# THE USE OF COGNITIVE DIAGNOSTIC MODELING IN THE ASSESSMENT OF COMPUTATIONAL THINKING

by

**Tingxuan Li**

**A Dissertation**

*Submitted to the Faculty of Purdue University*

*In Partial Fulfillment of the Requirements for the degree of*

**Doctor of Philosophy**


Department of Educational Studies

West Lafayette, Indiana

August 2019

# THE PURDUE UNIVERSITY GRADUATE SCHOOL
# STATEMENT OF COMMITTEE APPROVAL

Dr. Anne Traynor, Chair

Department of Educational Studies

Dr. Sharon Christ

Department of Human Development and Family Studies

Dr. Jill Newton

Department of Curriculum and Instruction

Dr. Xiaokang Qiu

School of Electrical and Computer Engineering

**Approved by:**

Dr. Richard Olenchak

Head of the Graduate Program

*This dissertation is dedicated to my family and friends who encouraged me and supported me during my PhD studies.*

# ACKNOWLEDGMENTS

With great joy, I give God thanks for Purdue University. I thank God for providing so many incredible opportunities for me at Purdue University during my PhD studies, especially, the opportunity to learn.

In this fantastic learning institution, God has placed many people into my life. Without them, I would never be able to achieve my goals. I thank my advisor, Dr. Traynor, for welcoming me into her research group. I enjoyed every meeting with her. She generously shared her extensive knowledge of educational measurement with me. The knowledge and skills I have acquired will surely carry with me in my future career. I also thank my dissertation committee members: Dr. Sharon Christ, Dr. Jill Newton, and Dr. Xiaokang Qiu. I thank them for their input to make this work so much better.

I thank all the faculty members in the College of Education at Purdue for making my learning experience so pleasant. I would particularly thank the faculty members who lectured me: Dr. Jim Lehman, Dr. Toni Rogat, and Dr. Ala Samarapungavan. Moreover, I thank all the staff members; they work diligently to help students (particularly international students) daily; I would like to especially thank Amanda Goodwin Bowman, Graduate Coordinator in Office of Graduate Studies and Jeannie Navarre, Administrative Assistant in the Department of Educational Studies.

In addition, I would like to thank Dr. Rhonda Phillips, the Dean of Honors College. She has supported me financially in the past. It was my pleasure to be her research assistant. I also would like to thank Dr. Kari Clase who is such a good teacher and collaborator. I thank my lovely peers in Purdue University. They have helped me in every possible way they can: Sarah Sams, Hengrong Du, Alexander Pijanowski, Xuan Wang, and Elizabeth Suazo Flores.

Finally, I am very grateful for my friends in Lafayette Chinese church who constantly pray for me. They are Dr. Jun Chen, Dr. Sai-Kee Yeung, Dr. Haiyan Zhang, Dr. Linda Nie, Dr. Wenbin Yu, Dr. Riyi Shi, and Mr. Lawrence Wang. In addition, I would like to thank my dear American friends who love me so much. They are: Sharon and Rick Fassnacht, April and Ashton Scott, Amy and Mac McKenzie.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABSTRACT

Author: Li, Tingxuan, PhD
Institution: Purdue University
Degree Received: August 2019
Title: The Use of Cognitive Diagnostic Modeling in the Assessment of Computational Thinking
Committee Chair: Anne Traynor

In order to achieve broadening participation in computer science and other careers related to computing, middle school classrooms should provide students opportunities (tasks) to think like a computer scientist. Researchers in computing education promote the idea that programming skill should not be a pre-requisite for students to display computational thinking (CT). Thus, some tasks that aim to deliberately elicit students' CT competency should be stand-alone tasks rather than coding fluency-oriented tasks. Guided by this approach, this assessment design process began by examining national standards in CT. A Q-matrix (i.e., item–attribute alignment table) was then developed and modified using (a) literature in CT, (b) input from subject-matter experts, and (c) cognitive interviews with a small sample of students. After multiple-choice item prototypes were written, pilot-tested, and revised, 15 of them were finally selected to be administered to 564 students in two middle schools in the Mid-western US. Through cognitive diagnostic modeling, the estimation results yielded mastery classifications or subscores that can be used diagnostically by teachers. The results help teachers facilitate students' *mastery orientations*, that is, to address the gap between what students know and what students need to know in order to meet desired learning goals. By equipping teachers with a diagnostic classification based assessment, this research has the capacity to inform instruction which, in turn, will enrich students' learning experience in CT.

# CHAPTER 1. INTRODUCTION

## 1.1 Computational Thinking Overview

Computational thinking (CT) refers to thinking recursively and abstractly as well as reasoning heuristically (Barr, Harrison, & Conery, 2011; Yaşar, 2018). It is a fundamental skill that, like reading or writing skills, *every* child should acquire. Wing (2006) was arguably the first scholar to emphasize the importance of CT for children and adolescents. Nationwide efforts have been made to integrate CT into K-12 education. For example, Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) jointly published an operational definition of CT for K-12 use (2011). The National Research Council recently (NRC, 2010; NRC, 2011) organized two workshops addressing the scope of CT, perspectives on how CT research should be conducted, and possible future directions for CT instruction. Lee et al. (2011) summarized the characteristics of several National Science Foundation-funded CT programs; they found that these programs emphasized CT development for high school and middle school students, with the most used context being computer game creation through programming.

Other than programming instruction, a variety of applications or contexts can also foster CT because CT is a human expression (Deschryver & Yadav, 2015). Seoane-Pardo (2016) introduced a set of CT lesson plans in ethics classrooms in middle schools in Spain. The idea driving the project was that education with moral dilemmas is a useful context for CT development. Students were required to evaluate different scenarios involving unavoidable self-driving car crashes on the Moral Machine Platform developed by the MIT Media Lab. Students used “moral principles to decide how intelligent agents should behave, both in their interactions with humans and other machines and the environment” (p. 41). In the United States, research on K-12 teacher

education has explored how to best prepare future educators in CT awareness. Yadav, Stephenson, and Hong (2017) have repeated that CT is not merely synonymous with using computer technology. Pre-service teachers should learn how to integrate CT across content areas; teaching CT does not necessarily rely on a computer interface or a programming language. A necessary component in CT instruction is teaching students to be conscious about *when* and *how* to apply it (Yadav, Zhou, Mayfield, Hambrusch, & Korb, 2011).

Indeed, CT is a way of thinