 measure. The outcomes from the development process, specifically the graphical displays and inferential statistics provided by a Rasch analysis of the data, were selectively taken to exemplify validity evidence. The validity argument, as discussed below, is organised according to seven aspects.

6.3.1. Evidence of the content aspect

Content validity concerns the representativeness of items constituting the construct which it espouses to measure (Kline, 1998). It relates to the relevance and representativeness of the content upon which the items are developed and the technical quality of the items (Wolfe & Smith, 2007a, 2007b). This can be justified using qualitative and quantitative analysis.

a) One way to provide content evidence is having a clear statement of purpose, such as specifying the research intentions and research questions (Wolfe & Smith, 2007a). For example, the purpose has been made clear with the objective measurement of construct–CS1 student competence based on the fundamental concepts of introductory computer programming. The rationale was to offer a tested measure to gauge student competence for a variety of research purposes because currently unreliable sources are employed to measure student competence for pedagogical purposes. Additionally, the intentions were

made explicit by the research question: *Can a measure of CS1 student competency be developed*?

Similarly, types of inferences and potential constraints and limitations compliment the explicating purpose (Wolfe & Smith, 2007a). The domain of inference is criterion-based, as it is concerned with the competencies of CS1 students after the completion of a typical CS1 course. The inference to be drawn from the study was the programming competence of CS1 students on five fundamental CS1 concepts that constitute a typical CS1 course. Additionally, the significance of some personal factors to CS1 student competence would also be revealed. The main constraint was the small sample size and limited range of topics representing the theoretical construct of the measure. This was because only those topics that were considered common across the different CS1 instructional paradigms had been chosen.

b) Instrument specification constitutes a description of the construct, the construct model and the construct map (Wolfe & Smith, 2007a). The construct of student competency in CS1 was deemed to be unidimensional and conceptualised as comprising of five constructs or topics. A construct model and construct-map were developed explicating the internal structure of the latent construct. Similarly, the item format, scoring, and scaling model were elucidated in detail. Throughout the process, ERG advice was incorporated from the construct model development to item design and construction.

c) Similarly, item development requires decisions about the type of scale, the number of response categories and the labeling of categories. Rasch unconstrained partial-credit scoring was applied to accommodate a variety of question types, which were elaborated in Chapter 5 (Section 5.5.4).

The professional judgment of ERG ensures the test items or tasks are relevant and representative of the construct domain (Cavanagh, 2009). The ERG reviews were conducted in the developmental stages for the relevance of the construct model and of the items and scoring model as per the construct model.

Item technical quality involves aspects such as unambiguous phrasing, accurate answer keys and providing suitable reading levels for the target population (Messick, 1996). A small pilot study conducted with 10 students resulted in subsequent item refinement. The technical quality of the items could be examined based on the response data (Wolfe & Smith, 2007a), and the empirical data concerning the difficulty and item-discrimination power (Messick, 1989). The quality of items was empirically tested by examining the item fit residuals produced

by the Rasch analysis of the response data, which showed the residuals were well fitting as revealed in Table 6.5 (Section 6.2.4.3), and the PSI was 0.87, suggesting a good reliability of the measure which can separate the sample into at least three meaningful competence groups. A PSI of 0.87 also supports the use of the CS1 measure for use within classroom for individual use and research (Fisher, 1992; Pallant & Tennant, 2007). Additionally, the Item person map, which also could be used as a source of evidence for the technical quality of items (Lim, Rodger, & Brown, 2009), revealed that the items covered a comprehensive range of the construct under investigation, suggesting the construct was adequately covered.

6.3.2. Evidence of the substantive aspect

The substantive aspect explains the theoretical rationales for observed consistencies in test responses with respect to a theory or predicted model (Wolfe & Smith, 2007a). For example, the theoretical model informing the study (Figure 3.3, Section 3.2.5) suggests that a typical CS1 measure encompasses a number of constructs or topics with four fundamental skills forming competency acquisition hierarchy. In this hierarchy the knowledge of programming constructs form the bottom of the hierarchy, code tracing and explaining form the intermediate skills, and code writing skills are higher order. The questions were developed and the location of the majority of the items conformed to this *priori*. Additionally, ANOVA results showed the higher levels of mathematics ability (Bergin & Reilly, 2006; Evans & Simkin, 1989; Jerkins et al., 2013; Leeper & Silver, 1982) and prior programming experience (Strnad et al., 2009; Wiedenbeck, 2005) were associated with CS1 student competence, which was consistent with past studies. The students who had done serious computer programming for at least six months before enrolling into CS1 course performed better, showing a smaller spread (Figure 6.13) with a higher mean score than those did not have experience. Furthermore, the One-way ANOVA analysis to examine the performance of the two groups showed the two groups were statistically significant at $p<0.05$ level [($F(1, 83) = 4.70$, $p = 0.03$)]. Similarly those who gained entry into the CS program after completing year 12 mathematics performed better than those who gained entry after completing year 10 mathematics [$F(1, 82) = 5.11$, $p = 0.03$, $\eta^2 = 0.54$].

Table 6.10

Expected Vs. Observed Difficulty of the Items

No	Constructs	Item (expected)	Actual Location
1	Fundamentals (variables, assignment, etc.)	1A	-4.59
		1B	-3.18
		1C	1.45
		1D	0.98
2	Selection Statement (if/else) (subsumes operators)	2A	-1.94
		2B	-0.95
		2C	-0.22
		2D	-0.08
3	Loops (subsumes operators)	3A	-0.94
		3B	1.82
		3C	1.97
		3D	1.27
4	Methods (includes functions, parameters, procedures, and subroutines)	4A	-0.87
		4B	0.88
		4C	0.91
		4D	0.44
5	1 Dimensional Arrays	5A	-1.33
		5B	1.11
		5C	1.69
		5D	1.57

An important priori of this study was that the questions measuring each concept retain an increasing order of difficulty based on the SOLO taxonomy. Therefore, it was expected that most of the A and B questions would sustain the lowest degree of difficulty, stretching the lower levels of the continuum, with all C and D questions spanning the top of the continuum. The majority of the questions maintained this hierarchy at a reasonable level both at the construct level as well as overall as revealed in Table 6.10 and Table 6.11. For example, all parts of question 2 (2A, 2B, 2C, 2C and 2D) show the locations of the items in increasing order of difficulty (-1.94 < -0.95 <- 0.22 < -0.08). Similarly, the mean of all A questions (-5.0 logits) are lower than the mean of B questions (-0.32 logits), although C questions have a slightly higher mean (5.8 logits) than the D questions (4.18 logits). This was expected, given that the writing questions could not be scored at the highest level as characterised in the rubric. This has been explained in Section 6.2.3 (outcome space).

Table 6.11

Overall Question Difficulty Distribution

Item	Location
1A	-4.59
1B	-3.18
2A	-1.94
5A	-1.33
2B	-0.95
3A	-0.94
4A	-0.87
2C	-0.22
2D	-0.08
4D	0.44
4B	0.88
4C	0.91
1D	0.98
5B	1.11
3D	1.27
1C	1.45
5D	1.57
5C	1.69
3B	1.82
3C	1.97

Additionally, the scoring rubric developed for the writing questions (All the D questions) contained a hierarchy of achievement levels (4 levels) for rating computer program writing competency levels. Figure 6.11 confirms the agreement between the theoretically based expectations and observed item functioning.

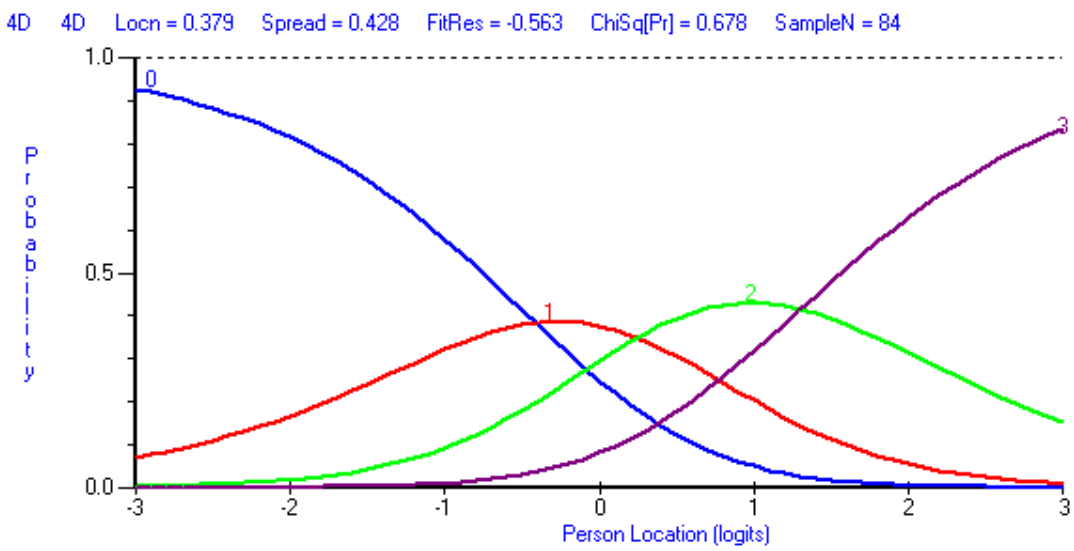


Figure 6.11. Category probability curves for Question 4D

6.3.3. Evidence of the structural aspect

The Structural aspect confirms the internal structure–adopted scoring model, and dimensionality of the construct model (Wolfe & Smith, 2007a). One way to show evidence for this aspect is by demonstrating the dimensionality of the CS1 student competence as deliberated. CS1 student competence construct was posited to be unidimensional consisting of five sub-constructs or topics (See Appendix I). To confirm this, PCA of the Rasch residual was carried out. The PCA of the Rasch residuals (See Table 6.8 Section 6.2.4.5) showed two of the principal components being slightly higher than 2 Eigenvalues, which is an indication of the possible presence of a secondary dimension. A series of independent t-tests conducted by comparing the person estimates from the two subsets of items (three highest positive and three highest negative loadings on the first principal component of item residual) showed the groups differed by 3.5%, which was less than 5% significance and is suggestive of strict unidimensionality (Smith, 2002; Tennant & Conaghan, 2007; Tennant & Pallant, 2006).

6.3.4. Evidence of the generalisability aspect

Generalisability addresses the properties of invariance of the scoring and the interpretations of the scores across different groups of the sample and invariance of meaning across measurement contexts (Wolfe & Smith, 2007b). The DIF statistics of the re-scored data confirmed that none of the items were biased towards the different demographic groups considered for this study (Table 6.6, Section 6.2.4.4). PSI is another indicator of invariance of a measure, which explains the proportion of variance considered true in the calibrated person

scores (Cavanagh, 2009). The internal reliability of the measure or the PSI is 0.87, indicating excellent reliability.

6.3.5. Evidence of the external aspect

The external aspect relates to the test measure's empirical relationship with other external measures of a similar construct (Messick, 1995). Unfortunately, as argued, no other tested instruments of this nature were available to examine this aspect of validity comprehensively. However, the four-level developmental model postulated about learning to computer program as hypothesised in the construct model was differentiated along the continuum at a reasonable level. For example, the PSI of 0.87 suggests that the persons can be separated into at least three distinct groups along the measurement continuum in addition to visual evidence of this separation shown in the item-person map. These partly support the external aspect of validity (Wolfe & Smith, 2007b).

6.3.6. Evidence of the consequential aspect

This aspect relates to the implications of test values and interpretation of scores (Messick, 1989). More specifically, the consequential aspect addresses the consequences of score interpretation as a basis for action as well as the actual and potential consequences of using the test scores, particularly identifying sources of invalidity such as bias, fairness, and distributive justice (Dimitrov, 2014). As there is no explicit way to determine the consequential aspect, arguments can be drawn from other aspects of construct validity to support this aspect (Lim et al., 2009). For example, no DIF of the items evidenced fairness that items were free from demographic and institution bias, and the item-person map revealed the questions were appropriately targeted to the sample (Lim et al., 2009); both ensured none was disadvantaged by construct underrepresentation and irrelevant variance warned by Messick (1989) as the biggest threat to construct validity.

6.3.7. Evidence of the interpretability aspect

The interpretability aspect indicates the extent to which the qualitative meaning of the measurement scores are communicated (Cavanagh, 2009). There is no specific Rasch analysis directly corresponding to the potential consequences of test use (Lim et al., 2009). However, lack of item bias as evidenced by the DIF analysis is a reflection of a person; scores interpreted were valid reflections of a person's ability (Lim et al., 2009). Additionally, a Rasch item-person map could convey information regarding the targeting of questions to the ability levels as well as the ability scores of the individuals, and meaning can be assigned to these individuals with

respect to their competencies. The item-person map calibrates both item and persons on an item-person map called a logit scale, which enables an invariant comparison of both item difficulties and scores of individual students as shown in Figure 6.9 (Section 6.2.4.6). In this map, the vertical dashed line in the middle of the map separating the persons and items is the unidimensional linear logit ruler of the construct representing different levels of competencies from the lowest at the bottom to the highest at the top. This map might seem very simple, however, it is a rich source of information where "differences between persons, between items, and between persons and items can be read directly make comparisons interpreted as 'how much difference exists between any two locations in probabilistic terms'" (Bond & Fox, 2015, p. 57).

The item-person map (Figure 6.9, Section 6.2.4.6) is one of the sources of information to understand student competence in multiple dimensions. Firstly, when the items are structured by a construct model of a hypothetical learning path, as in the case of a CS1 measure, substantive trait level meaning can also be inferred (Embretson, 1996b). The students found code reading (all C) and code writing (all D) questions more difficult than code tracing (all B) and knowledge (all A) questions. This is consistent with past CS1 literature that shows the majority of students were unable to write a fully functional piece of code at the conclusion of a typical CS1 course (Clear et al., 2008; Lister et al., 2004; McCracken et al., 2001; Soloway, Ehrlich, Bonar, & Greenspan, 1982). In addition, the majority of the students' ability levels sit below the meaningful code writing level and code comprehension level (All C questions), which also means approximately two-thirds of the students sit between the uni-structural and multi-structural level of the SOLO taxonomy. Similar sources of information can also be drawn from the item-person threshold map shown in (Figure 6.9, Section 6.2.4.6).

6.4. Phase Three: Correlational Analysis

This section provides the results of the third phase; the correlational analysis of the student and learning environment factors. SPSS was used to conduct ANOVA analysis, and some graphical displays from the Rasch analysis were also used to demonstrate the group variances.

6.4.1. Results

The purpose of this research is to investigate whether the two categories of independent variables presented in Table 5.3 (Section 5.7.1) has a significant association with CS1 student competencies measured by the interval-level logit scores obtained in Phase 2. However, due to

the small number of subjects in some of the demographic groups, only a few were able to test for association. As previously indicated in Chapter 5 (Section 5.7.1), some of the factors were dropped from the original list for likely confounding effects.

6.4.1.1. The impact of programming language choice on student competency

A total of 31 (36.9%) of the 84 students were instructed in Java programming language, 25 students (29.8%) students were instructed in C programming language and 28 students (33.3%) were instructed in Python. The item-person threshold distribution display generated by RUMM2030 as shown in Figure 6.12 is a useful graphical illustration showing the overall trait levels of these three groups. The top panel shows the person frequency and the bottom panel shows the item frequency for each trait level. It reveals that those who were instructed in C show an overall higher level of the trait (M = 0.36 and SD = 1.52) than those who were instructed in Java (M = -0.52 and SD = 2.10), and those who were instructed in Java performed better than those who were instructed in Python (M = -0.83 and SD = 1.62). The higher standard deviation of the Java group suggests that the student trait levels were more spread out along the continuum (logit ruler) than the other two groups. To conclude whether there was a significance difference in competence, an ANOVA analysis of the subjects was performed.

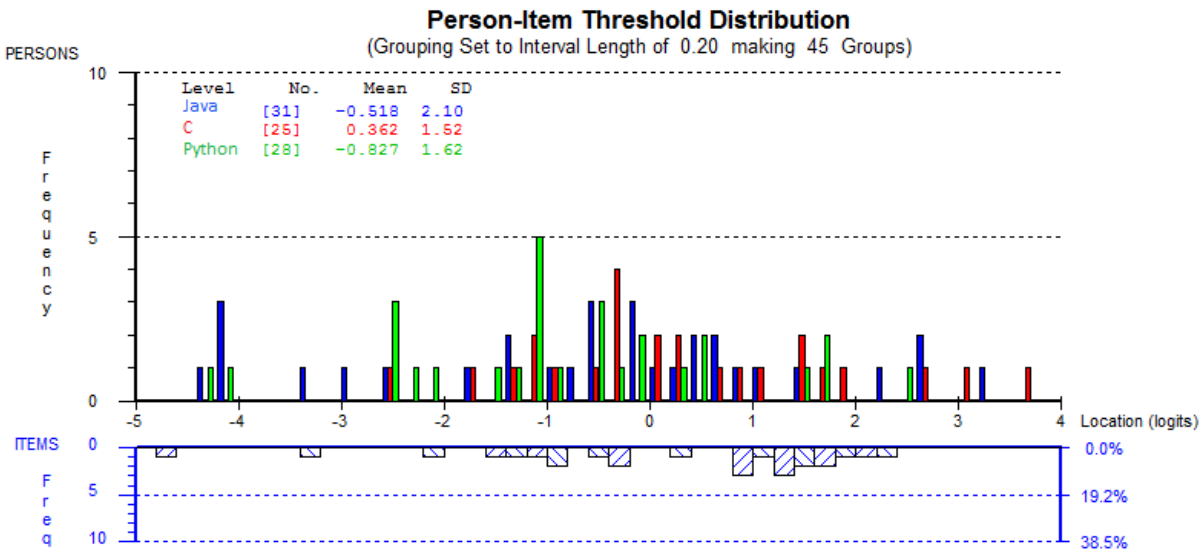


Figure 6.12. Frequency distributions of students based on programming language

However, the results of the one-way ANOVA of these three groups were shown not to be statistically significant at the $p<0.05$ [$F(2, 81) = 3.11$, $p = 0.05$]. The summary of these findings is shown in Table 6.12.

Table 6.12

One-way ANOVA: Effect of Language of Instruction on Student Competence

IV		Sum of Squares	df	Mean Square	F	Sig.
Programming Language	Between	19.93	2	9.96	3.11	0.05
	Within	259.58	81	3.21		
	Total	279.51	83			

6.4.1.2. The impact of programming experience on Student Competency

A total of 18 (21.4 % of the total) students indicated that they had learned computer programming before gaining entry into a CS degree program, and 66 students (78.6% of total students) indicated they did not have any programming experience before enrolling into CS1. The students who had taken High school CS were not considered as having programming experience as students indicated they did not do any serious computer programming with a programming language as part of the course.

The students who indicated having engaged in serious computer programming for more than six months before enrolling into CS1, as shown in Figure 6.13 (the item-person threshold distribution map), show an overall better performance than those did not have computer programming experience. The top panel shows the person frequency and the bottom panel shows the item frequency for each trait level. The students with computer programming experience showed an overall higher level of the trait (M = 0.46 and SD = 1.60) as compared to those without computer programming (M = -0.6 and SD = 1.84). Figure 6.13 suggests that the students with programming experience overall have a smaller spread, with the logit scores ranging between -1.5 to +3.5, whereas those without the programming experience have a larger spectrum of scores ranging between ±4.

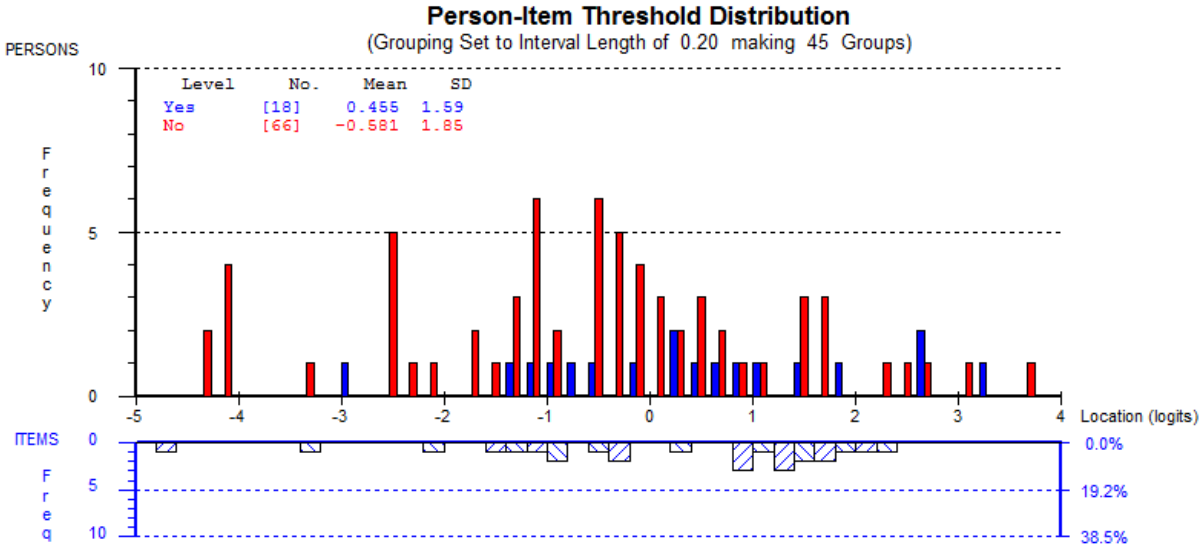


Figure 6.13. Frequency distributions of students with (Y) and without (N) programming experience

Table 6.13

One-way ANOVA: Effect of Prior Programming Experience on Student Competence

IV		Sum of Squares	df	Mean Square	F	Sig.	η2
Programming Experience	Between	15.18	1	15.18	4.71	0.03*	0.50
	Within	264.33	82	3.22			
	Total	279.51	83				

*p < 0.05

A one-way ANOVA between the subjects was conducted to determine if the trait level was different for the two groups. Table 6.13 shows that there was a significant effect computer programming experience on CS1 student competence at the $p<0.05$ level [$F(1, 82) = 4.71$, $p = 0.03$, $\eta^2 = 0.50$]. Together, these results suggest that prior programming experience has a significant impact on CS1 student competence, and based on Cohen's 1988 conventions for reporting effect size, the actual difference in the mean between the groups was medium level.

6.4.1.3. The impact of prior Mathematics background on student competency

A total of 50 (59.5%) of the 84 students indicated that they had successfully completed year 10 Mathematics (Sijil Pelajaran Malaysia (SPM) also known as the Malaysian Certificate of Education, or London GCE O'level or equivalent), whereas 34 (40.5%) students indicated they had successfully completed year 12 Mathematics (Sijil Tinggi Persekolahan Malaysia (STPM) also known as the Higher School Certificate or London GCE Advanced

Level). The descriptive statistics and frequency distributions of the two groups shown in Figure 6.14 suggests that those who were enrolled into a CS program with year 12 level Mathematics performed better (M = 0.18 and SD = 1.82) as compared to those who enrolled with year 10 level Mathematics (M = -0.72 and SD = 1.75).

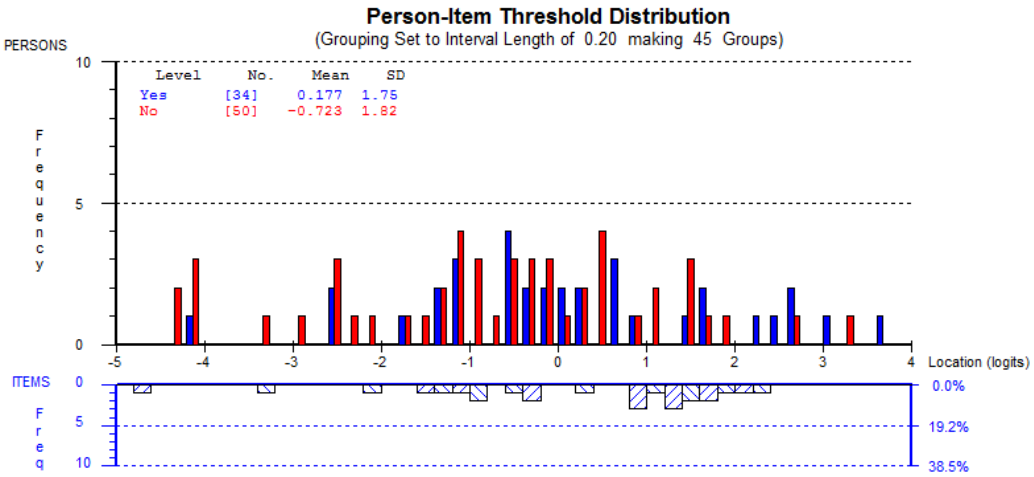


Figure 6.14. Frequency distributions of students with year 12 and year 10 mathematics background

A one-way ANOVA was conducted to determine if there was a notable effect on student competence between those who were enrolled into CS1 with these two levels of mathematics background. Table 6.14 reveals that their mean difference was statistically significant at the $p<0.05$ level for the two groups [$F(1, 82) = 5.11$, $p = 0.03$, $\eta^2 = 0.54$], and the effect size ($\eta^2 = 0.54$) was medium level.

Table 6.14

One-way ANOVA: Effect of Mathematics Background on Student Competence

IV		Sum of Squares	df	Mean Square	F	Sig.	η2
Year 10 or 12 Mathematics	Between	16.40	1	16.40	5.11	0.03*	0.54
	Within	263.11	82	3.21			
	Total	279.51	83				

*p < 0.05

6.4.1.4. The impact of high school stream on student competency

A total of fifty seven (69.5%) students chose the Science stream (comprising of Physics, Chemistry, Biology, English, and Mathematics) as the main subjects in High School. The remaining 25 (30.5%) of the 82 students studied the Commerce stream (which consists of

Economics, Accounting, Business Studies, English, and Mathematics) as their main subjects in High school. The subjects, Maths and English, are compulsory subjects irrespective of stream. Only two students were enrolled in the Arts stream, thus they were excluded from the analysis. The mean and the standard deviation calculated for these two groups and item-person threshold distribution showing the spread of items and persons along the logit scale is shown in Figure 6.15. It suggests that the students who have studied in the Commerce stream (M = 0.21 and SD = 1.87) performed slightly better than those who have studied in the Science stream (M = -0.48 and SD = 1.65).

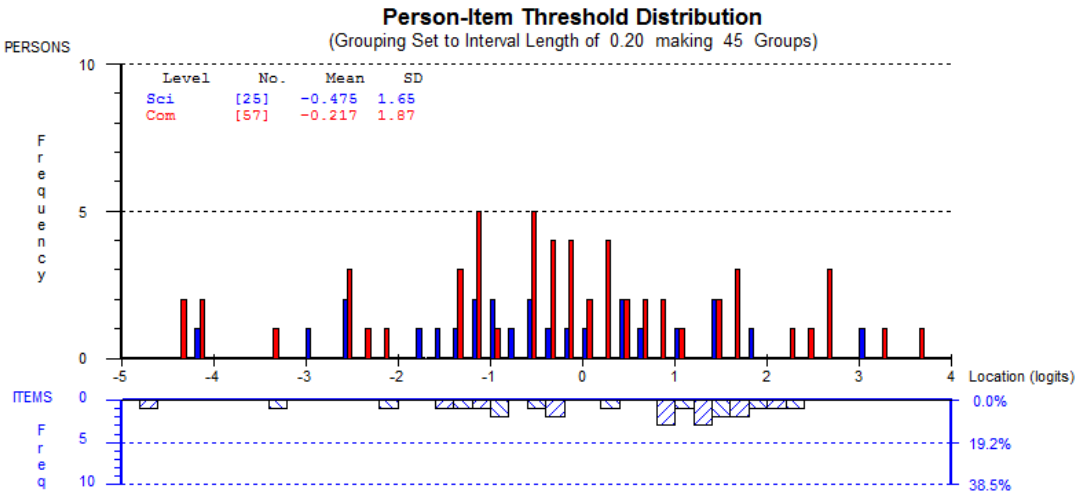


Figure 6.15. Frequency distributions of students studied in Science and Commerce stream

However, a one-way ANOVA conducted to examine the mean difference between those who studied Science stream subjects and those who studied Commerce stream subjects showed the means were not statistically significant at the $p<0.05$ level for the two groups [$F(1, 80) = 0.35$, $p = 0.55$] as shown in Table 6.15. Therefore, it can be concluded the stream of study has no impact on student performance.

Table 6.15

One-way ANOVA: Effect of High School Stream on Student Competence

IV		Sum of Squares	df	Mean Square	F	Sig.
High School Stream	Between	1.158	1	1.158	0.354	0.554
	Within	261.938	80	3.274		
	Total	263.097	81			

6.4.1.5. The impact of high school CS course on student competency

A total of 25 (29.8%) of the 84 students indicated that they had completed Cambridge GCSE/GCE O'level Computer Science (at the completion of year 10), while 59 (70.2%) students indicated that they did not complete a similar course in High School. The students who had taken High School CS did not do any computer programming in a programming language as indicated by these students in the survey (Appendix IX).

The mean and the standard deviation calculated for these two groups and the item-person threshold distribution showing the spread of items and person along the logit scale is shown in Figure 6.16. It suggests that the students who had studied at High school CS1 (M = -0.02 and SD = 1.95) performed slightly better than those who had not (M = -0.50 and SD = 1.78).

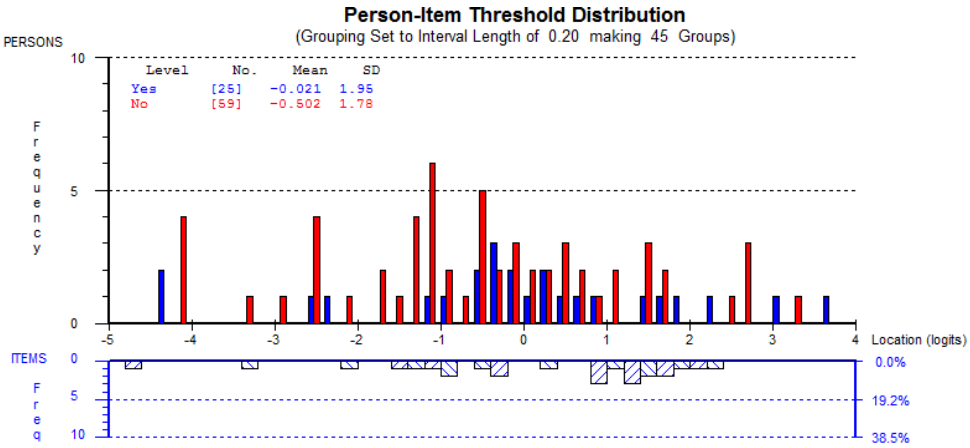


Figure 6.16. Frequency distributions of students who had (Y) and who had not (N) studied high school CS

A one-way ANOVA between the two groups was conducted to compare whether their mean differences were statistically significant. Table 6.16 presents the results of this analysis which shows that the difference was not statistically significant at the $p<0.05$ level for the two groups [$F(1, 81) = 1.21$, $p = 0.28$]. In sum, these results suggest the performance of those who had studied a CS course at High school did not perform better than those who had not studied a CS course at High school.

Table 6.16

One-way ANOVA: Effect of High School CS Course on Student Competence

IV		Sum of Squares	df	Mean Square	F	Sig.
High School CS	Groups	4.06	1	4.06	1.21	0.28
	Within	275.44	82	3.36		
	Total	279.51	83			

6.5. Summary

This chapter presented the results of the investigation, which was carried out in three sequential phases. The first phase was organised into the building blocks of Wilson's (2005) construct modeling approach and the results were presented in that order. Particular attention was given to demonstrating the results of the Rasch diagnostic estimations of a data-to-model fit and the consequent actions. Then, Wolfe and Smith (2007a, 2007b) validity framework was applied to exemplify the validity evidence of the instrument development activities of the investigation. Finally, the results of the correlational analysis were presented. The next chapter discusses the main findings and addresses the main research questions of the investigation.

Chapter 7 – Discussion and Conclusion

This chapter summarises the main findings of the investigation into the development of a CS1 measure. The chapter is organised into three phases, each of which addresses a principal research question in light of the main findings and research literature. The chapter concludes with reflections on the limitations of the study.

7.1. Phase One: Instrument Development

This section first presents the discussion of the overall CS1 measure development's process activities and outcomes. Next the main outcomes of the investigation are discussed. Following this, research question 1 is addressed taking into consideration the key measurement criteria of Wright and Masters (1982) as the benchmark to evaluate the psychometric properties of the measure constructed.

7.1.1. Research question 1

Can a measure of CS1 be constructed?

Wolfe and Smith (2007a) and Wilson (2005) stressed that the development of a new instrument must be initiated by examining the theoretical elements upon which the construct is grounded, in addition to investigating the theoretical basis of instruments of a similar nature. An equal concern must also be given to validity issues in the investigative process. This ensures the quality and rigor of the investigation, which facilitates the collection of evidence in the *post hoc* evaluation of the instrument development process (Messick, 1989). For validity evaluation, Wolfe and Smith (2007a) suggested a sequence of steps to guide the development processes as well as exemplified validity evidence in a Rasch based approach to measurement construction. The construct modeling approach complies with the main principles of these steps. This particular approach enabled the researcher to collate convincing arguments to address the investigation's two central research questions.

As the principal step of Wilson's (2005) construct modeling approach, the first phase began by the development of a construct map, which detailed the theoretical elements underpinning the CS1 student competence variable. The ERG appraised the construct map and the items developed to represent the learning hierarchy and revisions were made accordingly. Similarly, the pilot testing revealed consistency issues between the construct map and actual items, necessitating further enhancement of the items before final administration. These steps demonstrated the validity of the theoretical elements underpinning the construct of CS1 student competence from an external viewpoint. The construct modelling approach takes this a step

further by testing the outcome space of the items empirically, to confirm the hypothesised structure in the construct map by employing a measurement model – the Rasch model.

As argued in the problem statement (Section 1.2) a significant threat to the CS1 research community for conducting pedagogical research to inform instructional practice is the paucity of interval-level measures. The few available measures are typically based on CTT based theories, which at the most can only provide ordinal data. As Stevens (1946) also highlighted, ordinal level scores are not suitable for parametric calculations typically needed by quantitative investigations. The measurement theory employed in this investigation was RMT, which enables constructing interval-level measures, given the response data fits to the model's requirements. As argued by Tennant and Conaghan (2007), Rasch analysis is a unified approach to measurement development addressing key measurement and validity concerns. The Rasch approach allows the measurement outcomes to be tested against fundamental measurement criteria through various procedures available and, as a result improves the psychometric features of the measure. The variety of procedures and statistics provided to address the measurement criteria are also directly related to the validity aspects of Messick. Thus, a measure developed by employing RMT such as the CS1 measure can demonstrate evidence of manifesting the fundamental measurement properties advanced by Wright and Masters (1982), as well as validity aspects.

The Rasch method employs an iterative incremental approach to measurement development. In each iteration of Rasch analysis, the data set is evaluated to improve its properties to get closer to the requirements of RMT expectations. Typically, several such iterations are required to achieve the fit of the response data to the RMT requirements. However, due to the objective approach adopted in this investigation, 18 of the 20 items fit the RMT requirements in the first iteration with excellent reliability. With a re-scoring of the two items with disordered thresholds in the second iteration, all the items and persons achieved the requirements of RMT. Accordingly, the fit of the data means that all the 20 items meaningfully contribute to the construct of the CS1 student competence, and PSI above 0.85 support for individual use of the CS1 measure within CS1 classrooms and research (Fisher, 1992; Pallant & Tennant, 2007). Furthermore, all the Rasch procedures conducted to evaluate the theoretical requirements of the Rasch model, items were tested for unidimensionality, bias, stochastic independence, and monotonicity, all of which revealed that the measure fully accorded to the requirements of RMT, and thus, attained an interval-level measure.

From a statistical viewpoint, when the data fulfils the RMT requirements, it is deemed that an interval-level measure has been created. However, from a validity perspective, this could only be justified by rigorous arguments (Messick, 1989). Validity is defined as "the degree to which evidence and theory support the interpretations of test scores for proposed uses of tests" (AERA et al., 2014, p. 11). In the context of validity, it is important to highlight that the contemporary view of validity is not an empirical index or value attached to the measure, but rather an argument for validity evidence of the interpretation and use of the test scores. For the scores to be considered valid, the interpretation of test scores must be grounded in a construct theory and empirical evidence that demonstrates a relationship between the test and what it purports to measure (Furr & Bacharach, 2013; Sireci & Sukin, 2013).

With this view, validity aspects had been taken as a serious concern in the entire instrument development investigative process. For example, the investigation began by articulating a clear purpose of the research by explicating four research questions, which contribute to the content aspect of validity. Similarly, the investigation began with a detailed review of CS1 literature, which brought together the elements embodying the construct of CS1 student competence and proposed a construct model illustrating the relationships between these elements. These contribute to the substantive aspect of construct validity. This model was then used as the kernel underpinning the item development and then linked back to the score interpretation. As detailed in Section 6.3, the investigative process demonstrated all the aspects of unified construct validity. Thus, this confirms that an interval-level measure of the CS1 student has been constructed and there is a strong argument for the validity of the meaning of the scores and score interpretations. Finally, when the measure is grounded in a substantive theory, a substantive meaning can be given to scores in terms of underlying proficiencies. Therefore, this information could be useful to inform the instructional practice of CS1 (Wu & Adams, 2007), as well as to further understand the construct of CS1 student competence as discussed below.

Although there is as yet no direct empirical evidence revealed about the link and the entwined substructure of the CS1 concepts, it is generally agreed that the degree of sequential dependence of the content of CS1 is far greater than the subjects of other domains (Luxton-Reilly & Petersen, 2017; Porter & Zingaro, 2014; Robins, 2010). Robins (2010) presumes that the case for bi-modal distribution of student scores (a reverse bell curve) frequently exhibited in CS1 course marks (Bennedsen & Caspersen, 2007; Bornat & Dehnadi, 2008; Robins, 2010) is the effect of highly linked "build on" topics of CS1. Based on the simulation data, he argued

that the students become either high performers or failures due to a phenomenon called Learning Edge Momentum (LEM). LEM operates such that, success in acquiring one concept makes learning other closely linked concepts easier (whereas failure makes it harder). Robbins's (2010) study demonstrates several examples that CS1 topics are not self-contained and the successful implementation of a more advanced concept such as a loop requires the prior knowledge of If/Else (the topic prior to loops).

The item location statistics support the argument that the students find the topics harder as they advance. For example, the mean of the four questions representing the first construct (programming fundamentals) is -1.34, whereas, the mean of the second construct (If/Else) is -0.80, and the mean of the third construct (loop structure) is +1.03. This suggests that there is a significant increase in difficulty as the topics advance. One of the implications of this outcome on CS1 instructional practice is to acknowledge that the prior knowledge of preceding concepts is an important starting point for learning a subsequent topic in the CS1 curriculum. This means that without the mastery of the previous topic, there can be little progress made in subsequent topics. Thus, this situation could set the students on a certain path and it could be difficult to deviate them from that path unless an early intervention is provided.

It is one of the main concerns of the entire CS research community that the majority of the students do not achieve the CS1 curriculum competencies at the completion of a typical CS1 course. Consistent with the findings of past CS1 literature as discussed in the problem statement (Section 1.2), this investigation also showed that overall the students found the test difficult with the student mean logit score lower than the mean of the test (all items) scores. Furthermore, the item-person map (Figure 6.9, Section 6.2.4.6) which arrayed the persons and items of the test on the same linear scale, visually shows that the majority of the students were located at the lower levels of the map, suggesting many did not achieve code writing skills as claimed by McCracken et al. (2001) and Clear et al. (2008).

Similarly, consistent with the conceptualisation of the CS1 student competence construct, the item-person map (Figure 6.9, Section 6.2.4.6) also reveals the separation of at least three skills as hypothesised in the construct model, although the highest level was not well separated. For example, the top panel of the item-person map consists mostly of the high levels of code writing and explaining, and in the middle lower levels of writing and explaining questions, whereas the items arrayed at the bottom are tracing and simple knowledge level questions. A significant difference in terms of separation between code reading and writing was not found because students were not able to perform at the highest level of code writing as

discussed in Section 6.2.3. This suggests that there is a hierarchical relationship between code tracing, reading and writing as established by Lopez et al. (2008) in their first study and the follow-up study (Lopez, Sutton, & Clear, 2009). Thus, it could be concluded that higher orders of learning in this hierarchy build upon the lower levels, requiring progressively greater amounts of previous learning for their success in higher order skills. This knowledge needs to be incorporated into CS1 instructional practice by advocating sufficient time to develop lower level skills, which underpin more advanced skills.

This also means about two-thirds of the students sit between the uni-structural and multi-structural levels of the SOLO taxonomy. These results are consistent with the literature on the psychology of learning to computer program. This indicates that the majority of students know the fundamental concepts of programming, but they do not manifest the ability to see the relationship between the component parts of a computer program, thus they do not understand the purpose of the given code segments (all C questions). Similarly, this has implications on code writing, because if the students do not fully comprehend the relationship between the component parts of a program, it is not possible to combine multiple concepts in a logical manner to provide a written solution to a programming task.

It is also important to highlight some aspects of the CS1 measure that are relevant for the lecturers and researchers who might be interested in using the instrument for various pedagogical purposes. One of the defining characteristics of Rasch model measures is that if it is based on a construct theory as proposed in this investigation, the student competence levels can be linked to what students know with respect to the content of the curriculum. For example, it can be deduced that students located at the +2 logits have acquired most of the competencies expected and thus it can be safely concluded that the students can independently compose a programming solution to a given problem combining multiple constructs. Whereas, students located at the -2 and can be assumed to have not acquired a reasonable level of the most fundamental skills as shown in the Item-person map (Figure 6.9 Section 6.2.4.6). Similarly, this graphical illustration is an instant source of information for the lecturers to draw inferences on how well the students have performed overall in the test and how good are the items in capturing the students' abilities. For example, the student mean sitting below the item mean suggests that students find the test slightly difficult.

However, despite the strong argument for the validity of the scores and relevant interpretations to inform instructional practice, it should be noted that none of the psychological measures are free from flaws. Therefore, caution must be exercised in interpreting the results

because instrument development is an iterative, incremental process in which multiple future iterations are necessary to improve the psychometric qualities of the instrument. As noted previously, Messick (1995) asserts that test result interpretations can only be justified by means of rigorous argument.

First and foremost, to fully understand the writing proficiencies of students, all (D) questions (programming writing tasks) needed to be fully-fledged computer program writing tasks rather than writing small pieces of code. Due to the brief nature of these questions, both the researcher and the ERG member, who reviewed the questions, found difficulty rating the students at the highest level (Extended abstract), thus one scored at the highest level. This led to the conclusion that to capture the full range of student competencies as hypothesised, more comprehensive code writing questions needed to be incorporated in future iterations. Perhaps, this might explain why the measure was suboptimal to the students located at the highest level of the measurement continuum as shown in the item-person map (Figure 6.9, Section 6.2.4.6). Similarly, this is also the likely reason that there was no clear separation between the reading and highest level writing questions in the item-person map, in contrast to what was expected in the construct map (Appendix I). However, tests of this nature, which are cognitively complex and time consuming, that is not associated with students extrinsic goals will naturally have less response rates. The main reason is that assessments appeal to students' extrinsic rather than intrinsic goals such as passing a course or gaining admission to college (Usher & Kober, 2012). Therefore, strategies like combining the test items as part of their university formal assessments could improve the students' attitudes towards the test.

Similarly, the instrument needs to be tested on a larger and wider sample. The initial fitting of the data to the RUMM2030 indicated that two of the items (questions 3D and 5D) of the test together with four response categories were not well distinguished by the participants. Commonly, this condition emerges due to many non-responses or very few responses (less than 10) in the categories (Linacre, 2002). This was confirmed by the initial fitting of data to the RUMM2030, which revealed that the middle two thresholds of these items were disordered. The condition was resolved by condensing the middle two categories– each less than ten observations – into one because at least ten observations are required for estimating stable threshold values (Linacre, 2002), which resulted in improved fit with a slight improvement in PSI. However, it should be noted that studies demonstrate that the items exhibiting disordered thresholds are more likely in smaller samples than larger samples (>250; 0-1%) despite best targeting, and arguably the small sample of this study might have precluded demonstrating

category functioning resulting in the incorrect combining of categories (Adams, Wu, & Wilson, 2012; Chen et al., 2014). Another point of interest here is, that unlike attitude tests, it is natural to get a low response rate to questions of competence in cognitive tests similar to the current study, because the questions that exceed one's ability are likely to be left unattempted. The situation is further exacerbated when the items are designed to be scored polytomously, because the sample will be split over the different response categories of the polytomous items, unlike with dichotomous items. In this case, threshold parameters depend on the category probabilities, which are estimated from the response frequencies for each category (Adams et al., 2012). Therefore, researchers should think more carefully about collapsing categories since valuable trait information might be lost due to an inaccurate calculation of threshold orderings as a consequence of a low response rate for categories.

Another interesting argument about the PCM is that while the PCM requires ordinal response categories, it does not require the ordering of thresholds according to Masters (1988). He further explains:

> In the partial credit model . . . the item parameters $\delta_{i1}, \delta_{i2}, \ldots, \delta_{im}$ govern the transitions between adjacent response categories. Order is not incorporated through the values of these locally defined parameters, which are in fact free to take any values at all (p. 23).

Whether ordered thresholds are necessary for PCM has been an issue of interest, which has been studied by several researchers (Adams et al., 2012; Andrich, 2013; Wetzel & Carstensen, 2014). For example, Wetzel and Carstensen (2014) used an empirical approach to elucidate the topic with simulations. They concluded that categories can differentiate between participants with different trait levels despite reversed thresholds, and that category disordering can be analysed independently of the ordering of the thresholds. The same opinion was held by Adams et al. (2012), although Andrich (2013) argued differently. Therefore, despite the common practice of collapsing adjacent category data when disordered thresholds occur, more agreement about the merit of this topic in the context of this investigation and others employing the PCM is required.

Furthermore, the sample size is an important feature of any empirical study in which the goal is to make inferences about a population and generalise the findings. However, in practice, sample selection is always challenged by many factors: this study was no exception as detailed in Section 5.7.1. The minimum sample size (best to poor targeting) for Rasch

analyses is between 64 -100, which is reported to have 95% confidence that the estimated item difficulty is within ±0.5 logit of its stable value (Linacre, 1994). Although, the sample of 85 is deemed generally acceptable for a fairly well targeted measure like this study as shown in the item-person map (Figure 6.9, Section 6.2.4.6), a bigger sample would allow more stable estimates of both items and persons.

Similarly, the gaps found in the item-person map, particularly at the average level, is also an indication that more questions on explaining code are required to obtain an accurate location of the students situated in the middle range. However, one should be extremely cautious about designing these questions despite them being extremely popular in the BRACElet projects (Lister et al., 2006; Whalley et al., 2006) and other studies reported later (Corney et al., 2014; Venables et al., 2009). There has been some controversy as to whether these questions are really testing the ability of students to read and understand code (i.e. competence) or the ability of the students to express themselves in English (Lopez et al., 2009; Snowdon, 2011). In this study, despite a clear illustration with an example of how to attempt to explain the code questions, some students did not attempt them, which could be a reflection of their discomfort to these types of questions. Although the CS1 research literature and CS2013 curricula concedes the importance of code explaining, very little emphasis is given to developing the skill in CS1 classrooms. Therefore, in future iterations, it might be useful to conduct a few pre-training sessions to expose the students to these types of questions.

In brief, when the data fits the Rasch model, then all the attractive features described in the RMT throughout and including the measurement criteria, are achieved (Bond & Fox, 2015). However, the research question associated with this phase can be best addressed more analytically with reference to a standard such as the four measurement criteria of Wright and Masters (1982) (See Section 5.5.5 for details of this criteria) as discussed below.

(a) Unidimensionality:

The most fundamental principle of a measure is the unidimensionality. This is because unidimensionality is a basic requirement for the valid calculation of total scores, in addition to being an important property that allows for an unambiguous score interpretation, and comparison between individuals (Smith, 2002; Stout, 1987). As discussed in Chapter 6 and Chapter 7, Rasch software such as RUMM2030 provides several tests and indicators of unidimensionality. Therefore, the RUMM2030 unconstrained PCM – RUMM2030 computer program – was used to test the extent to which the data fit the RMT requirements, especially

the requirement that items indicate a dominant construct. Overall fit test, item-trait interaction chi-square are the first indicators of unidimensionality. Overall fit supported by a nonsignificant item-trait interaction chi-square statistic, and individual item fit ranging between -2.5 and +2.5 suggests that the measure forms a unidimensional construct (Andrich et al., 2011).

However, fit statistics alone are not sufficient to warrant unidimensionality as they are not always sensitive in detecting unidimensionality (Smith, 2002; Tennant & Pallant, 2006). A combined approach to unidimensionality testing suggested by Smith (2002) was used to evaluate unidimensionality, as discussed in Section 6.2.4.5, which suggested that the CS1 measure attained strict unidimensionality requirements. Additionally, DIF and locally dependant items could also be an indication that items share another dimension. The tests conducted to evaluate these conditions, as discussed in Section 6.2.4.4 and Section 6.4.4.5, revealed that none of the items manifest DIF or local dependency. This adds further evidence that the 20-item test is measuring a single dimension as hypothesised in the construct model (Figure 3.3, Section 3.2.5) of this study.

(b) Qualification:

Qualification requires that the data can be compared. To achieve this criterion, Guttman (1950) stressed that in instrument scales:

> If a person endorses a more extreme statement, he should endorse all less extreme statements if the statements are to be considered a scale ... We shall call a set of items of common content a scale if and only if a person with a higher rank than another person is just as high or higher on every item than the other person. (p. 62)

Adherence to the response structure to Guttman's scale pattern can be assayed by employing various Rasch model statistics. When the data fits the Rasch model, the items are ordered in relation to person ability as shown in Figure 6.9 (Section 6.2.4.6). The figure shows that both difficult and easy items are answered only by persons with higher abilities, whereas the easier items are answered by persons with lower abilities as expected in the Guttman scale. Similarly, the higher response categories (indicated by item thresholds) of polytomous items as revealed in Figure 6.9 are scored by the students with higher abilities, whereas the lower categories are scored by persons with lower abilities. This accordance in persons selecting categories could also be revealed through an examination of the ICC of polytomous items. The

ICC of the four polytomous items constituting the 20-item CS1 measure was shown to be working and monotonically increasing as exemplified and detailed in Section 6.2.4.1.

(c) Quantification:

Just as physical variables are measured in common units, psychological variables are also required to have a common unit of quantification for communicating the definite magnitude of the quantity; in this study's context, how much of CS1 competence is in a unit of definite magnitude. In fact, a unit is also required to provide repeatability in equal units, which can be additive along the length of the measurement continuum. However, it should be noted here that the additive nature of all physical science measures is not always demonstrated in repeated physical actions on concrete units, rather it has to be discovered (or constructed) indirectly such as how measurements of derived quantities like density are discovered (Cavanagh, 2007). Therefore, a unit is attained by a process of some kind "which can be repeated without modification in the different parts of the measurement scale" (Thurstone, 1931, p. 257). To derive repeatable units along a scale, firstly it is a requirement of the RMT that observations obtained from a test conform to the rule that the persons with the greater ability correctly answer more of the items, for example, easier items are more likely to be answered correctly by most persons (Bond & Fox, 2015). In other words, the data must meet the qualification requirements as discussed earlier. In this study, model fit means that the data affirms to these principles. Accordingly, the logarithmic transformation of the raw scores observed from the ordinal data into the interval data was fulfilled with the Rasch Partial-credit model. The diverse Rasch displays presented in various sections of Chapter 6 including the item-person map (Figure 6.9, Section 6.2.4.6), the ICC's (Figure 6.4, Section 6.2.4.1) and the item-person threshold distribution map (Figure 6.10, Section 6.2.4.6) were all calibrated in logits resulting from the process of the logarithmic transformation adopting the logit as its iterative unit. Therefore, these are evidential data to justify that the CS1 measure quantifies the construct of CS1 student competence using a repeatable unit along a measurement continuum, thus satisfying the requirements of quantification.

(d) Linearity:

Linearity is the property of a mathematical relationship or function, which means that it can be graphically represented as a straight line. Many kinds of measurement imply a linear continuum of some sort such as length, price, volume, weight, or age (See Thurstone & Chave, 1929, p. 11), which is always an abstraction as described by Thurstone (1931). When the same

idea is applied to "scholastic achievement, for example, it is necessary to force the qualitative variations into a scholastic linear scale of some sort" (Thurstone & Chave, 1929, p. 11). Therefore, the person and item estimates are forced to be linear and interval by the Rasch model (Bond & Fox, 2015).

As discussed throughout and at the beginning of this section, the fit of the data to the Rasch model is the determining factor for how well the data has achieved each of the measurement properties. Bond and Fox (2015) describe fit as the quality-control principle for deciding whether the observed data of items and persons are close enough to the Rasch model's requirements to be considered linear interval scale measures. All the fit statistics evaluated were revealed to have a good fit of the data to the Rasch model as presented in Chapter 6 (Section 6.2.4), which confirms that the criteria for interval scaling were met in the data. The linear properties of the scale are illustrated in the item-person map (Figure 6.9, Section 6.2.4.6) and item-person threshold map (Figure 6.10, Section 6.2.4.6), which shows both the items and persons have fixed positions along one straight line. In other words, both person and item difficulties are quantified and calibrated on the same scale, in equivalent logit units, which is a requirement for linearity. Moreover, the item-person map shows that one logit positive difference between any person and any item on the scale has the same stochastic consequence. This makes the logits equal intervals, hence, it can be concluded that the scores of the measure sufficiently manifest linearity.

7.2. Phase Two: Validity Evidence

This section presents the discussion of the results of the validity evidence by answering the research question associated with this phase.

7.2.1. Research question 2

What evidence is available to support an argument for the validity of the project?

The true concept of the Rasch approach to measurement development follows a construct modelling approach where the item design is based on *priori* construct theory. Unfortunately, to achieve the fit of the data to RMT, items do not necessarily have to be constructed upon a construct theory, thus, making Rasch models too easy to apply (Stenner, 2001). Consequently, Rasch software is commonly used to estimate item and person measures on a logit scale and claim to have achieved an interval-level measure without any *priori* framework grounding the investigation (Stenner, 2001). However, many advocates of the Rasch model refute these practices (See Bond & Fox, 2015; Messick, 1995; Stenner, 2001; Wu

& Adams, 2007), and emphasise the importance of item calibrations from a construct theory because it is directly related to validity issues (Wu & Adams, 2007). For example, Wu and Adams (2007) asserted that the inferences made from test scores and the use of test scores should reflect the definition of the construct. Similarly, Stenner (2001) also highlighted the importance of theory-based item construction by saying that "theory and item engineering improve as we bring observed item difficulties and theory-based item calibrations into closer and closer coincidence" (p.804). Therefore, when these two aspects are closely aligned, superior Rasch measures are produced and the validity argument of the measure is more defensible.

In the current investigation, the instrument construction process began with a similar approach: the development of a theoretical framework upon which the items were generated and then were empirically tested by employing RUMM2030. The purpose of this section is to assay the validity account of this investigation, which answers Research Question 2. The arguments will be structured around the validity aspects of Wolfe and Smith (2007a, 2007b) as discussed in Section 5.6.

(a) Content aspect of validity:

The content aspect of validity includes an unambiguous statement of the purpose of the study which could be explicated by the study's aim or the research questions (Wolfe & Smith, 2007a). The aim was made explicit with the clear statement of purpose "creating linear interval scale of CS1 student competence", and the first two research questions were articulated to achieve this purpose. Similarly, another way to support the content aspect is by elucidating the domain of inference (Wolfe & Smith, 2007a). The study's instrument development was underpinned by learning theories informed by the CS1 literature which suggests that learning to program requires the ability to trace, read, and write, forming a learning progression hierarchy in that order (Lopez et al., 2009; Lopez et al., 2008).

Similarly, explicating the types of inferences, and potential constraints and limitations clarify the purpose (Wolfe & Smith, 2007a), and hence reinforces the content validity argument. The inference to be drawn from the study was the programming competence of CS1 students on five fundamental CS1 concepts when they conclude a typical CS1 course. The small sample size had been iterated as the main constraint for validity in Chapter 5 (Section 5.5.5) as well as in this Chapter 7 (Section 7.1.1) providing evidence to account for the content aspect of validity.

The instrument specification, such as the construct model, construct map, item format, scoring and scaling model, all add value to explicating the content aspect of validity (Wolfe & Smith, 2007a). The construct model can be an external or internal model. An internal model depicts the components, facets, elements and factors and the hypothesised relationship between these components (Cavanagh & Koehler, 2013). On the other hand, an external model represents the relationship between the target construct and other constructs (Cavanagh & Koehler, 2013). Figure 3.3 (Section 3.2.5) is an internal model developed upon information sourced from empirical CS1 literature, CS1 curriculums, and ERG reviews to show the key elements of the CS1 student competence construct. These tasks are among the activities listed by Wolfe and Smith (2007a) in their suggestive list of sources for contributing to the development of construct models. The internal structure of the construct model was explicated in the construct map. For example, the construct map theorising the task order or the task difficulty was explicated based on the assumed skill hierarchy of learning to program and their levels were defined using SOLO taxonomy (Appendix I). The skills were postulated to form a learning progression – tracing, reading and writing – with each task assessed against a SOLO level. For example, the students at the tracing level of learning sophistication could only trace the output of the code segments; the students at the reading level could skilfully summarise the purpose of a given piece of code in addition to tracing the code; and the students at the highest level could write a full-function programming code to a given task in addition to tracing and reading the code. The construct map explains in detail the student behaviours at each level to differentiate between the stages of development of the learner as well as to order the levels. This kind of organisation of levels is important when the construct of interest is hypothesised to be cognitively developmental, when the attainment of prior levels is a prerequisite to mastering the following levels in the hierarchy of levels (Cavanagh & Koehler, 2013) as theorised for learning to computer program. For example, in the Cavanagh (2009) construct map, a similar ordering was used. Similarly, Wilson and Sloane (2000) and Wilson (2004) encouraged and exemplified a construct map to theorise student learning.

Item quality attributes such as unambiguous phrasing, accurate answer keys and suitable reading levels for the target population can also demonstrate the content aspect of validity (Messick 1996). The quality of the questions, which includes unambiguous phrasing, were checked with high school students studying computer science as well as by the ERG, which was then piloted to a sample of 10 students, resulting in a few amendments to the original questions. Both the ERG review and the pilot test revealed that the items adequately

represented the CS1 student competence construct suggesting there was sufficient construct representation. These forms of appraising were mainly used traditionally to evaluate the content aspect of validity (Messick 1996). The technical quality of items could also be empirically tested via a scaling model such as Rasch software. In the current study, RUMM2030 software was used to test the technical quality of the items. Demonstrating technical quality of the items is another form of evidence for the content aspect of validity (Lennon, 1956; Messick, 1989). The Rasch analysis of the item residuals showed good fit as displayed in Table 6.5 (Section 6.2.4.3), and the PSI was 0.87, indicating that the items were able to separate the participants along the measurement continuum. Additionally, the item- person map, which could also be used as a source of evidence for the technical quality of items (Lim et al., 2009), revealed that the items covered a comprehensive range (about ±4 logits) of the construct under investigation. The items contributed to the single construct indicating that the construct was adequately covered. This means the construct underrepresentation, warned by Messick (1989) as one of the main threats to content validity, were reasonably addressed.

The CS1 measure demonstrated several examples of the content aspect of validity including content relevance, representativeness and technical quality of items. However, several more sources could be incorporated to improve overall content validity in future iterations. For example, in addition to the subjective ERG reviews of the construct maps and the items, objective feedback can also be combined to corroborate validity evidence further. For example, objective feedback about the degree of agreement between the ERG members about the item difficulties can be evaluated with techniques such as inter-rater reliability. The content validity index is a very widely used approach to defend the content validity aspect (Davis, 1992; Grant & Davis, 1997) which could be applied in the future iterations.

(b) Substantive aspect of validity:

The substantive aspect of validity explains the theoretical rationale for observed consistencies in the data with respect to a *priori* model (Messick, 1989; Wolfe & Smith, 2007a). The items for the CS1 measure were based on the theoretical model of four distinct programming skills (Lopez et al., 2009; Lopez et al., 2008) and five concepts fundamental to learning to program (Tew & Guzdial, 2010) as illustrated in Figure 3.3 (Section 3.2.5). The person fit data of the measure revealed no misfitting persons, which is an indication of response patterns being in line with the theoretical model (Smith, 2001). Similarly, the location of the majority of the items also conformed to this a priori conceptualisation as shown in Table 6.10 (Section 6.3.2), which is another way to substantiate the observed consistencies in the data as

predicted by the model. As advanced in the construct map, the tracing questions are the easiest and are expected to span over the bottom of the learning continuum, whereas reading questions cover the middle and the writing questions are the most difficult, and thus there is an expectation that they are positioned at the top of the learning continuum. The item-person map, Figure 6.9 (Section 6.2.4.6), provides empirical support that the majority of questions fulfilled this expectation. Additionally, the polytomous items (writing questions) were designed to be scored using a four-point response scale, in which the respondents who had low ability were expected to score only on the lower levels of the scale, whereas those with greater abilities were expected to score the higher levels accordingly. The category probability curves (see Figure 6.5, Section 6.2.4.1) for these items were shown to accord to this specification demonstrating the substantive aspect of validity.

Additionally, the extent to which the type of responses or the selection of responses to the items by the individuals completing the test items fit the intended construct could be evaluated to provide further evidence of the substantive aspect of validity (Brown, 2010). Two points were stressed as central by Messick (1989) to evaluate this aspect: (a) the need for items that will provide appropriate sampling of the domain process in addition to traditional domain content, and (b) the need to move beyond the traditional ERG opinion to accrue empirical evidence of whether the process being measured truly engages the individuals responding to the task items. For example, the item-person map (Figure 6.9, Section 6.2.4.6) shows that there were enough items to test the students' abilities and that an adequate sample of students attempted the items, suggesting appropriate sampling to test the items.

The literature on CS1 student competence identifies differences in scores between students with prior programming experience and those enrolled in a CS1 course without programming experience. Similarly, the Mathematical background of the students was also found to be positively associated with CS1 student competence. Consistent with past CS1 literature, these two factors were shown to be statistically significant with CS1 student competence as discussed in Section 6.4.1 (that is, the mean score without programming experience = 0.46 logits and the mean score with programming experience was 1.84 logits [$F = 4.709$, $p = 0.033$]). Based on the examples and empirical evidence presented here, it can be concluded that there is sufficient evidence to support that the CS1 measure has the substantive aspect of validity.

(c) Structural aspect of validity:

The structural aspect concerns the internal structure of the construct model (Wolfe & Smith, 2007a). One way of providing evidence for this aspect is by testing whether item interrelationships support the conceptual framework of the instrument (AERA et al, 1999). The conceptual framework of the CS1 measure posited that the five topics (constructs) form a single dimension, which was confirmed by a strict dimensionality test proposed by Smith (2002). Details of this procedure were provided in Section 6.2.4.5 with supporting evidence (Table 6.8, Table 6.9 and Figure 6.8). Furthermore, the local dependence between the items could also be an indication of the presence of more than one dimension (Franchignoni, Giordano, Marcantonio, Alberto Coccetta, & Ferriero, 2012). Thus, the Rasch residual correlation was examined, and it was found there was no evidence of locally dependent items. Therefore, there is clear evidence that the 20-item test is measuring a single dimension.

Messick (1989) stressed that the internal structure of a scale should be consistent with what is known about the internal structure of the construct domain. For example, it was suggested by the CS1 literature that learning to program forms a hierarchy of three fundamental skills. The construct model, the construct map, and the items were based on these skills. Smith (2001) suggests that the assumed learning trajectory defined in these models could be answered by two working assumptions of the Rasch model. Firstly, the persons with greater ability are more likely to answer more items correctly than persons with lesser ability. Similarly, the second assumption is that easier items should be scored higher than more difficult items by all persons, regardless of their abilities (Bond & Fox, 2015). The construct model, the construct map, and the items were designed according to this hierarchy. The excellent fit of the data to the Rasch model expectations, no misfitting persons, and no item bias confirms the hypothesised internal structure.

Additionally, the item-person map which arrays the items from the most to the least difficult is an informative display to evaluate how well the test items are defining a variable (Boone, 2016). As hypothesised, the item difficulty measures (logit scores), as well as the item-person map, shows that the majority of tracing questions (all [A]) were located at the lower continuum of the item-person map, the majority of the reading (all [C]) questions covered the middle, and most of the full-functional writing (highest level of all [D]) questions were located at the top of the item-person map, which supports the hypothesised order of item difficulty as explicated in the construct maps, thus adding further structural evidence for validity.

(d) Generalisability aspect of validity:

Generalisability addresses the properties of invariance of the scoring and the interpretations of the scores across different groups of the sample (Dimitrov, 2014), and invariance of meaning across measurement contexts (Wolfe & Smith, 2007b). DIF statistics estimated for various demographic groups confirmed that none of the items were biased towards the different demographic groups considered in this study (See Table 5.3, Section 5.7.1); adding further evidence to the claim of the CS1 measure manifesting the generalisability aspect of validity evidence. The statistics such as PSI are another indicator of the generalisability of the results of the CS1 measure. It represents invariance of the measure, which explains the proportion of variance considered true in the calibrated person scores (Cavanagh, 2009). The internal reliability of the measure or the PSI is 0.87, indicating excellent reliability, which further substantiates the account for the generalisability aspect of validity. Generalisability also pertains to answer whether the findings are applicable in other research settings. The data for this study was collected from three different institutes, two from the Maldives and one from Malaysia. Most importantly, each of these universities used a different programming language for CS1 instruction. However, despite these conditions, no DIF due to membership of the different institutes was detected. This verified the applicability of the instruments to other countries and other languages.

(e) External aspect of Validity:

The external aspect relates to the test measure's empirical relationship with other external measures of a similar construct (Messick, 1995; Wolfe & Smith, 2007b). Unfortunately, as noted previously, no other instruments of this nature were available to examine this aspect comprehensively in the CS1 literature. Wolfe and Smith (2007b) discussed several other ways of exemplifying the external aspect of validity. One way to demonstrate this aspect is by monitoring the changes in individual person measures in the pre-test and post-test positions after an intervention study; however, this is beyond the scope of this investigation. Finally, it is also possible to draw evidence for this aspect if developmental models were created during the instrument development phase (Wolfe & Smith, 2007b). Item development in the current investigation began with a developmental model (See Figure 3.3, Section 3.2.5 and construct map shown in Appendix I) of the three essential programming competencies, which suggests forming a hierarchy as discussed in the literature review. Consistent with this model, the PSI of 0.87 warranted at least three meaningful consistent distinctions about the student competencies (Linacre, 2014). Therefore, this evidence could be documented as preliminary evidence for the external aspect of validity.

(f) Consequential aspect of validity:

This aspect relates to the implications of test values and the interpretation of scores (Messick, 1989). More specifically, the consequential aspect addresses the consequences of score interpretation as a basis for actions as well as the actual and potential consequences of using the test scores, particularly identifying sources of invalidity such as bias, fairness, and distributive justice (Dimitrov, 2014). In general, the consequential aspect seeks to prevent all sources of bias and unfairness, which may impact on the score of interpretation. As discussed in Section 6.2.4.4 and elsewhere, several measures were taken to prevent such bias. For example, a DIF analysis performed with several demographic groups revealed that none of the items were biased in favor of a particular group, confirming that no one's ability was miscalculated as an implication. Similarly, the item-person map revealed that the questions were appropriately targeted to the sample suggesting that there were adequate content representation and items to test the ability of the participants (Messick, 1989, 1995).

(g) Interpretability aspect (added by Wolf and Smith (2007b):

The interpretability aspect pertains to establish the degree to which qualitative meaning can be attributed to quantitative measures (Wolfe & Smith, 2007a). Its concern is about what meaningful inferences can be drawn about CS1 student competence. Rasch analysis provides many useful displays and other objective data that are both clear and easy to interpret. However, the item-person map (Figure 6.9, Section 6.2.4.6) is one of the effective sources of communication (Cavanagh, 2009). The item-person map shows the ability ranges of the students (roughly between ±3.5), as well as information such as whether the test is easy or not for the sample. For example, the majority of the students are below the average item difficulty level (0.0 logits) on the map suggesting most of the students find the test difficult. Most importantly, it enables comparison in logit differences between students and students, items and items as well as students and items. The item-person map shows that the items were well targeted to the students' ability levels; however, it reveals that there are gaps, more specifically, insufficiency of items to capture students at the multi-structural level and is to some extent suboptimal in capturing the abilities of those at the top.

The above discussion suggests that the validity requirements of the Wolfe and Smith (2007a, 2007b) framework were met in the development activities of the CS1 measure. However, it is noteworthy that it was simplified as a consequence of the application of a

particular approach to research and taking validity as the foremost concern in all aspects of the investigative process. The following discussion highlights these points.

Firstly, the relationship between the validity view and the research approach was made evident by selecting a research approach for the instrument development process that was congruous with aspects of validity evidence. For example, the instrument development model (Wilson's construct modeling approach) adopted an *a priori* approach consisting of methods that are synchronised with the structured view of unified validity and validity steps as suggested by Wolfe and Smith (2007a, 2007b). In this approach, the instrument development process begins with a construct model explicating the purpose of the investigation. This is followed by the item design, and then empirically testing the structure and functioning of the hypothesised construct of interest as characterised in the construct map by the application RMT. This approach is responsive to all aspects of unified validity. For example, a theoretical model characterising the behaviour of the task performer similar to that of the construct map developed in the first building block of the construct modelling approach is required to support the content validity of the unified validity framework. Similarly, the substantive aspect of validity requires the theoretical rationales for the observed consistencies in the test responses to be demonstrated empirically. The construct modelling approach employs Item Response Theory statistical models such as RMT (building block 4), to link the response data back to the theoretical model. The variety of outputs provided by RMT computer programs could provide supporting evidence as to whether the participants' response structure accorded to the *priori* models. For example, item difficulty measures estimated by a Rasch analysis of the data could be used as evidence to support whether items conformed to the *priori* models, thus lending support for the substantive aspect of validity. This shows the measurement development approach is complementary to the validity view.

Secondly, a similar relationship also exists between the validity theory and the measurement model employed by Wilson's approach for testing the conformity of the responses data to the construct map. The principles of RMT are responsive to Messick's unified view of validity. This was demonstrated by Wolfe and Smith (2007a, 2007b) by exemplifying the Rasch outputs to collate all the aspects of unified validity. Moreover, the current investigation as discussed in the next section also followed the same approach exemplified by Wolfe and Smith. Similarly, the statistical estimations and graphical displays of the Rasch analysis of the data enabled the articulation of a convincing argument for the *post-hoc* evaluation of validity for this investigation. For example, the fit of the CS1 measure data to the

Rasch model is an indication that the individual items are invariant across groups of participants, and that measures are stable across instrumentation and scoring designs (Cavanagh, 2009). Similarly, DIF analysis supported that the items of the CS1 measure are invariant across different demographic groups. These are some of the Rasch statistics, which supported the generalisability aspect of validity evidence. Therefore, the Rasch analysis of the data was a powerful tool for evaluating the validity of the investigative process against unified validity criteria.

Thirdly, a similar relationship could also be found between the research questions and the measurement model or the statistical model. A research question is a highly focused question that addresses a hypothesis (Cavanagh, 2009), thus requires evidence to support whether it has been achieved or not. This suggests that the choice of measurement model should support the inferences needed to test the hypothesis of the research question; thus the choice of measurement model is influenced by the research question. For example, as demonstrated in the previous section, RMT was used as a tool to draw inferences for demonstrating that the research objectives have been achieved. From a validity perspective, the extent to which the inferences supported the attainment of the research objectives contributed to validity as objectives stating the reason for constructing a measure and articulation of purpose is one way to support the content aspect of validity (Wolfe & Smith, 2007a). Similarly, the selection of a measurement model in the test specification also provides further evidence of the content aspect of validity. Therefore, it could be concluded that every aspect of the instrument development process including articulation of research objectives, methodological decisions, and measurement model selection should be informed by considering the validity aspects as the key concern.

In conclusion, the attainments of Wolfe and Smith's (2007a, 2007b) validity aspects in the current investigation have been briefly discussed. Next, the importance of considering validity issues in all aspects of the research design and investigative process has been highlighted. The next section follows the discussion of the correlational study, which answers research question 3 and 4.

7.3. Phase Three: Correlational Study

This section presents the discussion of the correlational study answering the two research questions associated with this phase.

7.3.1. Research question 3

Are there statistically significant associations between student competency in CS1 and student and classroom learning environment characteristics?

Although the main aim of the current investigation was to construct a linear measure of CS1 student competence, the data collected for the investigation can be taken further because several kinds of demographic data from the participants were collected (prior programming experience, Mathematics background, High School CS course, Programming language of CS1 instruction etc.) for instrument testing. The effects of some of these variables to CS1 student competence have been profoundly debated within the CS1 literature. Recently, the choice of programming language for CS1 instruction has also been the subject of much debate, especially due to the overwhelming number of new programming languages available for CS1 instruction. The CS1 literature shows that some of these variables such as student prior programming experience and mathematics background influences CS1 student competence. Other variables such as gender and high school stream have equivocal influence. Despite the large array of CS1 literature investigating the role of these variables on student competence, no research in the past has tried to measure the dependent variable (CS1 student competence) at the interval-level. Interval-level scores allow parametric analysis such as ANOVA to be performed legitimately; hence the validity of the outcomes of this investigation can be better defended. Therefore, the results of this phase will further enlighten the debate on the role of these variables on CS1 student competence.

As revealed in this study, prior programming experience and mathematics background of the student are shown to have a significant impact on CS1 student competence and are consistent with the majority of the research investigations into these two factors. For example, Hagan and Markham (2000) note that the more programming languages the students were familiar with prior to enrolling in CS1, the more successful they were, at least in the first CS course. There are also several other empirical studies that support the premise of prior programming experience (Bergin & Reilly, 2006; Hagan & Markham, 2000; Strnad et al., 2009) and a Mathematics background (Bergin & Reilly, 2006; Evans & Simkin, 1989; Jerkins et al., 2013; Lambert, 2015; Leeper & Silver, 1982) as strong influencing factors on student performance. One-way ANOVA performed to investigate whether differences exist between these groups showed those with programming experience performed better than those without programming experience (the mean score without programming experience was 0.46 logits and the mean score with programming experience was 1.84 logits,($F = 4.709$, $p = 0.033$). The effect

size, however, was not very strong ($\eta^2 = 0.50$). Similar findings were also observed between students with different levels of mathematics background. Therefore, the outcome of this study further reinforces the positive effect of prior programming experience and mathematics background on CS1 student competence debated in the CS1 literature.

However, it was surprising to find that there was no significant difference between the groups who had taken a CS course in high school and those who had not. One of the reasons could be the limited focus and coverage of programming topics in the curriculums studied. Most of the students who had taken High school CS indicted they did not do serious computer programming with a high-level programming language as part of their CS course. Therefore, despite having studied computer science as a subject at high school, the students were usually not exposed to programming languages, particularly translating algorithmic code to computer programming code. Taking this into consideration, the lack of statistical significance was not unanticipated.

Although there have been a few studies (Byrne & Lyons, 2001; Rountree, Rountree, Robins, & Hannah, 2005) that suggest that students enrolling in CS1 with science backgrounds (for example those who have studied physics or chemistry) perform better than those who come from humanities backgrounds, there is little empirical evidence in the CS1 literature. The current investigation revealed that there was no significant difference between students who enrolled in the CS1 course from different high school streams. Therefore, it can be concluded the stream is not related to CS1 student competence, rather the variable that confounds the student competence is the mathematics background.

Unarguably, students will perform better when the learning environment is aligned with the learning needs of the course. The CS1 literature reports many such factors, of which some are unique to the CS1 learning environment. The choice of programming language and the programming paradigm of CS1 instruction are topics of long-time debate, which are unique to CS1 learning environments. The current investigation, to some extent, enlightened the debate on an ideal programming language for CS1 instruction. This study examined whether or not the CS1 student competence differed by the programming language that was taught in the course. The study revealed that there was no significant statistical difference between the students who were instructed with different programming languages. The study was able to test three different programming languages, Java, C/C++, and Python which has been most popular in instructing CS1 as confirmed by the CS1 literature (Lewis, Blank, Bruce, & Osera, 2016; Shein, 2015). Recently, Python programming language has been cited as the favorite language

for CS1 (Moons & De Backer, 2013). However, this study revealed that the student competence score does not vary considerably by a statistically significant level when students are instructed with different programming languages. Given the affirmative view about Python as the most appropriate for CS1 instruction (Agarwal & Agarwal, 2005; Agarwal et al., 2008; Norman & Adams, 2015), the result was somewhat unanticipated.

However, a more critical analysis of these studies revealed that they were not conclusive objective studies; rather they were more based on hands-on experience reports of using Python and using it as CS0 (preparatory course for CS1) courses. For example, Norman and Adams' (2015) study showed the students' performance was improved when students were instructed with Python. However, the authors were unable to conclude as to whether the little improvement made by students was due to programming language or other confounding variables. On the other side of the spectrum, there have been studies confirming no statistical difference between the groups of students who were instructed with different programming languages (Enbody & Punch, 2010; Enbody, Punch, & McCullen, 2009; McPheron et al., 2015). Similarly, more rigorous studies such as that of Stefik and Siebert (2013) did not focus on the impact of student performance, rather it supported the findings that students find learning programming syntax with Python easier as it is more language-oriented than the other two. Therefore, based on these arguments, along with the results of the current investigation, it could be concluded that programming language has no impact on student competence. However, a noteworthy point here is that the result of the current study might have been confounded by institutional factors such as the learning environment.

In this investigation, the sample was selected from different institutions. Therefore, it can be argued that the result was confounded by factors such as lecturers' experience and qualifications and resources available. On this note, although differences between groups were not significant, overall, the Python instructed student scores (Asia Pacific University of Technology (APU)) were lower than the overall student performance of the other two institutes (The Maldives National University (MNU) and Villa College). Comparing the learning environment factors, APU has a long-standing reputation as a quality tertiary education provider with leading-edge learning technologies. According to the websites of these institutes, APU has been offering IT courses since 1993 (APIIT Education Group 2018) compared to MNU (The Maldives National University, 2018) and Villa College (Villa College, 2018), which had a more recent history of conducting IT courses that employ sessional lecturers with limited resources. Additionally, highlighting the qualifications held by the CS1 instructors, the

APU lecturer held a Ph.D. degree, whereas, the two lecturers from the other two institutions had master's level qualifications. This information about the lecturers was sourced from the communications that took place during the data collection process. Therefore, although the result is not fully conclusive, this result might be more than an implication of confounding variables on the dependent variable.

7.3.2. Research question 4

What are the consequences of the research for the design and delivery of CS1 instruction?

Given what has been revealed from Research Question 3, a number of inferences and recommendations can be drawn in regards to two specific areas pertaining to the performance of CS1 students: (a) selection criteria for CS courses, and (b) future CS1 design and instruction.

The results show a clear association between student competence and prior mathematical background. This result converged with previous research literature on this factor (See Haungs, Clark, Clements, & Janzen, 2012; Lambert, 2015; Rizvi, Humphries, Major, Jones, & Lauzun, 2011). Similarly, the positive link between the students' prior programming experience and the CS1 student performance in the CS literature (See Haungs et al., 2012; Lambert, 2015; Rizvi et al., 2011) has been further substantiated by this study. Therefore, considering non-credit CS1 preparatory courses – CS0 and Math0 – may help up-skill students with the fundamentals essential for learning CS1 such as abstract thinking and problem-solving. In particular, for the students without prior programming experience, a CS0 course could help boost their self-confidence when they study alongside students of the CS1 course who have programming experience. Some studies suggest CS1 students have a self-perception that prior programming experience is a factor that helps to succeed in CS1, in addition to improving the perception of self and of their peers (Hagan & Markham, 2000; Tafliovich, Campbell, & Petersen, 2013). Conjointly, it may also help improve student enrolment and the retention rate of CS courses in general. However, caution must be exercised as the results of this study simply reveals an association and not a causation; therefore, further research is recommended to examine the impact of these two factors on CS1 student competence in a causal model.

Some studies have espoused that the students enrolling into CS1 with science backgrounds (Byrne & Lyons, 2001; Rountree et al., 2005), such as those who have studied physics or chemistry at high school, perform better than those who come from a humanities background. However, there is little empirical evidence to support this premise. The

hypothetical link between the science stream subjects and competence was not supported in this study. Therefore, the arguments in favor of including science subjects such as physics and chemistry (See Byrne & Lyons, 2001) in the selection criteria may further exacerbate an already declining number of enrolments to CS programs. However, taking into account the link between a mathematics background and CS1 student competence as conceded by this study and several studies of the past, students who wish to pursue CS degrees may need to consider choosing an appropriate mathematics course alongside their other stream subjects. Finally, given the limited research on this aspect and the common typecast link between science backgrounds and CS1 success, this is a factor that entails more documented empirical evidence through future research.

This study revealed a lack of association between high school CS and CS1 student competence. However, a lack of an association in this study does not rule out its significance to CS1 student competence. Some of the high school CS courses such as CS Advanced Placement exams are parallel to first-year university introductory computer programming courses, which normally carry credit towards first year CS1 study. This study's findings may be an indication that the choice of CS courses offered at the high schools of the participants is not well aligned with the needs of a typical CS1 course. Therefore, offering a high school CS course that is more goal-congruent and computer programming focused course such as High School AP Computer Science 1 might help prepare the students for CS1 study. Additionally, such a course would help students evaluate themselves and make an informed decision as to whether they want to pursue a CS degree before committing to CS programs.

Equivocal arguments and empirical studies of the positive and negative effects of using Python as a language of CS1 instruction were found in the CS1 literature. Consistent with many studies (See Alzahrani et al., 2018; Enbody & Punch, 2010; Enbody et al., 2009; Ivanović, Budimac, Radovanović, & Savić, 2015; McPheron et al., 2015; Watson & Li, 2014), this investigation also suggests that there is no performance difference between the students who were instructed with Python, Java or C++ assuming that differences in programming language were not masked by instate factors. Therefore, based on the CS1 literature, and what this study has revealed, it can be concluded that commonly used programming languages such as Java and C/C++ are also likely good candidates like Python to introduce the foundational concepts of computer programming.

However, a noteworthy point here is that the instructors need to consider several other criteria to determine the language that is most effective for CS1 instruction. For example,

choosing a programming language that is more structured and statically-typed languages such as Java or C/C++ may promote overall understanding of the programming concepts than those languages that are less structured and dynamically-typed languages such as Python or Visual Basic (Kunkle & Allen, 2016). This is because statically typed languages require data types of the variables to be defined before using them and perform type checking at program compilation time. These stricter rules help students understand common programming errors (Alzahrani et al., 2018). On the other hand, dynamically typed languages do not require the variable type to be defined which is only checked in the program runtime. Therefore, there is a possibility that Python's dynamic typing leaves students lacking practical exposure to many concepts associated with variable type. Furthermore, while programming languages like Python makes programming easier, it hides away many essential concepts related to fundamental programming constructs, potentially leaving students with fragile knowledge of essential concepts (Alzahrani et al., 2018). Similarly, other criteria such as industry relevance, ease of use and usefulness to more advanced CS courses, are among other factors instructors may consider.

7.4. Summary and Conclusion

The main conclusion of this study is that the evidence provided in the first iteration of an investigation into the development of a widely applicable measure of CS1 student competence suggests that the construct of CS1 student competence is measurable and quantifiable. Furthermore, the findings support the use of CS1 measure for individual use within classrooms and research. As outlined in the problem statement and first two chapters, CS1 educators and researchers mainly depend on a summed raw score of university exam scores and other classroom-based evaluative tools as accurate measures of student competence. This study illustrated the flaws of such evaluative tools and demonstrated an alternative method for measurement construction by a conjoint application of contemporary measurement models and an established validity framework to construct a linear measure of CS1 student competence. This allowed a stable reference frame for comparison of students and linking student scores to tasks so that a substantive meaning can be given to scores in terms of underlying proficiencies. Similarly, the linear measure allowed for drawing inferences about a hypothesised developmental model for learning to computer program. This information can inform the development of several aspects of instructional practice.

As re-iterated, the main threat to conducting CS1 research is the paucity of instruments developed upon stringent measurement theories. The logit scores of student competence are

interval-level scores generated by Rasch scaling on which parametric analysis can be performed without having to assume linearity, as in the case of CTT-based scores. This means the study has addressed a gap identified within the CS1 literature and as claimed in the problem statement at the beginning of the thesis. The Phase three – correlational study – is a simple demonstration of the value of this measure to CS1 researchers seeking to understand the factors determining CS1 student competence. Therefore, with this measure, CS1 research can be taken further such as to understand the factors related to student competence in causal and regressions models, which can be used to inform curriculum design and selection criteria to CS degree programs.

Lastly, as this is the first investigation of this kind employing the Rasch model and Wolfe and Smith's (2007a, 2007b) framework, this study also makes an important theoretical and methodological contribution to the CS1 body of knowledge. The methodological framework guided by this study could be used as a reference framework for those who are interested in further studies of this nature. Similarly, the investigation contributed to the theoretical knowledge in understanding the embodiment concepts and relationships constituting the construct of CS1 student competence by visually illustrating and empirically testing the models. Hence, these models can serve as a conceptual framework for future studies seeking to understand the theoretical underpinnings of how novices learn to program. Finally, the recommendation, limitations and future directions presented in the following section will possibly show researchers of similar interests the possible paths this research can be taken in the future.

7.5. Limitations and Future Directions

From a generalisability perspective, the measure was shown to perform invariantly across samples from different institutes, countries, and across programming languages, in addition to demonstrating high reliability. However, this investigation is the first of its kind to measure CS1 student competence by employing the RMT, as there is limited data to support the external aspect of validity. Another limitation is prior to this study no investigation had established what defines the construct of CS1 student competence. Therefore, it is unlikely that the construct model itself is free from flaws. Thus, this could be an area of interest to focus on in future research. Psychological instrument development is not a one-off process; rather it is an iterative process and a continuing quest to improve the psychometric features of the measure. Hence, multiple future iterations are necessary to improve the psychometric

properties. In particular, examining the properties of the measure with bigger samples from a variety of countries and including more programming languages.

Similar to this was the first effort to develop a widely applicable CS1 measure employing the RMT. This was also the first investigation to evaluate psychometric properties employing a contemporary validity framework. Since validity is a continuing quest, any future studies need to concentrate on gaining additional validity evidence, particularly the generalisability aspect of validity. This is mainly because the sample of the current study is only from the Maldives and Malaysia. Such studies should focus on examining the stability (i.e., generalisability) of the item difficulty estimates across other institutes with varying instructional languages other than the current three languages. Item bias should also be conducted on important demographic groups when sufficient sample sizes exist. For example, item bias by gender was not examined in this study due to the limited number of female participants available.

The findings of this study have revealed critical information pertaining to student and learning environment factors, which has important implications for selection, teaching and supporting students through a CS1 course. The scores of the dependent variable were generated by a measure developed by the researcher as part of the construction and testing of an instrument for validity employing RMT. Although authoritative work on Rasch modelling has previously been conducted on smaller samples similar to the current study, it may have some validity issues and significant bias. Thus, the CS1 measure needs to be tested in a broader sample to help improve its generalisability further. Furthermore, as the students were from three different institutes, the correlation study results could have been confounded by various institutional factors to some extent.

However, to some extent, the findings of this study have answered some of the intriguing questions surrounding association between some of the student and learning environment factors with CS1 student competence. The strength of this study is the validity and measurement level of the scores (interval-level) used for the dependent variable (CS1 student competence). In the past, performance predictor studies were conducted employing raw-scores that are not suitable for parametric analysis. Therefore, some of the variables that are shown to have a statistical significance with student competence might also be causal factors, which need further consideration for testing. As this is beyond the scope of this investigation, future research could be directed at investigating the role of these variables in causal models and regression models employing interval-level data in multi-institutional,

multi-national studies. Similarly, another area of interest for future research could be testing the hypothetical relationship between the three programming skills – tracing, reading, and writing – advanced in the construct theory of this investigation via a Structural Equation Modelling (SEM) or a similar technique.

Furthermore, future researchers can explore several other compelling factors explored in CS education research in relation to student performance such as the levels of Piaget's stages of development, cognitive learning style, and many other latent variables cited as determinants of student competence. However, as an antagonist of using summed scores for any measurement purposes, the researcher would like to emphasise that the true benefit of this instrument would be realised in such studies given the other latent variables are also measured at interval-level.

References

Abedalaziz, N., & Leng, C. H. (2013). The Relationship between CTT and IRT Approaches in Analyzing Item Characteristics. Malaysian Online Journal of Educational Sciences, 1(1), 64-70. Retrieved from https://files.eric.ed.gov/fulltext/EJ1086220.pdf

Adams, R. J., Wu, M. L., & Wilson, M. (2012). The Rasch rating model and the disordered threshold controversy. Educational and Psychological Measurement, 72(4), 547-573. https://doi.org/10.1177/0013164411432166

Agarwal, K. K., & Agarwal, A. (2005). Python for CS1, CS2 and beyond. Journal of Computing Sciences in Colleges, 20(4), 262-270. Retrieved from https://dl.acm.org/citation.cfm?id=1047887

Agarwal, K. K., Agarwal, A., & Celebi, M. E. (2008). Python puts a squeeze on Java for CS0 and beyond. Journal of Computing Sciences in Colleges, 23(6), 49-57. Retrieved from https://dl.acm.org/citation.cfm?id=1352393

Al-Rukban, M. O. (2006). Guidelines for the construction of multiple choice questions tests. Journal of Family & Community Medicine, 13(3), 125. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3410060/

Alagumalai, S., & Curtis, D. D. (2005). Classical Test Theory. In S. Alagumalai, D. D. Curtis, & N. Hungi (Eds.), *Applied Rasch measurement: A book of exemplars*: Springer. https://doi-org.dbgw.lis.curtin.edu.au/10.1007/1-4020-3076-2_1

Alaoutinen, S., & Smolander, K. (2010). *Student self-assessment in a programming course using bloom's revised taxonomy.* In Proceedings of the 15th annual conference on Innovation and technology in computer science education (pp. 155-159), *ACM*. https://doi.org/10.1145/1822090.1822135

Alasuutari, P., Bickman, L., & Brannen, J. (Eds.). (2008). *The SAGE handbook of social research methods*: Thousand Oaks, CA: Sage.

Albano, A. D. (2017). Introduction to educational and psychological measurement using R. Retrieved from https://cehs01.unl.edu/aalbano/intro-measurement-r/intro-measurement.pdf

Alemán, J. L. F. (2011). Automated assessment in a programming tools course. IEEE Transactions on Education, 54(4), 576-581. https://doi.org/10.1109/te.2010.2098442

Allison, P. D. (2001). Missing data: Sage University papers series on quantitative applications in the social sciences (07–136). Thousand Oaks, CA.

Allison, P. D. (2012). *Handling missing data by maximum likelihood.* In SAS global forum, *Statistical Horizons, Havenford, PA.*

Alvarado, C., Lee, C. B., & Gillespie, G. (2014). *New CS1 pedagogies and curriculum, the same success factors?* In Proceedings of the 45th ACM technical symposium on Computer science education, *ACM*. https://doi.org/10.1145/2538862.2538897

Alzahrani, N., Vahid, F., Edgcomb, A., Nguyen, K., & Lysecky, R. (2018). *Python Versus C++: An Analysis of Student Struggle on Small Coding Exercises in Introductory Programming Courses.* In Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland, USA, *ACM*. https://doi.org/10.1145/3159450.3160586

American Psychological Association, American Educational Research Association, National Council on Measurement in Education, American Educational Research Association, & Committee on Test Standards. (1966). *Standards for educational and psychological tests and manuals*. American Psychological Association.

American Psychological Association, American Educational Research Association, National Council on Measurement in Education, American Educational Research Association, & Committee on Test Standards. (1999). *Standards for educational and psychological testing*. American Psychological Association.

American Psychological Association, American Educational Research Association, National Council on Measurement in Education, American Educational Research Association, & Committee on Test Standards. (2014). *Standards for educational and psychological testing*. American Psychological Association.

Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). *A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objective*. Longman, New York, 2001.

Andrich, D. (1978). A rating formulation for ordered response categories. Psychometrika, 43(4), 561-573. https://doi.org/10.1007/bf02293814

Andrich, D. (1988). Rasch Models for Measurement: Sage Publications, Newbury Park, CA.

Andrich, D. (1989). Distinctions between assumptions and requirements in measurement in the social sciences. Mathematical and Theoretical Systems, 4, 7-16.

Andrich, D. (2013). An expanded derivation of the threshold structure of the polytomous Rasch model that dispels any "threshold disorder controversy". Educational and Psychological Measurement, 73(1), 78-124. https://doi.org/10.1177/0013164412450877

Andrich, D., Sheridan, B., & Luo, G. (2011). RUMM2030 software and manuals. Perth, Australia: University of Western Australia. Retrieved from http://www.rummlab.com.au/

Andrich, D., Sherridan, B., & Luo, G. (2010). RUMM2030 (Version 5.3): RUMM Laboratory Pty Ltd. Retrieved from http://www.rummlab.com.au/

Antony, M. M., & Barlow, D. H. (2002). *Handbook of assessment and treatment planning for psychological disorders*: Guilford press.

AP Central. (2018). AP Computer Science A. Retrieved from https://apcentral.collegeboard.org/courses/ap-computer-science-a?course=ap-computer-science-a

APIIT Education Group (2018). Asia Pacific University of Technology & Innovation (APU). Retrieved from http://www.apu.edu.my/explore-apu/apiit-education-group

Atlas, G. D., Taggart, T., & Goodell, D. J. (2004). The effects of sensitivity to criticism on motivation and performance in music students. British Journal of Music Education, 21(1), 81-87. https://doi.org/10.1017/s0265051703005540

Badia, X., Prieto, L., & Linacre, J. M. (2002). Differential item and test functioning (DIF & DTF). Rasch Measurement Transactions, 16(3), 889. Retrieved from https://www.rasch.org/rmt/rmt163g.htm

Bailie, F., Courtney, M., Murray, K., Schiaffino, R., & Tuohy, S. (2003). Objects First-does it work? Journal of Computing Sciences in Colleges, 19(2), 303-305. Retrieved from https://dl-acm-org.dbgw.lis.curtin.edu.au/

Baker, F. B. (2001). *The basics of item response theory* (2 ed.): ERIC clearinghouse on assessment and evaluation.

Bandura, A. (1986). Social foundation of thought and actions: A cognitive social theory. Prentice Hall, Englewood Cliffs, New York.

Barker, R. J., & Unger, E. (1983). *A predictor for success in an introductory programming class based upon abstract reasoning development.* In ACM SIGCSE Bulletin, *ACM*. https://doi.org/10.1145/800038.801037

Becker, W. E., & Johnston, C. (1999). The relationship between multiple choice and essay response questions in assessing economics understanding. Economic Record, 75(4), 348-357. https://doi.org/10.1111/j.1475-4932.1999.tb02571.x

Bennedsen, J., & Caspersen, M. E. (2005). *An investigation of potential success factors for an introductory model-driven programming course.* In Proceedings of the first

international workshop on Computing education research (pp. 155-163), Seattle, WA, USA, *ACM*. http://dx.doi.org.dbgw.lis.curtin.edu.au/10.1145/1272848.1272879

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. ACM SIGCSE Bulletin, 39(2), 32-36. http://dx.doi.org.dbgw.lis.curtin.edu.au/10.1145/1272848.1272879

Bergin, S., & Reilly, R. (2006). Predicting introductory programming performance: A multi-institutional multivariate study. Computer Science Education, 16(4), 303-323. https://doi.org/10.1080/08993400600997096

Biggs, J. B., & Collis, K. F. (1982). *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. New York: Academic Press.

Biggs, J. B., & Tang, C. (2011). *Teaching for quality learning at university: What the student does* (4 ed.). Glasgow, UK: Bell and Bain Ltd

Black, P., & Wiliam, D. (1998). Assessment and classroom learning. Assessment in Education: Principles, Policy & Practice, 5(1), 7-74.

Bland, J. M., & Altman, D. G. (1995). Multiple significance tests: the Bonferroni method. Bmj, 310(6973), 170. https://doi.org/10.1136/bmj.310.6973.170

Bloom, B. S. (1956). *Taxonomy of educational objectives, handbook I: The cognitive domain* (Vol. 19): Addison Wesley.

Bode, R. K., & Wright, B. D. (1999). Rasch measurement in higher education. *Higher education: Handbook of theory and research* (pp. 287-316): Springer. https://doi.org/10.1007/978-94-011-3955-7_7

Bond, T. (2003). Validity and assessment: A Rasch measurement perspective. Metodología de las Ciencias del Comportamiento, 5(2), 179-194. Retrieved from https://researchonline.jcu.edu.au/1799/

Bond, T., & Fox, C. M. (2015). *Applying the Rasch model: Fundamental measurement in the human sciences*. New York: Routledge.

Bond, T. G., & Fox, C. M. (2013). *Applying the Rasch model: Fundamental measurement in the human sciences*: Psychology Press.

Boone, W. J. (2016). Rasch analysis for instrument development: Why, when, and how? CBE-Life Sciences Education, 15(4), rm4. https://doi.org/10.1187/cbe.16-04-0148

Boone, W. J., & Scantlebury, K. (2006). The role of Rasch analysis when conducting science education research utilizing multiple-choice tests. Science Education, 90(2), 253-269. http://dx.doi.org/10.1002/sce.20106

Borgatta, E. F., & Bohrnstedt, G. W. (1980). Level of measurement: Once over again. Sociological Methods & Research, 9(2), 147-160. https://doi.org/10.1177/004912418000900202

Bornat, R., & Dehnadi, S. (2008). *Mental models, consistency and programming aptitude.* In Proceedings of the tenth conference on Australasian computing education, *Australian Computer Society, Inc.* Retrieved from https://dl-acm-org.dbgw.lis.curtin.edu.au/

Borsboom, D., Mellenbergh, G. J., & van Heerden, J. (2004). The concept of validity. Psychological review, 111(4), 1061.

Boulton-Lewis, G. (1994). Tertiary students' knowledge of their own learning and a SOLO taxonomy. Higher Education, 28(3), 387-402. https://doi.org/10.1007/bf01383724

Boyle, R., Carter, J., & Clark, M. (2002). What makes them succeed? Entry, progression and graduation in Computer Science. Journal of Further and Higher Education, 26(1), 3-18. https://doi.org/10.1080/03098770120108266

Brosnan, M. J. (1998). The impact of computer anxiety and self-efficacy upon performance. Journal of Computer Assisted Learning, 14(3), 223-234. http://dx.doi.org/10.1046/j.1365-2729.1998.143059.x

Brown, T. (2010). Construct validity: A unitary concept for occupational therapy assessment and measurement. Hong Kong Journal of Occupational Therapy, 20(1), 30-42. https://doi.org/10.1016/S1569-1861(10)70056-5

Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. ACM SIGCSE Bulletin, 33(3), 49-52. https://doi.org/10.1145/507758.377467

Caceffo, R., Wolfman, S., Booth, K. S., & Azevedo, R. (2016). *Developing a computer science concept inventory for introductory programming.* In Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, Tennessee, USA, *ACM*. https://doi.org/10.1145/2839509.2844559

Cambridge International Examinations. (2018). Cambridge International AS and A Level Computer Science (9608). . Retrieved from http://www.cambridgeinternational.org/images/202629-2017-2019-syllabus.pdf

Cano, S. J., Mayhew, A., Glanzman, A. M., Krosschell, K. J., Swoboda, K. J., Main, M., . . . Payan, C. A. (2014). Rasch analysis of clinical outcome measures in spinal muscular atrophy. Muscle & Nerve, 49(3), 422-430. http://dx.doi.org/10.1002/mus.23937

Cantrell, C. E. (1997). Item Response Theory: Understanding the One-Parameter Rasch model. In B. Thompson (Ed.), *The Annual Meeting of the Southwest Educational*

Research Association (Vol. 5, pp. 171-192). Austin, TX: Stamford, CT: JAI Press. Retrieved from https://eric.ed.gov/?id=ED415281

Cappelleri, J. C., Lundy, J. J., & Hays, R. D. (2014). Overview of Classical Test Theory and Item Response Theory for the quantitative assessment of items in developing patient-reported outcomes measures. Clinical Therapeutics, 36(5), 648-662. https://doi.org/10.1016/j.clinthera.2014.04.006

Cardell-Oliver, R. (2014). How students experience learning to program. Education Research and Perspectives, 41, 196-216. Retrieved from https://search.informit.com.au/documentSummary;dn=135981868467940;res=IELAPA

Case, S. M., & Swanson, D. B. (1998). *Constructing written test questions for the basic and clinical sciences* Retrieved from http://www.au.af.mil/au/awc/awcgate/documents/nbme_iwgindex.pdf

Cavanagh, R. F. (2007). Measurement issues in the use of rating scale instruments in learning environment research. Australian Association For Research In Education Research and Perspectives, 15(6), 1-9. Retrieved from http://hdl.handle.net/20.500.11937/7965

Cavanagh, R. F. (2009). Applying a unified theory of validity: Identifying evidence for validity arguments in an investigation of student engagement in classroom learning. Learning Environments Research, 18(3), 349-361. https://doi.org/10.1007/s10984-015-9188-z

Cavanagh, R. F., & Koehler, M. J. (2013). A turn toward specifying validity criteria in the measurement of technological pedagogical content knowledge (TPACK). Journal of Research on Technology in Education, 46(2), 129-148. https://doi.org/10.1080/15391523.2013.10782616

Cavanagh, R. F., & Romanoski, J. T. (2008). Sequential application of Rasch analysis and structural equation modeling to investigate elementary school classroom learning culture. In R. Waugh (Ed.), *Frontiers in Educational Psychology* (pp. 67-87). New York: Nova Science Publishers.

Cavanagh, R. F., Waldrip, B., Romanoski, J., Dorman, J., & Fisher, D. (2005). *Measuring student perceptions of classroom assessment.* In International Education Research Conference-Creative Dissent:t: Constructive Solutions Parramatta, Australia, *AARE Inc*. Retrieved from https://espace.curtin.edu.au/bitstream/handle/20.500.11937/11434/156244_156244.pdf?sequence=2

Chang, C. H. (1996). Finding two dimensions in MMPI-2 depression. Structural Equation Modeling: A Multidisciplinary Journal, 3(1), 41-49. https://doi.org/10.1080/10705519609540028

Chao, S., & Henderson, M. (2012). *Gendered differences in the participation of Australian tertiary computer science: Implications for schools.* In Proceedings of Australian Computers in Education Conference, Perth, Australia, *Australian Council for Computers in Education (ACCE)*. Retrieved from https://pdfs.semanticscholar.org/e23b/8a2aff85646373a3899fb008f11d5ce809c5.pdf

Chen, W. H., Lenderking, W., Jin, Y., Wyrwich, K. W., Gelhorn, H., & Revicki, D. A. (2014). Is Rasch model analysis applicable in small sample size pilot studies for assessing item characteristics? An example using PROMIS pain behavior item bank data. Quality of Life Research, 23(2), 485-493. http://dx.doi.org.dbgw.lis.curtin.edu.au/10.1007/s11136-013-0487-5

Chen, W. H., & Thissen, D. (1997). Local dependence indexes for item pairs using item response theory. Journal of Educational and Behavioral Statistics, 22(3), 265-289. https://doi.org/10.3102/10769986022003265

Chen, X. (2013). *STEM Attrition: College Students' Paths into and out of STEM Fields. Statistical Analysis Report. NCES 2014-001*. Retrieved from https://nces.ed.gov/pubs2014/2014001rev.pdf

Chen, Z., & Marx, D. (2005). Experiences with Eclipse IDE in programming courses. Journal of Computing Sciences in Colleges, 21(2), 104-112. Retrieved from https://dl.acm.org/citation.cfm?id=1089068

Chien, T. W., Hsu, S. Y., Tai, C., Guo, H. R., & Su, S. B. (2008). Using Rasch analysis to validate the revised PSQI to assess sleep disorders in Taiwan's hi-tech workers. Community Mental Health Journal, 44(6), 417-425. http://dx.doi.org.dbgw.lis.curtin.edu.au/10.1007/s10597-008-9144-9

Chilisa, B., & Kawulich, B. (2012). *Selecting a research approach: paradigm, methodology, and methods*. London: McGraw-Hill Education.

Chinn, D., de Raadt, M., Philpott, A., Sheard, J., Laakso, M.-J., D'Souza, D., . . . Lister, R. (2012). *Introductory programming: Examining the exams.* In Proceedings of the 14th Australasian Computing Education Conference-Volume 123 (pp. 61-70), Melbourne, Australia, *Australian Computer Society, Inc.* Retrieved from https://search-proquest-com.dbgw.lis.curtin.edu.au/docview/1313030622?accountid=10382

Christensen, K. B., Makransky, G., & Horton, M. (2017). Critical values for Yen's Q 3: Identification of local dependence in the Rasch model using residual correlations. Applied Psychological Measurement, 41(3), 178-194. https://doi.org/10.1177/0146621616677520

Chudowsky, N., Glaser, R., & Pellegrino, J. W. (2001). *Knowing what students know: The science and design of educational assessment*. Washington, DC: National Academy Press.

Clear, T., Whalley, J., Lister, R., Carbone, A., Hu, M., Sheard, J., . . . Thompson, E. (2008). *Reliably classifying novice programmer exam responses using the SOLO taxonomy.* In 21st Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2008), Auckland, New Zealand. Retrieved from http://hdl.handle.net/10453/12626

Close, R., Kopec, D., & Aman, J. (2000). *CS1: Perspectives on programming languages and the breadth-first approach.* In Journal of Computing Sciences in Colleges, *Consortium for Computing Sciences in Colleges*. Retrieved from http://spider.sci.brooklyn.cuny.edu/~kopec/Publications/Publications/R_1_E.pdf

Collis, K. F., & Biggs, J. (1982). *Evaluating the quality of learning: The SOLO Taxonomy*. New York: Academic Press.

Collis, K. F., Romberg, T. A., & Jurdak, M. E. (1986). A technique for assessing mathematical problem-solving ability. Journal for Research in Mathematics Education, 17(3), 206-221. http://dx.doi.org/10.2307/749302

Computing Research Association. (2017). *Generation CS: Computer Science Undergraduate Enrollments Surge Since 2006*. Retrieved from https://cra.org/wp-content/uploads/2017/02/Generation-CS.pdf

Cook, D. A., & Hatala, R. (2016). Validation of educational assessments: A primer for simulation and beyond. Advances in Simulation, 1(1), 31. https://doi.org/10.1186/s41077-016-0033-y

Coombs, C. H., Dawes, R. M., & Tversky, A. (1970). *Mathematical psychology: An elementary introduction*: Englewood Cliffs, N.J. : Prentice-Hall.

Corney, M., Fitzgerald, S., Hanks, B., Lister, R., McCauley, R., & Murphy, L. (2014). *'Explain in plain English' questions revisited: Data structures problems.* In Proceedings of the 45th ACM technical symposium on Computer science education, Atlanta, Georgia, USA, *ACM*. http://dx.doi.org/10.1145/2538862.2538911

Creswell, J. W. (2012). *Educational Research Palnning, Conducting, and Evaluating Quatitative and Qualitative Research* (4 ed.). Boston Pearson Education, Inc.

Cronbach, L. J. (1980). Validity on parole: How can we go straight? New directions for testing and measurement: Measuring achievement, progress over a decade, 5, 99-108. http://dx.doi.org/10.1037/h0040957

Cronbach, L. J. (1988). Five perspectives on validity argument. In L. J. Cronbach, H. Wainer, & H. Braun (Eds.), *Test validity* (pp. 3-17): Hillsdale, NJ: Erlbaum.

Cronbach, L. J., & Meehl, P. E. (1955). Construct Validity in Psychological Tests. Psychological Bulletin, 52(4), 281-302. http://dx.doi.org/10.1037/h0040957

Davis, L. L. (1992). Instrument review: Getting the most from a panel of experts. Applied Nursing Research, 5(4), 194-197. https://doi.org/10.1016/s0897-1897(05)80008-4

De Ayala, R. J. (2003). The effect of missing data on estimating a respondent's location using ratings data. Journal of Applied Measurement, 4, 1-9. Retrieved from https://eric.ed.gov/?id=ED468463

De Champlain, A. F. (2010). A primer on classical test theory and item response theory for assessments in medical education. Medical Education, 44(1), 109-117. http://dx.doi.org.dbgw.lis.curtin.edu.au/10.1111/j.1365-2923.2009.03425.x

De Jong, G. J., & Kamphuls, F. (1985). The development of a Rasch-type loneliness scale. Applied Psychological Measurement, 9(3), 289-299. Retrieved from http://journals.sagepub.com/doi/10.1177/014662168500900307

Decker, A. (2007). *How students measure up: An assessment instrument for introductory computer science.* (Doctoral Thesis), State University of New York at Buffalo. Retrieved from https://ubir.buffalo.edu/xmlui/bitstream/handle/10477/34582/2007-06.pdf?sequence=2

Deitel, P., & Deitel, H. (2010). *C++ How to Program 7th Edition*: Prentice Hall.

des Rivières, J., & Wiegand, J. (2004). Eclipse: A platform for integrating development tools. IBM Systems Journal, 43(2), 371-383. https://doi.org/10.1147/sj.432.0371

Dillon, E., Anderson, H., & Brown, M. (2012). *Studying the novice's perception of visual vs. Command line programming tools in CS1*. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, *SAGE Publications Sage CA: Los Angeles, CA*. https://doi.org/10.1177/1071181312561126

Dimitrov, D. M. (2014). *Statistical methods for validation of assessment scale data in counseling and related fields*. Alexandria, VA: American Counseling Association.

Dong, Y., & Peng, C.-Y. J. (2013). Principled missing data methods for researchers. SpringerPlus, 2(1), 222. https://doi.org/10.1186/2193-1801-2-222

Duckor, B. M., Draney, K., & Wilson, M. (2009). Measuring measuring: Toward a theory of proficiency with the Constructing Measures framework. Journal of Applied Measurement, 10(3), 296. Retrieved from http://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1007&context=second_ed_pub

Eachus, P., & Cassidy, S. (1997). *Self-efficacy, locus of control and styles of learning as contributing factors in the academic performance of student health professionals.* In Proceedings of First Regional Congress of Psychology for Professionals in the Americas - Interfacing the Science and Practice of Psychology, Mexico City.

Ebrahimi, A. (2012). How does early feedback in an online programming course change problem solving? Journal of Educational Technology Systems, 40(4), 371-379. https://doi.org/10.2190/et.40.4.c

Embretson, S. E. (1996a). Item Response Theory Models and spurious interaction effects in factorial ANOVA designs. Applied Psychological Measurement, 20(3), 201-212. https://doi.org/10.1177/014662169602000302

Embretson, S. E. (1996b). The new rules of measurement. Psychological Assessment, 8(4), 341. http://dx.doi.org/10.1037/1040-3590.8.4.341

Embretson, S. E. (Ed.) (1993). *Psychometric models for learning and cognitive processes*: Mahwah, NJ: Erlbaum.

Embretson, S. E., & Hershberger, S. L. (1999). *The new rules of measurement: What every psychologist and educator should know*. New York: Psychology Press.

Embretson, S. E., & Reise, S. P. (2000). Item response theory for psychologists: Mahwah, NJ: Lawrence Erlbaum Associates.

Enbody, R. J., & Punch, W. F. (2010). *Performance of python CS1 students in mid-level non-python CS courses.* In Proceedings of the 41st ACM technical symposium on Computer science education, *ACM*. http://dx.doi.org/10.1145/1734263.1734437

Enbody, R. J., Punch, W. F., & McCullen, M. (2009). Python CS1 as preparation for C++ CS2. ACM SIGCSE Bulletin, 41(1), 116-120. http://dx.doi.org/10.1145/1539024.1508907

Erdogan, B. D., Elhan, A. H., Demirtas, H., Öztuna, D., Küçükdeveci, A. A., & Kutlay, S. (2013). Multiple Imputation of Missing Values Using the Response Function Method

Based on a Data Set of the Health Assessment Questionnaire Disability Index. Turkish Journal of Rheumatology, 28(1), 2-9. https://doi.org/10.5606/tjr.2013.001

Erguven, M. (2013). Two approaches to psychometric process: Classical Test Theory and Item Response Theory. Journal of Education, 2(2), 23-30.

Evans, G. E., & Simkin, M. G. (1989). What best predicts computer proficiency? Communications of the ACM, 32(11), 1322-1327. https://doi.org/10.1145/68814.68817

Fairbrother, R. (1975). The reliability of teachers' judgment of the abilities being tested by multiple choice items. Educational Research, 17(3), 202-210. https://doi.org/10.1080/0013188750170306

Fan, X. (1998). Item Response Theory and Classical Test Theory: An empirical comparison of their item/person statistics. Educational and Psychological Measurement, 58(3), 357-381. https://doi.org/10.1177/0013164498058003001

Farag, W., Ali, S., & Deb, D. (2013). *Does language choice influence the effectiveness of online introductory programming courses?* Paper presented at the Proceedings of the 14th annual ACM SIGITE conference on Information technology education, Orlando, Florida, USA. http://dx.doi.org/10.1145/2512276.2512293

Feldt, L. S., Steffen, M., & Gupta, N. C. (1985). A comparison of five methods for estimating the standard error of measurement at specific score levels. Applied Psychological Measurement, 9(4), 351-361. https://doi.org/10.1177/014662168500900402

Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., de Raadt, M., . . . Lister, R. (2006). *Predictors of success in a first programming course.* In Proceedings of the 8th Australasian Conference on Computing Education-Volume 52 (pp. 189-196), Hobart, Australia, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=1151894

Fisher, W. (1993). Robustness and invariance. Rasch Measurement Transactions, 7, 295. Retrieved from https://www.rasch.org/rmt/rmt72m.htm

Fisher, W. P. (1992). Reliability statistics. Rasch Measurement Transactions, 6(3), 238. Retrieved from https://www.rasch.org/rmt/rmt63i.htm

Forrest, M., & Andersen, B. R. (1986). Ordinal scale and statistics in medical research. Br Med J (Clin Res Ed), 292(6519), 537-538. https://doi.org/10.1136/bmj.292.6519.537

Fraenkel, J. R., & Wallen, N. E. (2003). *How to design and evaluate research in education* (5 ed.): McGraw-Hill Higher Education.

Franchignoni, F., Giordano, A., Marcantonio, L., Alberto Coccetta, C., & Ferriero, G. (2012). Current issues in psychometric assessment of outcome measures. Medicina Fluminensis, 48(4), 463-470. Retrieved from https://hrcak.srce.hr/95732

Fuller, U., Johnson, C. G., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jackova, J., . . . Riedesel, C. (2007). Developing a computer science-specific learning taxonomy. ACM SIGCSE Bulletin, 39(4), 152-170. http://dx.doi.org/10.1145/1345443.1345438

Furlow, C., Fouladi, R., Gagne, P., & Whittaker, T. (2007). A Monte Carlo study of the impact of missing data and differential item functioning on theta estimates from two polytomous Rasch family models. Journal of Applied Measurement, 8(4), 388-403. Retrieved from http://europepmc.org/abstract/MED/18250525

Furr, R., & Bacharach, V. (2013). *Psychometrics. An Introduction*. Thousand Oaks, CA: SAGE Publications.

Furr, R. M., & Bacharach, V. R. (2008). *Psychometrics: An introduction* (2 ed.): SAGE Publications.

Gaito, J. (1980). Measurement scales and statistics: Resurgence of an old misconception. 87(3), 564-567. http://dx.doi.org/10.1037/0033-2909.87.3.564

Ganglmair, A., & Lawson, R. (2003). Advantages of Rasch modelling for the development of a scale to measure affective response to consumption. European Advances in Consumer Research, 6, 162-167. Retrieved from http://acrwebsite.org/volumes/11738/volumes/e06/E-06

Ginat, D., & Menashe, E. (2015). *SOLO Taxonomy for assessing novices' algorithmic design.* In Proceedings of the 46th ACM Technical Symposium on Computer Science Education Kansas City, Missouri, USA, *ACM*. http://dx.doi.org/10.1145/2676723.2677311

Gluga, R., Kay, J., Lister, R., Kleitman, S., & Lever, T. (2012a). *Coming to terms with Bloom: An online tutorial for teachers of programming fundamentals.* In Proceedings of the Fourteenth Australasian Computing Education Conference-Volume 123 (pp. 147-156), Melbourne, Australia, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=2483734

Gluga, R., Kay, J., Lister, R., Kleitman, S., & Lever, T. (2012b). *Over-confidence and confusion in using bloom for programming fundamentals assessment*. Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education, Raleigh, North Carolina, USA. https://doi.org/10.1145/2157136.2157181

Goldstein, H. (2015). Validity, science and educational measurement. Assesment in Education, Policy & Practice, 22(2), 193-201.

Gomes, A., Carmo, L., Bigotte, E., & Mendes, A. (2006). *Mathematics and programming problem solving.* In Proceedings of the third E-Learning Conference–Computer Science Education (pp. 1-5), Coimbra, Portugal. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.532.7543&rep=rep1&type=pdf

Goodwin, L. D., & Leech, N. L. (2003). The meaning of validity in the new standards for educational and psychological testing: Implications for measurement courses. Measurement and Evaluation in Counseling and Development, 36(3), 181-192. https://doi.org/10.1080/07481756.2003.11909741

Goold, A., & Rimmer, R. (2000). Factors affecting performance in first-year computing. ACM SIGCSE Bulletin, 32(2), 39-43. https://doi.org/10.1145/355354.355369

Graham, J. W. (2009). Missing data analysis: Making it work in the real world. Annual Review of Psychology, 60, 549-576. http://dx.doi.org.dbgw.lis.curtin.edu.au/10.1146/annurev.psych.58.110405.085530

Grant, J. S., & Davis, L. L. (1997). Selection and use of content experts for instrument development. Research in Nursing & Health, 20(3), 269-274. https://doi.org/10.1002/(sici)1098-240x(199706)20:3<269::aid-nur9>3.3.co;2-3

Guilford, J. P. (1946). New standards for test evaluation. Educational and Psychological Measurement, 6(4), 427-438. https://doi.org/10.1177/001316444600600401

Guttman, L. (Ed.) (1950). *The basis for scalogram analysis*. Princeton, N.J.: Princeton University Press.: Bobbs-Merrill, College Division.

Hadjerrouit, S. (1998). Java as first programming language: A critical evaluation. ACM SIGCSE Bulletin, 30(2), 43-47. https://doi.org/10.1145/292422.292440

Hagan, D., & Markham, S. (2000). Does it help to have some programming experience before beginning a computing degree program? ACM SIGCSE Bulletin, 32(3), 25-28. https://doi.org/10.1145/353519.343063

Hagquist, C., & Andrich, D. (2004). Is the sense of coherence-instrument applicable on adolescents? A latent trait analysis using Rasch-modelling. Personality and Individual Differences, 36(4), 955-968. https://doi.org/10.1016/S0191-8869(03)00164-8

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), 181.

Hagquist, C., Bruce, M., & Gustavsson, J. P. (2009). Using the Rasch model in nursing research: An introduction and illustrative example. International Journal of Nursing Studies, 46(3), 380-393. https://doi.org/10.1016/j.ijnurstu.2008.10.007

Haladyna, T. M., Downing, S. M., & Rodriguez, M. C. (2002). A review of multiple-choice item-writing guidelines for classroom assessment. Applied Measurement in Education, 15(3), 309-333. https://doi.org/10.1207/S15324818AME1503_5

Hambleton, R. K., & Jones, R. W. (1993). Comparison of Classical Test Theory and Item Response Theory and their applications to test development. Educational Measurement: Issues and Practice, 12(3), 38-47. http://dx.doi.org/10.1111/j.1745-3992.1993.tb00543.x

Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). Fundamentals of item response theory London: SAGE.

Harrington, B., & Cheng, N. (2018). *Tracing vs. Writing Code: Beyond the Learning Hierarchy*. Paper presented at the Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland, USA. https://doi.org/10.1145/3159450.3159530

Harwell, M. R., Gatti, G. G., & Linacre, J. M. (2002). "Linear" rescaling vs. linear measurement. Rasch Measurement Transactions, 16(3), 890-891. Retrieved from https://www.rasch.org/rmt/rmt163h.htm

Haungs, M., Clark, C., Clements, J., & Janzen, D. (2012). *Improving first-year success and retention through interest-based CS0 courses.* In Proceedings of the 43rd ACM technical symposium on Computer Science Education, Raleigh, North Carolina, USA, *ACM*. http://dx.doi.org/10.1145/2157136.2157307

Hays, R. D., Morales, L. S., & Reise, S. P. (2000). Item Response Theory and health outcomes measurement in the 21st century. Medical Care, 38(9 Suppl), 28–42. https://doi.org/10.1097/00005650-200009002-00007

Herman, N. S., Salam, S. B., & Noersasongko, E. (2011). A study of tracing and writing performance of novice students in introductory programming. Communications in Computer and Information Science, 181, 557-570. https://doi.org/10.1007/978-3-642-22203-0_48

Hertz, M. (2010). *What do CS1 and CS2 mean?: Investigating differences in the early courses.* In Proceedings of the 41st ACM technical symposium on Computer science education, NY, USA, *ACM*. http://dx.doi.org/10.1145/1734263.1734335

Hohensinn, C., & Kubinger, K. D. (2011). On the impact of missing values on item fit and the model validness of the Rasch model. Psychological Test and Assessment Modeling, 53, 380-393. Retrieved from http://p16277.typo3server.info/fileadmin/download/ptam/3-2011_20110927/07_Hohensinn.pdf

Hubbard, J. R. (1999). *Programming with Java*: McGraw-Hill Companies, Inc.

Ivanović, M., Budimac, Z., Radovanović, M., & Savić, M. (2015). *Does the choice of the first programming language influence students' grades?* In Proceedings of the 16th International Conference on Computer Systems and Technologies, *ACM*. https://doi.org/10.1145/2812428.2812448

Izu, C., Weerasinghe, A., & Pope, C. (2016). *A study of code design skills in novice programmers using the SOLO taxonomy.* In Proceedings of the 2016 ACM Conference on International Computing Education Research, Melbourne, VIC, Australia, *ACM*. http://dx.doi.org/10.1145/2960310.2960324

Jerkins, J. A., Stenger, C. L., Stovall, J., & Jenkins, J. T. (2013). Establishing the impact of a computer science/mathematics anti-symbiotic stereotype in CS students. Journal of Computing Sciences in Colleges, 28(5), 47-53. Retrieved from https://dl.acm.org/citation.cfm?id=2458578

Johnson, C. G., & Fuller, U. (2006). *Is Bloom's taxonomy appropriate for computer science?* In Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006, *ACM*. https://doi.org/10.1145/1315803.1315825

Johnson, R. (2012). *An introduction to java programming and object-oriented application development*: Cengage Learning.

Jones, L. V. (1971). The nature of educational measurement. In R. L. Thorndike (Ed.), (2 ed., pp. 485-498). Washington, DC: American Council on Education.

Kane, M. T. (2001). Current concerns in validity theory. Journal of Educational Measurement, 38(4), 319-342. http://dx.doi.org/10.1111/j.1745-3984.2001.tb01130.x

Kane, M. T. (2013). Validating the interpretations and uses of test scores. Journal of Educational Measurement, 50(1), 1-73. http://dx.doi.org/10.1111/jedm.12000

Kang, H. (2013). The prevention and handling of the missing data. Korean journal of anesthesiology, 64(5), 402-406.

Kim, B.-S., Lee, D.-W., Bae, J.-N., Kim, J.-H., Kim, S., Kim, K. W., . . . Chang, S. M. (2017). Effects of education on differential item functioning on the 15-item modified Korean version of the Boston Naming Test. Psychiatry investigation, 14(2), 126-135.

Kim, J.-O., & Curry, J. (1977). The treatment of missing data in multivariate analysis. Sociological Methods & Research, 6(2), 215-240.

Kimberlin, C. L., & Winterstein, A. G. (2008). Validity and reliability of measurement instruments used in research. American Journal of Health-System Pharmacy, 65(23), 2276-2284. https://doi.org/10.2146/ajhp070364

Kinnunen, P., & Simon, B. (2010). *Experiencing programming assignments in CS1: The emotional toll.* In Proceedings of the Sixth international workshop on Computing education research, Aarhus, Denmark, *ACM*. http://dx.doi.org/10.1145/1839594.1839609

Kline, T. B. J. (1998). *Principles and practice of structural equation modeling*. New York: Guilford Press.

Kline, T. B. J. (2005). *Psychological testing: A practical approach to design and evaluation*: SAGE Publications, Thousand Oaks, CA.

Knapp, T. R. (1990). Treating ordinal scales as interval scales: An attempt to resolve the controversy. Nursing Research, 39(2), 121-123. https://doi.org/10.1097/00006199-199003000-00019

Kochan, S. G. (2015). *Programming in C*: Pearson Education.

Kölling, M. (2015). Lessons from the design of three educational programming environments: Blue, blueJ and Greenfoot. International Journal of People-Oriented Programming (IJPOP), 4(1), 5-32. http://dx.doi.org/10.4018/IJPOP.2015010102

Kölling, M., Quig, B., Patterson, A., & Rosenberg, J. (2003). The BlueJ System and its pedagogy. Computer Science Education, 13(4), 249-268. http://dx.doi.org/10.1076/csed.13.4.249.17496

Koulouri, T., Lauria, S., & Macredie, R. D. (2015). Teaching introductory programming: A quantitative evaluation of different approaches. ACM Transactions on Computing Education (TOCE), 14(4), 26. http://dx.doi.org/10.1145/2662412

Kreitzer, A. E., & Madaus, G. F. (1994). Empirical investigations of the hierarchical structure of the taxonomy. In L. A. Anderson & L. A. Sosniak (Eds.), *Bloom's taxonomy: A forty-year retrospective. Ninety-third yearbook for the National Society for the Study of Education: Part II* (pp. 64-81): Chicago: The University of Chicago Press. .

Kropp, R. P. (1966). The construction and validation of tests of the cognitive processes as described in the" taxonomy of educational objectives". Tallahassee: Florida State University, Institute of Human Learning and Department of Educational Research and Testing. Retrieved from https://files.eric.ed.gov/fulltext/ED010044.pdf

Kunkle, W. M., & Allen, R. B. (2016). The impact of different teaching approaches and languages on student learning of introductory programming concepts. ACM Transactions on Computing Education (TOCE), 16(1), 3. https://doi.org/10.1145/2785807

Kurtz, B. L. (1980). Investigating the relationship between the development of abstract reasoning and performance in an introductory programming class. ACM SIGCSE Bulletin, 12(1), 110-117. http://dx.doi.org/10.1145/953032.804622

Lai, J. S., Cella, D., Chang, C. H., Bode, R. K., & Heinemann, A. W. (2003). Item banking to improve, shorten and computerize self-reported fatigue: an illustration of steps to create a core item bank from the FACIT-Fatigue Scale. Quality of Life Research, 12(5), 485-501. Retrieved from https://link.springer.com/article/10.1023/A:1025014509626

Lakanen, A. J., Lappalainen, V., & Isomöttönen, V. (2015). *Revisiting rainfall to explore exam questions and performance on CS1*. In Proceedings of the 15th Koli Calling Conference on Computing Education Research, Koli, Finland, *ACM*. http://dx.doi.org/10.1145/2828959.2828970

Lambert, L. (2015). Factors that predict success in CS1. Journal of Computing Sciences in Colleges, 31(2), 165-171. Retrieved from https://dl.acm.org/citation.cfm?id=2831458

Lee, H. K. O. (2012). Physical Education in Higher Education in Hong Kong: The Effects of the Intervention on Pre-service Sports Coaches' Attitudes Towards Assessment for Learning Used in Sports *Self-directed Learning Oriented Assessments in the Asia-Pacific* (pp. 359-392): Springer.

Lee, M. J., & Ko, A. J. (2011). *Personifying programming tool feedback improves novice programmers' learning.* In Proceedings of the seventh international workshop on Computing education research, Providence, Rhode Island, USA, *ACM*. https://doi.org/10.1145/2016911.2016934

Lee, M. J., & Ko, A. J. (2015). *Comparing the effectiveness of online learning approaches on CS1 learning outcomes.* In Proceedings of the 11th Annual International Conference on International Computing Education Research, Omaha, Nebraska, USA, *ACM*. http://dx.doi.org/10.1145/2787622.2787709

Leeper, R. R., & Silver, J. L. (1982). Predicting success in a first programming course. SIGCSE Bull, 14(1), 147-150. https://doi.org/10.1145/800066.801357

Lennon, R. T. (1956). Assumptions underlying the use of content validity. Educational and Psychological Measurement, 16(3), 294-304. http://dx.doi.org/10.1177/001316445601600303

Leping, V., Lepp, M., Niitsoo, M., Tõnisson, E., Vene, V., & Villems, A. (2009). *Python prevails.* In Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, Ruse, Bulgaria, *ACM*. https://doi.org/10.1145/1731740.1731833

Lewis, M. C., Blank, D., Bruce, K., & Osera, P.-M. (2016). *Uncommon teaching languages*. Paper presented at the Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, Tennessee, USA. https://doi.org/10.1145/2839509.2844666

Lim, S. M., Rodger, S., & Brown, T. (2009). Using Rasch analysis to establish the construct validity of rehabilitation assessment tools. International Journal of Therapy & Rehabilitation, 16(5). http://dx.doi.org/10.12968/ijtr.2009.16.5.42102

Linacre, J. M. (1994). Sample size and item calibration stability. Rasch Measurement Transactions, 7(4), 328. Retrieved from https://www.rasch.org/rmt/rmt74m.htm

Linacre, J. M. (1998). Detecting multidimensionality: Which residual data-type works best? Journal of Outcome Measurement, 2, 266-283. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/9711024

Linacre, J. M. (1999). Investigating rating scale category utility. Journal of Outcome Measurement, 3, 103-122. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/10204322

Linacre, J. M. (2000). Item discrimination and infit mean-squares. Rasch Measurement Transactions, 14(2), 743. Retrieved from https://www.rasch.org/rmt/rmt142a.htm

Linacre, J. M. (2002). Optimizing rating scale category effectiveness. J Appl Meas, 3(1), 85-106. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/11997586

Linacre, J. M. (2005). Correlation coefficients: Describing relationships. Rasch Measurement Transactions, 19(3), 1028-1029. Retrieved from https://www.rasch.org/rmt/rmt193c.htm

Linacre, J. M. (2014). Reliability and separation of measures. *A user's guide to Winsteps Ministep Rasch-model computer programs (version 3.81. 0).* Retrieved from http://www.winsteps.com/winman/reliability.htm

Linderbaum, B. A., & Levy, P. E. (2010). The development and validation of the Feedback Orientation Scale (FOS). Journal of Management, 36(6), 1372-1405. https://doi.org/10.1177/0149206310373145

Lissitz, R. W. (2009). *The concept of validity: Revisions, new directions and applications*: IAP.

Lister, R. (2001). Objectives and objective assessment in CS1. ACM SIGCSE Bulletin, 33(1), 292-296. https://doi.org/10.1145/364447.364605

Lister, R. (2011). Ten years after the McCracken Working Group. ACM Inroads, 2(4), 18-19. https://doi.org/10.1145/2038876.2038882

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., . . . Seppälä, O. (2004). A multi-national study of reading and tracing skills in novice programmers. ACM SIGCSE Bulletin, 36(4), 119-150. https://doi.org/10.1145/1044550.1041673

Lister, R., Fidge, C., & Teague, D. (2009). Further evidence of a relationship between explaining, tracing and writing skills in introductory programming. ACM SIGCSE Bulletin, 41(3), 161-165. https://doi.org/10.1145/1562877.1562930

Lister, R., & Leaney, J. (2003). Introductory programming, criterion-referencing, and bloom. ACM SIGCSE Bulletin, 35(1), 143-147. https://doi.org/10.1145/792548.611954

Lister, R., Simon, B., Thompson, E., Whalley, J. L., & Prasad, C. (2006). Not seeing the forest for the trees: novice programmers and the SOLO taxonomy. ACM SIGCSE Bulletin, 38(3), 118-122. https://doi.org/10.1145/1140123.1140157

Lopez, M., Sutton, K., & Clear, T. (2009). *Surely we must learn to read before we learn to write!* In Proceedings of the Eleventh Australasian Conference on Computing Education, Wellington, New Zealand, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=1862736

Lopez, M., Whalley, J., Robbins, P., & Lister, R. (2008). *Relationships between reading, tracing and writing skills in introductory programming.* In Proceedings of the fourth international workshop on computing education research, Sydney, Australia, *ACM*. https://doi.org/10.1145/1404520.1404531

Lord, F., & Novick, M. (1968). Statistical Theories of Mental Test Scores. Addison-Westley Publ. Co. Reading, Mass.

Loui, M. C. (1995). Computer science is a new engineering discipline. ACM Computing Surveys (CSUR), 27(1), 31-32. https://doi.org/10.1145/214037.214049

Luce, R. D., & Tukey, J. W. (1964). Simultaneous conjoint measurement: A new type of fundamental measurement. Journal of Mathematical Psychology, 1(1), 1-27. https://doi.org/10.1016/0022-2496(64)90015-x

Luxton-Reilly, A., & Petersen, A. (2017). *The compound nature of novice programming assessments.* In Proceedings of the Nineteenth Australasian Computing Education Conference, *ACM*. Retrieved from https://dl.acm.org/citation.cfm?id=3174784

Magno, C. (2009). Demonstrating the difference between Classical Test Theory and Item Response Theory using derived test data. The International Journal of Educational and Psychological Assessment, 1(1), 1-11. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1426043

Malhotra, N. K. (1987). Analyzing marketing research data with incomplete information on the dependent variable. Journal of marketing research, 74-84.

Mannila, L., & de Raadt, M. (2006). *An objective comparison of languages for teaching introductory programming.* In Proceedings of the sixth Baltic Sea conference on Computing education research: Koli Calling 2006, Uppsala, Sweden, *ACM*. https://doi.org/10.1145/1315803.1315811

Marais, I. (2009). Response dependence and the measurement of change. Journal of Applied Measurement, 10(1), 17-29. Retrieved from http://europepmc.org/abstract/med/19299882

Marais, I. (2013). Local dependence. In S. Kreiner & M. Mesbah (Eds.), *Rasch models in health* (pp. 111-130). London, UK: Wiley-ISTE Ltd. https://doi.org/10.1002/9781118574454.ch7

Marais, I. (2014). Implications of removing random guessing from Rasch item estimates in Vertical Scaling. Journal of Applied Measurement, 16(2), 113-128. Retrieved from http://europepmc.org/abstract/MED/26075662

Marais, I., & Andrich, D. (2008). Formalizing dimension and response violations of local independence in the unidimensional Rasch model. Journal of Applied Measurement, 9(3), 200-215. Retrieved from http://europepmc.org/abstract/MED/18753691

Masapanta-Carrión, S., & Velázquez-Iturbide, J. Á. (2018). *A systematic review of the use of Bloom's taxonomy in Computer Science education.* In Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland, USA, *ACM*. https://doi.org/10.1145/3159450.3159491

Masters, G. N. (1982). A Rasch model for partial credit scoring. Psychometrika, 47(2), 149-174. https://doi.org/10.1007/bf02296272

Masters, G. N. (1988). Measurement models for ordered response categories. In R. Langeheine & J. Rost (Eds.), *Latent trait and latent class models* (pp. 11-29): Springer. https://doi.org/10.1007/978-1-4757-5644-9_2

Maxwell, S. E., & Delaney, H. D. (1985). Measurement and statistics: An examination of construct validity. Psychological Bulletin, 97(1), 85. https://doi.org/10.1037//0033-2909.97.1.85

Maxwell, S. E., Delaney, H. D., & Manheimer, J. M. (1985). ANOVA of residuals and ANCOVA: Correcting an illusion by using model comparisons and graphs. Journal of Educational Statistics, 10(3), 197-209. https://doi.org/10.2307/1164792

McAllister, S. (2008). Introduction to the use of Rasch analysis to assess patient performance. International Journal of Therapy & Rehabilitation, 15(11). https://doi.org/10.2307/1164792

McCamey, R. (2014). A Primer on the One-Parameter Rasch Model. American Journal of Economics and Business Administration, 6(4), 159. https://doi.org/10.3844/ajebasp.2014.159.163

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., . . . Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. ACM SIGCSE Bulletin, 33(4), 125-180. https://doi.org/10.1145/572134.572137

McDonald, J. H. (2009). *Handbook of biological statistics* (Vol. 2): sparky house publishing Baltimore, MD.

McKnight, P. E., McKnight, K. M., Sidani, S., & Figueredo, A. J. (2007). *Missing data: A gentle introduction*: Guilford Press.

McPheron, B. D., Gratiano, S. M., & Palm, W. J. (2015). *Does choice of programming language affect student understanding of programming concepts in a first year engineering course?* In Proceedings of the Seventh Annual First Year Engineering Education Conference, Roanoke, VA. Retrieved from https://docs.rwu.edu/cgi/viewcontent.cgi?article=1006&context=seccm_fp

Merbitz, C., Morris, J., & Grip, J. C. (1989). Ordinal scales and foundations of misinference. Archives of Physical Medicine and Rehabilitation, 70(4), 308-312. Retrieved from https://www.archives-pmr.org/article/0003-9993(89)90151-2/fulltext

Messick, S. (1988). The once and future issues of validity: Assessing the meaning and consequences of measurement. ETS Research Report Series, 1986(2). https://doi.org/10.1002/j.2330-8516.1986.tb00185.x

Messick, S. (1989). Validity. In R. Linn (Ed.), *Educational measurement* (3 ed., pp. 13-103): New York, NY: American Council on Education and Macmillan.

Messick, S. (1993). Foundations of validity: Meaning and consequences in psychological assessment. ETS Research Report Series, 1993(2). https://doi.org/10.1002/j.2333-8504.1993.tb01562.x

Messick, S. (1995). Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning. American Psychologist, 50(9), 741-749. http://dx.doi.org/10.1037/0003-066X.50.9.741

Messick, S. (1996). Validity and washback in language testing. ETS Research Report Series, 1996(1), 1-18. https://doi.org/10.1177/026553229601300302

Michell, J. (1990). *An introduction to the logic of psychological measurement*. Hillsdale: Erlbaum.

Michell, J. (1997). Quantitative science and the definition of measurement in psychology. British Journal of Psychology, 88(3), 355-383. https://doi.org/10.1111/j.2044-8295.1997.tb02641.x

Michell, J. (2000). Normal science, pathological science and psychometrics. Theory & Psychology, 10(5), 639-667.

MOK, M. M. C. (2010). Self-directed learning oriented assessment: Assessment that informs learning & empowers the learner.

Montiel-Overall, P. (2006). Implications of missing data in survey research. Canadian Journal of Information and Library Science, 30(3-4), 241-269. Retrieved from https://arizona.pure.elsevier.com/en/publications/implications-of-missing-data-in-survey-research

Moons, J., & De Backer, C. (2013). The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism. Computers & Education, 60(1), 368-384. http://dx.doi.org/10.1016/j.compedu.2012.08.009

Morshead, R. W. (1965). Taxonomy of educational objectives Handbook II: Affective domain. Studies in Philosophy and Education, 4(1), 164-170. Retrieved from https://link.springer.com/article/10.1007%2FBF00373956

Moskal, B., Lurie, D., & Cooper, S. (2004). Evaluating the effectiveness of a new instructional approach. ACM SIGCSE Bulletin, 36(1), 75-79. https://doi.org/10.1145/1028174.971328

Mullner, R. M. (2009). *Encyclopedia of health services research*. Thousand Oaks, California: Sage.

Narciss, S., & Huth, K. (2004). How to design informative tutoring feedback for multimedia learning. Instructional design for multimedia learning, 181195.

National Research Council. (2001). *Knowing what students know: The science and design of educational assessment*: National Academies Press.

Nguyen, C. D., Carlin, J. B., & Lee, K. J. (2017). Model checking in multiple imputation: an overview and case study. Emerging themes in epidemiology, 14(1), 8.

Nguyen, T. H., Han, H.-R., Kim, M. T., & Chan, K. S. (2014). An introduction to item response theory for patient-reported outcome measurement. The Patient-Patient-Centered Outcomes Research, 7(1), 23-35. https://doi.org/10.1007/s40271-013-0041-0

Nodoushan, M. S. (2009). Measurement theory in language testing: Past traditions and current trends. i-Manager's Journal on Educational Psychology, 3(2), 1. https://doi.org/10.26634/jpsy.3.2.1023

Norman, V. T., & Adams, J. C. (2015). *Improving non-CS major performance in CS1*. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, Kansas City, Missouri, USA, *ACM*. https://doi.org/10.1145/2676723.2677214

Novick, M. (1966). The axioms and principal results of classical test theory. Journal of Mathematical Psychology, 3(1), 1-18. https://doi.org/10.1002/j.2333-8504.1965.tb00132.x

Nowaczyk, R. H. (1983). *Cognitive Skills Needed in Computer Programming.* In Annual Meeting of the Southeastern Psychological Association, Atlanta, GA. Retrieved from https://eric.ed.gov/?id=ED236466

Nowaczyk, R. H. (1984). The relationship of problem-solving ability and course performance among novice programmers. International Journal of Man-Machine Studies, 21(2), 149-160. https://doi.org/10.1016/s0020-7373(84)80064-4

Olinsky, A., Chen, S., & Harlow, L. (2003). The comparative efficacy of imputation methods for missing data in structural equation modeling. European Journal of Operational Research, 151(1), 53-79. https://doi.org/10.1016/s0377-2217(02)00578-7

Oliver, D., Dobele, T., Greber, M., & Roberts, T. (2004). *This course has a Bloom Rating of 3.9.* In Proceedings of the Sixth Australasian Conference on Computing Education-Volume 30 (pp. 227-231), Dunedin, New Zealand, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=979998

Ott, C. F. P. (1988). *Predicting achievement in computer science through selected academic, cognitive, and demographic variables.* (Doctoral dissertation), Georgia State University - College of Education. Retrieved from http://dl.acm.org/citation.cfm?id=914537

Owolabi, J., Olanipekun, P., & Iwerima, J. (2014). Mathematics ability and anxiety, computer and programming anxieties, age and gender as determinants of achievement in basic programming. GSTF Journal on Computing (JoC), 3(4), 109 -109. https://doi.org/10.7603/s40601-013-0047-4

Pallant, J. F., & Tennant, A. (2007). An introduction to the Rasch measurement model: an example using the Hospital Anxiety and Depression Scale (HADS). British Journal of Clinical Psychology, 46(1), 1-18. https://doi.org/10.1348/014466506x96931

Parker, P. D., Marsh, H. W., Ciarrochi, J., Marshall, S., & Abduljabbar, A. S. (2013). Juxtaposing math self-efficacy and self-concept as predictors of long-term achievement outcomes. Educational Psychology, 34(1), 29-48. https://doi.org/10.1080/01443410.2013.797339

Partchev, I. (2004). A visual guide to Item Response Theory. Retrieved from https://pdfs.semanticscholar.org/8431/9dbff44dc42f80e8c1fbc3307138415ac242.pdf

Patterson, A., Kölling, M., & Rosenberg, J. (2003). Introducing unit testing with BlueJ. ACM SIGCSE Bulletin, 35(3), 11-15. https://doi.org/10.1145/961290.961518

Paul, R. W., & Binker, A. (1993). Critical thinking: What every person needs to survive in a rapidly changing world. NASSP Bulletin, 75(533), 120-122 https://doi.org/10.1177/019263659107553325

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., . . . Paterson, J. (2007). A survey of literature on the teaching of introductory programming. ACM SIGCSE Bulletin, 39(4), 204-223. https://doi.org/10.1145/1345375.1345441

Perkins, D., & Martin, F. (1986). Fragile knowledge and neglected strategies in novice programmers. In E. Soloway & S. Iyengar (Eds.), *Empirical studies of programmers, First Workshop* (Vol. 1, pp. 213-229). Washington, DC: National Institute of Education Retrieved from https://eric.ed.gov/?id=ED295618

Perline, R., Wright, B. D., & Wainer, H. (1979). The Rasch model as additive conjoint measurement. Applied Psychological Measurement, 3(2), 237-255. https://doi.org/10.1177/014662167900300213

Perneger, T. V. (1998). What's wrong with Bonferroni adjustments. Bmj, 316(7139), 1236-1238. https://doi.org/10.1136/bmj.316.7139.1236

Petersen, A., Craig, M., & Zingaro, D. (2011). *Reviewing CS1 exam question content.* In Proceedings of the 42nd ACM technical symposium on Computer science education, Dallas, TX, USA, *ACM*. https://doi.org/10.1145/1953163.1953340

Peugh, J. L., & Enders, C. K. (2004). Missing data in educational research: A review of reporting practices and suggestions for improvement. Review of Educational Research, 74(4), 525-556. https://doi.org/10.3102/00346543074004525

Pickard, M. J. (2007). The new Bloom's taxonomy: An overview for family and consumer sciences. Journal of Family and Consumer Sciences Education, 25(1), 45-55. Retrieved from https://uncwweb.uncw.edu/cas/documents/PickardNewBloomsTaxonomy.pdf

Pillay, N., & Jugoo, V. R. (2005). An investigation into student characteristics affecting novice programming performance. ACM SIGCSE Bulletin, 37(4), 107-110. https://doi.org/10.1145/1113847.1113888

Popper, K. (2014). *Conjectures and refutations: The growth of scientific knowledge*: routledge.

Porter, L., & Zingaro, D. (2014). *Importance of early performance in CS1: two conflicting assessment stories.* In Proceedings of the 45th ACM technical symposium on Computer science education, Atlanta, Georgia, USA, *ACM*. https://doi.org/10.1145/2538862.2538912

Potter, M., & Kustra, E. (2012). A primer on learning outcomes and the SOLO taxonomy. Retrieved from http://www1.uwindsor.ca/ctl/system/files/PRIMER-on-Learning-Outcomes.pdf

Poundstone, W. (2014). *Rock Breaks Scissors: A Practical Guide to Outguessing and Outwitting Almost Everybody*: Little, Brown and Company.

Price, K., & Smith, S. (2014). Improving student performance in CS1. Journal of Computing Sciences in Colleges, 30(2), 157-163. Retrieved from https://dl.acm.org/citation.cfm?id=2667454

Putano, B. (2018). Most popular and influential programming languages of 2018. Retrieved from https://stackify.com/popular-programming-languages-2018/

Python. (2015) (Version 3.4.3) [Computer Software]: Python Software Foundation. Retrieved from https://www.python.org/downloads/release/python-343/

Raigoza, J. (2017). *A study of students' progress through introductory computer science programming courses.* In Frontiers in Education Conference *IEEE*. https://doi.org/10.1109/fie.2017.8190559

Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. ACM SIGCSE Bulletin, 36(3), 171-175. https://doi.org/10.1145/1026487.1008042

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests*. Chicago: University of Chicago Press.

Rasch, G. (1961). *On general laws and the meaning of measurement in psychology.* In Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, Berkeley, CA: University of California Press. Retrieved from http://econ.ucsb.edu/~doug/245a/Papers/Meaning%20of%20Measurement.pdf

Rizvi, M., Humphries, T., Major, D., Jones, M., & Lauzun, H. (2011). A CS0 course using scratch. Journal of Computing Sciences in Colleges, 26(3), 19-27. Retrieved from https://dl.acm.org/citation.cfm?id=1859166

Roberts, E., & Engel, G. (2001). *Computing Curricula 2001: Final report of the joint ACM/IEEE-CS task force on computer science education*. Retrieved from http://www.sigcse.org/cc2001/cs-introductory-courses.html.

Robins, A. (2010). Learning edge momentum: A new account of outcomes in CS1. Computer Science Education, 20(1), 37-71. https://doi.org/10.1080/08993401003612167

Romanoski, J., & Douglas, G. (2002). Rasch-transformed raw scores and two-way ANOVA: A simulation analysis. Journal of Applied Measurement, 3(4), 421-430. Retrieved from https://eric.ed.gov/?id=EJ662107

Roth, P. L. (1994). Missing data: A conceptual review for applied psychologists. Personnel Psychology, 47(3), 537-560. https://doi.org/10.1111/j.1744-6570.1994.tb01736.x

Rountree, J., Rountree, N., Robins, A., & Hannah, R. (2005). *Observations of student competency in a CS1 course.* In Proceedings of the Seventh Australasian conference on Computing education, Newcastle, New South Wales, Australia, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=1082442

Rountree, N., Rountree, J., & Robins, A. (2002). Predictors of success and failure in a CS1 course. ACM SIGCSE Bulletin, 34(4), 121-124. https://doi.org/10.1145/820127.820182

Royal, K. D. (2010). Making meaningful measurement in survey research: A demonstration of the utility of the rasch model. IR Applications, 28, 1-16. Retrieved from https://files.eric.ed.gov/fulltext/ED524824.pdf

Royal, K. D., & Eli, J. A. (2013). Developing a psychometric ruler: An alternative presentation of rasch measurement output. Journal of Applied Quantitative Methods, 8(3), 1-10. Retrieved from http://jaqm.ro/issues/volume-8,issue-3/pdfs/1_royal_eli.pdf

Royal, K. D., & Hedgpeth, M. (2013). Investigating guessing strategies and their success rates on items of varying difficulty levels. Rasch Measurement Transactions, 27, 1407-1408. Retrieved from https://www.rasch.org/rmt/rmt271d.htm

Rubin, D. B. (1976). Inference and missing data. Biometrika, 63(3), 581-592. https://doi.org/10.1093/biomet/63.3.581

Rupp, A. A., & Zumbo, B. D. (2006). Understanding parameter invariance in unidimensional IRT models. Educational and Psychological Measurement, 66(1), 63-84. https://doi.org/10.1177/0013164404273942

Sajaniemi, J., & Hu, C. (2006). Teaching programming: Going beyond "Objects First". In S. Bryant (Ed.), *18th Workshop of the Psychology of Programming Interest Group* (pp. 255-265). UK: University of Sussex. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.3914&rep=rep1&type=pdf

Saltstone, R., Skinner, C., & Tremblay, P. (2001). Conditional standard error of measurement and personality scale scores: An investigation of classical test theory estimates with four MMPI scales. Personality and Individual Differences, 30(4), 691-698. http://dx.doi.org/10.1016/S0191-8869(00)00065-9

Salzberger, T. (1999). *How the Rasch model may shift our perspective of measurement in marketing research.* In Proceedings of the 1999 Australia and New Zealand Marketing Academy Conference (ANZMAC), Sydney, NSW.

Salzberger, T. (2013). Attempting measurement of psychological attributes. Frontiers in psychology, 4, 75. https://doi.org/10.3389/fpsyg.2013.00075

Sampaio, C., Goetz, C. G., & Schrag, A. (2012). *Rating scales in Parkinson's disease: clinical practice and research*: Oxford University Press.

Sauter, V. L. (1986). Predicting computer programming skill. Computers & Education, 10(2), 299-302. https://doi.org/10.1016/0360-1315(86)90031-x

Sawatzky, R., Chan, E. K., Zumbo, B. D., Ahmed, S., Bartlett, S. J., Bingham, C. O., . . . Sajobi, T. (2017). Montreal Accord on patient-reported outcomes (PROs) use series–Paper 7: Modern perspectives of measurement validation emphasize justification of

inferences based on patient reported outcome scores. Journal of Clinical Epidemiology, 89, 154-159. https://doi.org/10.1016/j.jclinepi.2016.12.002

Schulte, C. (2008). *Block Model: An educational model of program comprehension as a tool for a scholarly approach to teaching.* In Proceedings of the Fourth international Workshop on Computing Education Research, Sydney, Australia, *ACM*. https://doi.org/10.1145/1404520.1404535

Scott, T. (2003). Bloom's taxonomy applied to testing in computer science classes. Journal of Computing Sciences in Colleges, 19(1), 267-274. Retrieved from https://dl.acm.org/citation.cfm?id=948775

Sébille, V., Hardouin, J.-B., Le Néel, T., Kubis, G., Boyer, F., Guillemin, F., & Falissard, B. (2010). Methodological issues regarding power of classical test theory (CTT) and item response theory (IRT)-based approaches for the comparison of patient-reported outcomes in two groups of patients - a simulation study. BMC Medical Research Methodology, 10(1), 24. https://doi.org/10.1186/1471-2288-10-24

Sekiya, T., & Yamaguchi, K. (2013). *Tracing quiz set to identify novices' programming misconceptions.* In Proceedings of the 13th Koli Calling International Conference on Computing Education Research, *ACM*. https://doi.org/10.1145/2526968.2526978

Shavelson, R. J., & Webb, N. M. (1991). *Generalizability theory: A primer* (Vol. 1): Sage Publications.

Sheard, J., Carbone, A., Chinn, D., Clear, T., Corney, M., D'Souza, D., . . . Teague, D. (2013). *How difficult are exams?: A framework for assessing the complexity of introductory programming exams.* In Proceedings of the 15th Australasian Computing Education Conference-Volume 136 (pp. 145-154), *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=2667215

Sheard, J., Carbone, A., Chinn, D., Laakso, M.-J., Clear, T., de Raadt, M., . . . Philpott, A. (2011). *Exploring programming assessment instruments: A classification scheme for examination questions.* In Proceedings of the seventh international workshop on Computing education research, Providence, Rhode Island, USA, *ACM*. https://doi.org/10.1145/2016911.2016920

Sheard, J., Carbone, A., Lister, R., Simon, B., Thompson, E., & Whalley, J. L. (2008). Going SOLO to assess novice programmers. ACM SIGCSE Bulletin, 40(3), 209-213. https://doi.org/10.1145/1384271.1384328

Sheard, J., Dermoudy, J., D'Souza, D., Hu, M., & Parsons, D. (2014). *Benchmarking a set of exam questions for introductory programming.* In Proceedings of the 16th

Australasian Computing Education Conference-Volume 148 (pp. 113-121), Auckland, New Zealand, *Australian Computer Society, Inc*. Retrieved from https://dl.acm.org/citation.cfm?id=2667504

Shein, E. (2015). Python for beginners. Communications of the ACM, 58(3), 19-21. 10.1145/2716560 Retrieved from https://dl.acm.org/citation.cfm?id=2716560

Shneiderman, B., & Mayer, R. (1979). Syntactic/semantic interactions in programmer behavior: A model and experimental results. International Journal of Parallel Programming, 8(3), 219-238. https://doi.org/10.1007/bf00977789

Shuhidan, S., Hamilton, M., & D'Souza, D. (2009). *A taxonomic study of novice programming summative assessment.* In Proceedings of the Eleventh Australasian Conference on Computing Education-Volume 95 (pp. 147-156), Wellington, New Zealand, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=1862734

Sick, J. (2010). Assumptions and requirements of Rasch measurement. Shiken: JALT Testing & Evaluation SIG Newsletter, 14(2), 23-29. Retrieved from http://hosted.jalt.org/test/sic_5.htm

Sijtsma, K., & Van der Ark, L. A. (2003). Investigation and treatment of missing item scores in test and questionnaire data. Multivariate Behavioral Research, 38(4), 505-528. https://doi.org/10.1207/s15327906mbr3804_4

Sireci, S. G., & Sukin, T. (2013). Test validity. In K. F. Geisinger, B. A. Bracken, J. F. Carlson, J.-I. C. Hansen, S. P. R. Nathan R. Kuncel, & M. C. Rodriguez (Eds.), APA handbooks in psychology. APA handbook of testing and assessment in psychology, Vol. 1. Test theory and testing and assessment in industrial and organizational psychology (pp. 61-84). Washington, DC, USA: American Psychological Association. doi:http://dx.doi.org/10.1037/14047-004

Sjaastad, J. (2014). Enhancing measurement in science education research through Rasch analysis: Rationale and properties. Nordic Studies in Science Education, 10(2), 212-230. https://doi.org/10.5617/nordina.662

Smith, E. V. (2001). Evidence for the reliability of measures and validity of measure interpretation: A Rasch measurement perspective. Journal of Applied Measurement, 2(3), 281-311. Retrieved from http://psycnet.apa.org/record/2002-01664-005

Smith, E. V. (2002). Understanding Rasch measurement: Detecting and evaluating the impact of multidimensionality using item fit statistics and principal component analysis of

residuals. Journal of Applied Measurement, 3, 205–231. Retrieved from http://psycnet.apa.org/record/2002-01670-005

Smits, N., Cuijpers, P., & van Straten, A. (2011). Applying computerized adaptive testing to the CES-D scale: A simulation study. Psychiatry research, 188(1), 147-155. https://doi.org/10.1016/j.psychres.2010.12.001

Snowdon, S. (2011). *Explaining program code: giving students the answer helps-but only just.* In Proceedings of the seventh international workshop on Computing education research, Providence, Rhode Island, USA, *ACM*. https://doi.org/10.1145/2016911.2016931

Soh, S. E., Barker, A., Morello, R., Dalton, M., & Brand, C. (2016). Measuring safety climate in acute hospitals: Rasch analysis of the safety attitudes questionnaire. BMC health services research, 16(1), 497. https://doi.org/10.1186/s12913-016-1744-4

Soloway, E. (1986). Learning to program = learning to construct mechanisms and explanations. Communications of the ACM, 29(9), 850-858. https://doi.org/10.1145/6592.6594

Soloway, E., Ehrlich, K., Bonar, J., & Greenspan, J. (1982). What do novices know about programming. In A. Badre & B. Schneiderman (Eds.), *Directions in human-computer interaction* (pp. 27-54). Norwood, Nj: Ablex.

Spearman, C. (1904). The proof and measurement of association between two things. The American journal of psychology, 15(1), 72-101. https://doi.org/10.1037/11491-005

Stanger-Hall, K. F. (2012). Multiple-choice exams: an obstacle for higher-level thinking in introductory science classes. CBE-Life Sciences Education, 11(3), 294-306. https://doi.org/10.1187/cbe.11-11-0100

Stanley, J. C., & Bolton, D. L. (1957). A review of "Bloom's Taxanomy of objectives". Educational and Psychological Measurement, 17, 631-634.

Starr, C. W., Manaris, B., & Stalvey, R. H. (2008). Bloom's taxonomy revisited: Specifying assessable learning objectives in computer science. ACM SIGCSE Bulletin, 40(1), 261-265. https://doi.org/10.1145/1352135.1352227

Stefik, A., & Siebert, S. (2013). An Empirical Investigation into Programming Language Syntax. Transactions on Computing Education, 13(4), 1946-6226. https://doi.org/10.1145/2534973

Steinberg, L., & Thissen, D. (1996). Uses of item response theory and the testlet concept in the measurement of psychopathology. Psychological Methods, 1(1), 81. https://doi.org/10.1037//1082-989x.1.1.81

Stenner, J. (2001). The necessity of construct theory. Rasch Measurement Transactions, 15(1), 804-805. Retrieved from https://www.rasch.org/rmt/rmt151q.htm

Stephenson, C., & West, T. (1998). Language choice and key concepts in introductory computer science courses. Journal of Research on Computing in Education, 31(1), 89-95. https://doi.org/10.1080/08886504.1998.10782243

Stevens, S. S. (1946). On the theory of scales of measurement. Science, 103, 677–680. Retrieved from http://gaius.fpce.uc.pt/niips/novoplano/mip1/mip1_201314/scales/Stevens_1946.pdf

Stout, W. (1987). A nonparametric approach for assessing latent trait unidimensionality. Psychometrika, 52(4), 589-617. https://doi.org/10.1007/bf02294821

Strauss, M. E., & Smith, G. T. (2009). Construct validity: Advances in theory and methodology. Annual review of clinical psychology, 5, 1-25.

Streib, J. T., & Soma, T. (2014). *Guide to java: A concise introduction to programming*. London: Springer.

Streiner, D. L. (2003). Starting at the beginning: An introduction to coefficient alpha and internal consistency. Journal of Personality Assessment, 80(1), 99-103. https://doi.org/10.1207/s15327752jpa8001_18

Strnad, M., Šerbec, I. N., & Rugelj, J. (2009). *Programming aptitude and learning success in the introductory course on programming*. Paper presented at the 12th International Conference on Interactive Computer aided Learning, Villach, Austria Retrieved from https://pdfs.semanticscholar.org/24e3/9e8ecf38f6852d5243fede749e0cc6386af5.pdf

Sugrue, B. (2002). Problems with Bloom's taxonomy. Retrieved from https://eppicinc.files.wordpress.com/2011/08/sugrue_bloom_critique_perfxprs.pdf

Sumintono, B. (2017). Rasch Model Measurement as Tools in Assessment for Learning.

Tafliovich, A., Campbell, J., & Petersen, A. (2013). *A student perspective on prior experience in CS.* In Proceeding of the 44th ACM technical symposium on Computer science education, Denver, Colorado, USA, *ACM*. https://doi.org/10.1145/2445196.2445270

Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. International journal of medical education, 2, 53. https://doi.org/10.5116/ijme.4dfb.8dfd

Tennant, A., & Conaghan, P. G. (2007). The Rasch measurement model in rheumatology: what is it and why use it? When should it be applied, and what should one look for in a Rasch paper? Arthritis Care & Research, 57(8), 1358-1362. https://doi.org/10.1002/art.23108

Tennant, A., & Pallant, J. (2006). Unidimensionality matters (A Tale of Two Smiths?). Rasch Measurement Transactions, 20(1), 1048-1051. Retrieved from https://www.rasch.org/rmt/rmt201c.htm

Tennant, A., & Pallant, J. (2007). DIF matters: a practical approach to test if differential item functioning makes a difference. Rasch Measurement Transactions, 20(4), 1082-1084.

Tew, A. E. (2010). *Assessing fundamental introductory computing concept knowledge in a language independent manner.* Georgia Institute of Technology. Retrieved from https://search.proquest.com/openview/35877528a9ec75089cff5306e88109d7/1?pq-origsite=gscholar&cbl=18750&diss=y

Tew, A. E., & Guzdial, M. (2010). *Developing a validated assessment of fundamental CS1 concepts.* In Proceedings of the 41st ACM technical symposium on Computer science education, Milwaukee, Wisconsin, USA, *ACM*. https://doi.org/10.1145/1734263.1734297

The Maldives National University. (2018). Faculty of Engineering, Science and Technology. Retrieved from http://fest.mnu.edu.mv/

Thompson, E., Luxton-Reilly, A., Whalley, J. L., Hu, M., & Robbins, P. (2008). *Bloom's taxonomy for CS assessment.* In Proceedings of the tenth conference on Australasian computing education, Wollongong, NSW, Australia, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=1379265

Thurstone, L. L. (1927). A law of comparative judgment. Psychological review, 34(4), 273. http://dx.doi.org/10.1037/h0070288

Thurstone, L. L. (1931). The measurement of social attitudes. The journal of Abnormal and Social Psychology, 26(3), 249. http://dx.doi.org/10.1037/h0070363

Thurstone, L. L., & Chave, E. J. (1929). *The Measurement of Attitude*: University of Chicago Press.

Tran, H. P., Griffin, P., & Nguyen, C. (2010). Validating the university entrance English test to the Vietnam National University: A conceptual framework and methodology. Procedia-Social and Behavioral Sciences, 2(2), 1295-1304. https://doi.org/10.1016/j.sbspro.2010.03.190

Traynor, D., Bergin, S., & Gibson, J. P. (2006). *Automated assessment in CS1*. In Proceedings of the 8th Australasian Conference on Computing Education-Volume 52, Hobart, Australia, *Australian Computer Society, Inc.*

Tu, J. J., & Johnson, J. R. (1990). Can computer programming improve problem-solving ability? ACM SIGCSE Bulletin, 22(2), 30-33. https://doi.org/10.1145/126445.126451

Usher, A., & Kober, N. (2012). Student Motivation: An Overlooked Piece of School Reform. Summary. Center on Education Policy. Retrieved from https://eric.ed.gov/?id=ED532666

Venables, A., Tan, G., & Lister, R. (2009). *A closer look at tracing, explaining and code writing skills in the novice programmer*. In Proceedings of the fifth international workshop on Computing education research workshop, Berkeley, CA, USA, *ACM*. https://doi.org/10.1145/1584322.1584336

Verguts, T., & Boeck, P. (2001). Some Mantel-Haenszel tests of Rasch model assumptions. British Journal of Mathematical and Statistical Psychology, 54(1), 21-37. https://doi.org/10.1348/000711001159401

Verma, R., & Goodale, J. C. (1995). Statistical power in operations management research. Journal of Operations Management, 13(2), 139-152. https://doi.org/10.1016/0272-6963(95)00020-s

Vickers, A. J. (2005). Parametric versus non-parametric statistics in the analysis of randomized trials with non-normally distributed data. BMC Medical Research Methodology, 5(1), 35. https://doi.org/10.1186/1471-2288-5-35

Villa College. (2018). A brief history. Retrieved from http://villacollege.edu.mv/qi/public/

Wainer, H., Dorans, N. J., Flaugher, R., Green, B. F., & Mislevy, R. J. (2000). *Computerized adaptive testing: A primer*: Routledge.

Wainer, H., & Kiely, G. L. (1987). Item clusters and computerized adaptive testing: A case for testlets. Journal of Educational Measurement, 24(3), 185-201. https://doi.org/10.1111/j.1745-3984.1987.tb00274.x

Wang, J., Hong, H., Ravitz, J., & Ivory, M. (2015). *Gender differences in factors influencing pursuit of computer science and related fields.* In Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, Vilnius, Lithuania, *ACM*. https://doi.org/10.1145/2729094.2742611

Watson, C., & Li, F. W. (2014). *Failure rates in introductory programming revisited.* In Proceedings of the 2014 conference on Innovation & technology in computer science education, *ACM*. https://doi.org/10.1145/2591708.2591749

Watson, C., Li, F. W., & Godwin, J. L. (2013). *Predicting performance in an introductory programming course by logging and analyzing student programming behavior.* In Proceedings of the 2013 IEEE 13th International Conference on Advanced Learning Technologies (pp. 319-323), Beijing, China, *IEEE*. https://doi.org/10.1109/icalt.2013.99

Wetzel, E., & Carstensen, C. H. (2014). Reversed thresholds in partial credit models: a reason for collapsing categories? Assessment, 21(6), 765-774. https://doi.org/10.1177/1073191114530775

Whalley, J., Clear, T., Robbins, P., & Thompson, E. (2011). *Salient elements in novice solutions to code writing problems.* In Proceedings of the Thirteenth Australasian Computing Education Conference, Perth, Australia, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=2459941

Whalley, J., & Kasto, N. (2013). *Revisiting models of human conceptualisation in the context of a programming examination.* In Proceedings of the Fifteenth Australasian Computing Education Conference, Adelaide, Australia, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=2667207

Whalley, J. L., Lister, R., Thompson, E., Clear, T., Robbins, P., Kumar, P., & Prasad, C. (2006). *An Australasian study of reading and comprehension skills in novice programmers, using the Bloom and SOLO taxonomies.* In Proceedings of the 8th Australasian Conference on Computing Education-Volume 52, Hobart, Australia, *Australian Computer Society, Inc.* Retrieved from https://dl.acm.org/citation.cfm?id=1151901

Whipkey, K. L. (1984). Identifying predictors of programming skill. ACM SIGCSE Bulletin, 16(4), 36-42. https://doi.org/10.1145/382200.382544

White, G. (2003). Standardized mathematics scores as a prerequisite for a first programming course. Mathematics and Computer Education, 37(1), 96-104. Retrieved from http://search.proquest.com/docview/235809433?accountid=10382

Wiedenbeck, S. (2005). *Factors affecting the success of non-majors in learning to program*. Paper presented at the Proceedings of the first international workshop on Computing education research, Seattle, WA, USA. https://doi.org/10.1145/1089786.1089788

Wilson, B. C., & Shrock, S. (2001). Contributing to success in an introductory computer science course: a study of twelve factors. ACM SIGCSE Bulletin, 33(1), 184-188. https://doi.org/10.1145/364447.364581

Wilson, M. (2004). *Constructing Measures: An Item Response Modeling Approach*: Taylor & Francis.

Wilson, M. (2005). *Constructing measures: An item response modeling approach*: Mahwah, NJ:Lawrence Erlbaum Associates.

Wilson, M., & Sloane, K. (2000). From principles to practice: An embedded assessment system. Applied Measurement in Education, 13(2), 181-208. https://doi.org/10.1207/s15324818ame1302_4

Wolfe, E., & Smith, E. (2007a). Instrument development tools and activities for measure validation using Rasch models: Part I-Validation activities. Journal of Applied Measurement, 8(1), 97-123.

Wolfe, E., & Smith, E. (2007b). Instrument development tools and activities for measure validation using Rasch models: Part II--validation activities. Journal of Applied Measurement, 8(2), 204-234.

Wolfe, E. W., & Smith , E. V. (2007). Instrument development tools and activities for measure validation using Rasch models: Part II-Validation activities. Journal of Applied Measurement, 8(2), 204 -234.

Wothke, W. (1993). Nonpositive definite matrices in structural modeling. Sage Focus Editions, 154, 256-256.

Wright, B. D. (1977). Solving measurement problems with the Rasch model. Journal of Educational Measurement, 14(2), 97-116. https://doi.org/10.1111/j.1745-3984.1977.tb00031.x

Wright, B. D. (1985). Additivity in psychological measurement. In E. E. Roskam (Ed.), *Measurement and personality assessment* (pp. 101-112). Amsterdam: Elsevier Science. Retrieved from https://www.rasch.org/memo33b.pdf

Wright, B. D. (1996a). Comparing Rasch measurement and factor analysis. Structural Equation Modeling: A Multidisciplinary Journal, 3(1), 3-24. https://doi.org/10.1080/10705519609540026

Wright, B. D. (1996b). Local dependency, correlations and principal components. Rasch Measurement Transactions, 10(3), 509-511.

Wright, B. D. (1997). Fundamental measurement. Rasch Measurement Transactions, 11(2), 558. https://doi.org/10.4135/9781412985598.n2

Wright, B. D. (Ed.) (1999). *Fundamental measurement for psychology*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Wright, B. D., & Masters, G. N. (1982). *Rating Scale Analysis. Rasch Measurement*: Chicago: MESA Press.

Wright, B. D., & Stone, M. H. (1979). *Best Test Design. Rasch Measurement*: The University of Chicago, MESA Press.

Wu, M., & Adams, R. (2007). *Applying the Rasch model to psycho-social measurement: A practical approach*. Melbourne: Educational Measurement Solutions.

Wu, M., Tam, H. P., & Jen, T.-H. (2016). *Educational measurement for applied researchers*. Singapore: Springer.

Yamamoto, M., Sekiya, T., Mori, K., & Yamaguchi, K. (2012). *Skill hierarchy revised by SEM and additional skills*. Paper presented at the International Conference on Information Technology Based Higher Education and Training (ITHET), Istanbul, Turkey. https://doi.org/10.1109/ithet.2012.6246009

Zhang, B., & Walker, C. M. (2008). Impact of missing data on person—Model fit and person trait estimation. Applied Psychological Measurement, 32(6), 466-479.

Zingaro, D., & Porter, L. (2016). *Impact of student achievement goals on CS1 outcomes.* In Proceedings of the 47th ACM Technical Symposium on Computing Science Education, *ACM*. https://doi.org/10.1145/2839509.2844553

Appendices

Appendix I: Construct Map

Competence Levels **Constructs**	**Uni-structural** The student manifests a correct grasp of simple problems that do not integrate multiple concepts	**Multi-structural** The student manifests an understanding problems integrating multiple concepts	**Relational** The student manifests an understanding of the code as a single coherent whole, by describing the function	**Extended Abstract** The student is able to integrate in a non-simple manner two or more concepts to derive a coherent solution to the problem
Variables, expressions and assignments.	Trace the outcome of simple variable initialization, referencing, accessing and modifying variables.	Trace the output of a given code segment involving initializing referencing, accessing modifying a set of variables.	Explain by summarizing the outcome of a given code segment involving initializing referencing, accessing modifying a set of variables.	Write a code segment involving concepts such as, initializing referencing, accessing modifying a set of variables to achieve a given task.
Single, multiple and nested IF/ELSE structures (subsumes relational and logical operators	Trace the values of control variables in given code segment implementing simple IF/ELSE structure as they are being executed	Trace the output of a given code segment implementing multiple and nested IF/ELSE structure	Explain by summarizing the purpose of a given code segment implementing multiple and nested IF/ELSE structures.	Write a code segment implementing multiple nested If/Else structures to achieve a given task
Loops (subsumes relational and logical operators)	Trace the value of Loop control variables implementing single non-nested Loop structure as the code segment gets executed	Trace the output of a given code segment implementing one-level nested loops structure	Explain by summarizing the purpose of given code segment implementing one-level nested loop structure	Write a code segment implementing multiple nested loop structures integrating the concept of If/Else structure to achieve a given task
Functions/methods	Trace/identify the parameter types, return types based on the function signature.	Trace the output or the return value of given function/method which integrates substantial amount of other concepts such as Loops, If/Else structure	Explain by summarizing the purpose of given function/method which integrates substantial amount of other concepts such as Loops, If/Else structure	Write a complete function/method definition for a given scenario implementing substantial amount of other concepts such as Loops, If/Else structure
Arrays (single dimension) and	Trace the outcome of simple array processing concepts such	Trace the output of a code segment which implements basic array algorithms such as copying, sorting,	Explain by summarizing the purpose of a given code segment which implements basic array algorithms	Write a code segment implementing basic array manipulation task such as copying, sorting, searching, or

basic Array processing	as declaration, initialization and accessing array elements.	finding maximum/minimum and reversing of elements.	such as copying, sorting, searching, and reversing of elements.	reversing of elements to achieve a given task
Skill	**Basics/Data**	**Tracing**	**Explaining**	**Writing**

Appendix II : Scoring Model for Writing Questions (i.e., part (d))

Construct	1 (Uni-structural)	2 (Multi-Structural)	3 (Relational)	4 (Extended Abstract)
Variables, expressions, and assignments.	Partly correct code towards the progression of actual solution definition with correct initialization of variables	Mostly correct code and almost towards the progression of actual solution definition with correct initialization of variables producing a partial solution	Fully correct code producing the expected solution definition in simple linear fashion	Fully correct code producing the expected solution definition in an efficient non simple manner
Single, multiple and nested IF/ELSE structures (subsumes relational and logical operators) may subsume any of the concepts above	Partly correct code towards the progression of actual solution definition with inclusion of main IF/Else case with the correct conditions	Mostly correct code towards the progression of actual solution definition with inclusion of main IF/Else and nested IF/Else case with correct conditions producing a partial solution	Fully correct code producing the expected solution definition in simple linear fashion	Fully correct code producing the expected solution definition in an efficient non simple manner
Loops (subsumes relational and logical operators) may subsume any of the concepts above	Partly correct code towards the progression of actual solution definition with inclusion of main outer Loop with the correct conditions	Mostly correct code towards the progression of actual solution definition with inclusion of outer and nested loop with their correct conditions producing a partial solution	Fully correct code producing the expected solution definition in simple linear fashion	Fully correct code producing the expected solution definition in an efficient non simple manner
Functions/methods Subsumes may subsume any of the concepts above	Partly correct code towards the progression of actual solution definition which includes the correct method definition and return type	Mostly correct code towards the progression of actual solution definition which includes the correct method definition, return type and mostly correct logic producing a partial solution	Fully correct code producing the expected solution definition in simple linear fashion	Fully correct code producing the expected solution definition in an efficient non simple manner
Arrays (single dimension) and basic Array processing may subsume any of the concepts above	Partly correct code towards the progression of actual solution definition which includes initialization of variables and iteration of array elements	Mostly correct code towards the progression of actual solution definition which includes initialization of variables, iteration of loop variables and comparison of the values producing a partial solution	Fully correct code producing the expected solution definition in simple linear fashion	Fully correct code producing the expected solution definition in an efficient non simple manner

Appendix III: Expert Feedback

Expert Reviewer 1

Sample Feedback Forms

1. Do competencies in the construct model form a hierarchy from easy to difficult? Please comment

   Yes.

2. Are the questions in accord with construct model? Do they form the hierarchy postulated in the construct model.

   Yes.

3. Is the content appropriate and typically the content taught in the first year first CS1 courses? Please note that the content is not based on any specific programming pedagogy or paradigm. It represents the common concepts independent of any programming language, paradigm or pedagogy. Also note that the instrument will be translated to target participant's language of instruction such as Python, C/C++ etc at a later stage.

   Yes

4. Is phrasings of the questions clear and grammatically correct?

   Yes.

5. Any Typo error
   a) Code Tracing and Explaining (Section A) Question Three:
      "What is the value of ***y*** when the condition at ***line 10*** becomes false?"
      line 10 looks like ***line 13***

Appendix IV: Sample Question Set for Loops (Java Version)

Question Three

To answer parts (a), (b) and (c) of this question, refer to the code segment below.

```
1.      int x,y;
2.      int z=0;
3.      for (x=1; x<=4;x++ )
4.      {
5.           y=1;
6.           while(y<=x)
7.           {
8.
9.                z=z+x;
10.               y++;
11.
12.          }
13.     }
```

(a) When the above code segment is executed, how many times is the `for-loop` (outer loop) iterated?

Answer: _____

Answer: _____

(b) After the above code segment is executed, what is the value of `z`?

Answer: _____

(c) What is the purpose or what does the code segment do? [Do not give the line-by-line code explanation of what the code does, instead, write the summary or overall purpose of the code segment. Grammar is not important here]

(d) Write a code segment that prints out a right-angled triangle with `x` number of rows by filling the rows of the triangle with consecutive numbers, starting with 1 in the top left corner. For example, if the value of x = 4, the code segment should print the following triangle. Assume that the variable `x` has already been initialized to an integer value.

```
1
2   3
4   5   6
7   8   9   10
```

Fig 3.1: Expected output when x = 4

Appendix V: Ethics Approval

Office of Research and Development

GPO Box U1987
Perth Western Australia 6845

Telephone +61 8 9266 7863
Facsimile +61 8 9266 3793
Web research.curtin.edu.au

25-May-2016

Name: Robert Cavanagh
Department/School: School of Education
Email: R.Cavanagh@exchange.curtin.edu.au

Dear Robert Cavanagh

RE: Ethics approval
Approval number: RDHU-66-16

Thank you for submitting your application to the Human Research Ethics Office for the project **Student & Learning Environment Characteristics associated with Student Competency in Introductory Computer Programming**.

Your application was reviewed through the Curtin University low risk ethics review process.

The review outcome is: **Approved**.

Your proposal meets the requirements described in National Health and Medical Research Council's (NHMRC) *National Statement on Ethical Conduct in Human Research (2007)*.

Approval is granted for a period of one year from **25-May-2016** to **24-May-2017**. Continuation of approval will be granted on an annual basis following submission of an annual report.

Personnel authorised to work on this project:

Name	Role
Cavanagh, Robert	

Standard conditions of approval

1. Research must be conducted according to the approved proposal
2. Report in a timely manner anything that might warrant review of ethical approval of the project including:
   - proposed changes to the approved proposal or conduct of the study
   - unanticipated problems that might affect continued ethical acceptability of the project
   - major deviations from the approved proposal and/or regulatory guidelines
   - serious adverse events
3. Amendments to the proposal must be approved by the Human Research Ethics Office before they are implemented (except where an amendment is undertaken to eliminate an immediate risk to participants)
4. An annual progress report must be submitted to the Human Research Ethics Office on or before the anniversary of approval and a completion report submitted on completion of the project
5. Personnel working on this project must be adequately qualified by education, training and experience for their role, or supervised
6. Personnel must disclose any actual or potential conflicts of interest, including any financial or other interest or affiliation, that bears on this project
7. Changes to personnel working on this project must be reported to the Human Research Ethics Office
8. Data and primary materials must be retained and stored in accordance with the Western Australian University Sector Disposal Authority (WAUSDA) and the Curtin University Research Data and Primary Materials policy
9. Where practicable, results of the research should be made available to the research participants in a timely and clear manner
10. Unless prohibited by contractual obligations, results of the research should be disseminated in a manner that will allow public scrutiny; the Human Research Ethics Office must be informed of any constraints on publication
11. Ethics approval is dependent upon ongoing compliance of the research with the Australian Code for the Responsible Conduct of Research, the National Statement on Ethical Conduct in Human Research, applicable legal requirements, and with Curtin University policies, procedures and governance requirements
12. The Human Research Ethics Office may conduct audits on a portion of approved projects.

pecial Conditions of Approval
one.

his letter constitutes ethical approval only. This project may not proceed until you have met all of the Curtin University research governance quirements.

ould you have any queries regarding consideration of your project, please contact the Ethics Support Officer for your faculty or the Ethics Office hrec@curtin.edu.au or on 9266 2784.

ours sincerely

r Catherine Gangell
lanager, Research Integrity

Appendix VI: Participant Consent Form

PARTICIPANT CONSENT FORM

HREC Project Number:	RDHU-66-16
Project Title:	Student & Learning Environment Characteristics associated with Student Competency in Introductory Computer Programming
Principal Investigator:	Professor Rob Cavanagh
Student researcher:	Leela Waheed
Version Number:	Version 1
Version Date:	28 April 2016

- I have read, the information statement version listed above and I understand its contents.
- I believe I understand the purpose, extent and possible risks of my involvement in this project.
- I voluntarily consent to take part in this research project.
- I have had an opportunity to ask questions and I am satisfied with the answers I have received.
- I understand that this project has been approved by Curtin University Human Research Ethics Committee and will be carried out in line with the National Statement on Ethical Conduct in Human Research (2007) – updated March 2014.
- I understand I will receive a copy of this Information Statement and Consent Form.

Student ID	
Participant Signature	
Date	

Declaration by researcher: I have supplied an Information Letter and Consent Form to the participant who has signed above, and believe that they understand the purpose, extent and possible risks of their involvement in this project.

Researcher Name	Leela Waheed
Researcher Signature	
Date	

Appendix VII: Participant Information Statement

PARTICIPANT INFORMATION STATEMENT

HREC Project Number:	RDHU-66-16
Project Title:	Student & Learning Environment Characteristics associated with Student Competency in Introductory Computer Programming
Principal Investigator:	Professor Rob Cavanagh Email: R.Cavanagh@exchange.curtin.edu.au Tel: 61 08 9266 2162 Fax: 61 9266 2547 School of Education, Faculty of Humanities, Curtin University, Kent Street, Bentley, Western Australia 6102.
Student researcher:	Leela Waheed **Email:** leela.waheed@postgrad.curtin.edu.au Mobile: (+61) 0401214884 School of Education, Faculty of Humanities, Curtin University, Kent Street, Bentley, Western Australia 6102.
Version Number:	Version 1
Version 1	28 April 2016

Curtin University

Student & Learning Environment Characteristics associated with Student Competency in University Introductory Computer Programming

Dear Student,

This research seeks to examine ***Student & Learning Environment Characteristics associated with Student Competency in University Introductory Computer Programming (CS1)***. The first phase of the study involves construction and testing of an instrument to measure the student competence of CS1 to provide data required for *Phase 2,* which examines the extent of hypothesized associations between Student & Learning Environment Characteristics associated with Student Competency in CS1.

I am a research student at Curtin University at the School of Education of the Faculty of Humanities. The Government of Australia funds this research through Endeavour International Postgraduate Scholarships. My research profile is available at http://hgso.curtin.edu.au/research/

You are invited to take part in this study because the participant criteria I am looking for is first-year undergraduate computer science and IT students who have completed their first course of computer programming. A total of 70 students who have completed University Introductory Programming from two institutes of Maldives, Villa College and The Maldives National University (MNU), and Asia Pacific University (APU) of Malaysia will be invited to participate. As a participant of this study, you will be asked to complete two instruments: the ***CS1 Student Competency Instrument*** (CS1 Instrument) and a ***Survey.*** CS1 instrument is similar to a university computer programming final exam, which will be conducted in an exam setting organized by the researcher in one of the locations of Male'. It will take roughly 60 minutes. The survey asks you to select the options that are true about you from a set of multiple choice questions regarding you and learning environment characteristics. It will take roughly 30 minutes to complete.

There may be no direct benefit to you from participating in this research. However, based on the empirical results of the study recommendations will be made for changing the learning environment, modifying student selection criteria and designing of remedial work which will be useful for students currently enrolled or wishing to enroll in Computer Science programs. This study will also benefit CS1 instructors and curriculum developers, and enable them to reflect on their current teaching practices for making informed decisions about the pedagogical approach, programming language, and IDE for CS1 instruction and curriculum reform. The envisaged linear scale of phase 1, CS1 instrument, will redound to the existing need for instruments for research and pedagogical purposes.

There is no foreseeable risk from this research to you or any other stakeholder.

Please be assured that any information we collect will be treated as confidential and used only in this project unless otherwise specified. The following people will have access to the information we collect in this research: the research team and the Curtin University Ethics Committee.

Electronic data will be password-protected, and hard copy data (including survey and CS1 assessment) will be kept in secure storages. The information we collect in this study will be kept under secure conditions at

Curtin University for seven years after the research has ended and then it will be destroyed. The results of this research may be presented at conferences or published in professional journals.

Taking part in a research project is voluntary. It is your choice to take part or not. You do not have to agree if you do not want to. You can withdraw from the project at any point if you decide not to continue. You do not have to give us a reason as to why you want to withdraw. Please let us know if you want to stop, so we can make sure you are aware of anything that needs to be done so you can withdraw safely. If you chose not to take part or start and then stop the study, it would not affect your relationship with the University, staff or colleagues.

If you decide to take part in this research, we will ask you to sign the consent form at the time of taking CS1 Competency assessment. By signing, it is telling us that you understand what you have read and what has been discussed. Signing the consent indicates that you agree to be in the research project and have your information used as described.

This study has been approved by the Curtin University Human Research Ethics Committee (Approval Number RDHU-66-16). If needed, verification of approval can be obtained either by writing to the Curtin University Human Research Ethics Committee, c/o Office of Research and Development, Curtin University, GPO Box U1987, Perth, WA 6845 or by telephoning 9266 2784 or by emailing hrec@curtin.edu.au.

Kind Regards,

Leela Waheed
Research Student (Doctor of Education),
School of Education, Faculty of Humanities,
Curtin University, Kent Street, Bentley, Western Australia 6102.

Appendix VIII: Sample Request for Approval

29 May 2016

Villa Collge,
Male, Maldives

Dear Dr. Adil,

I am a research student at Curtin University at the school of education of Faculty of Humanities. The Government of Australia funds this research through Endeavour International Postgraduate Scholarships. My research profile is available at http://hgso.curtin.edu.au/research/

As part of my doctoral research, I am seeking participation of undergraduate computer science and IT students in my research that examines the ***Student & Learning Environment Characteristics associated with Student Competency in University Introductory Computer Programming (CS1).***The first phase of the study involves construction and testing of an instrument to measure the student competence of CS1 to provide data required for ***Phase 2*** *by* employing the Rasch Model with Mesick's validity theory and contemporary measurement instrument construction methods. Phase 2 examines the extent of hypothesized associations between student competency in CS1 with student and learning environment characteristics using correlational analysis.

For this research, I am seeking 70 first-year undergraduate computer science and IT students who have completed their first course of university computer programming from Maldives. Participants of this study will be asked to complete two instruments: the ***CS1 Student Competency Instrument*** (CS1 Instrument), and a ***survey instrument.*** The CS1 instrument is similar to a university computer programming final exam, which will be conducted in an exam setting organized by the researcher in one of the locations of Male'. It will take roughly an hour to complete. Students will fill the ***survey instrument*** online at a mutually agreed date. The survey asks students to select the options that are true about them from a set of multiple choice questions regarding student and learning environment characteristics. It will take roughly 45 minutes to complete.

Student consent to participate in this study will be sought at the time of taking the CS1 instrument. The identity of the participants will be anonymous in the reporting of findings in my dissertation and resulting publications. The research data will be kept in a secure place and will not be released to a third party or used for any purpose other than for my research.

I seek your consent to go ahead with data collection at Villa College. The data collection is scheduled to be conducted at a mutually agreed time and date between 7th July 2016 and 15th August 2016. The survey will be hosted online and the link will be shared with participants, at a closer date to the actual survey date.

If you have any questions regarding this research, please contact me by email at leelawaheed@hotmail.com or leela.waheed@postgrad.curtin.edu.au. Alternatively, you could also contact my research supervisor Professor Rob Cavanagh through e-mail at R.Cavanagh@exchange.curtin.edu.au or Tel: 08 9266 2162

This study has been approved by the Curtin University Human Research Ethics Committee (Approval Number: RDHU-66-16).If needed, verification of approval can be obtained either by writing to the Curtin University Human Research Ethics Committee, c/o Office of Research and Development, Curtin University, GPO Box U1987, Perth, WA 6845 or by telephoning 9266 2784 or by emailing hrec@curtin.edu.au.

Kind Regards,

Leela Waheed
Research Student (Doctor of Education),
School of Education, Faculty of Humanities,
Curtin University, Kent Street, Bentley, Western Australia 6102

Appendix IX: Survey Form

INSTRUCTIONS:

The survey will ask you to select an option that is true about you from a set of multiple choice questions regarding student and learning environment characteristics. It will take roughly 15 minutes to complete.

Choose the most appropriate option that describes you. Tick ONLY ONE option unless specified.

1. Gender

☐ Male

☐ Female

2. Secondary School Stream

☐ Science

☐ Commerce

☐ Arts

4. Did you complete Year 12 (A 'level /(STPM) /Equivalent) or Year 10 (O'Level/SPM/Equivalent) Mathematics?

☐ Year 10

☐ Year l2

5. Did you study computer science as a subject in secondary school?

☐ Yes

☐ No

If "yes" did you do computer programming in a programming language?

☐ Yes

☐ No

6. Would you consider you have at least more than 6 months of serious computer programming in a programming language?

☐ Yes

☐ No

6. Which of the following language was used to learn CS1?

☐ C

☐ Java

☐ Python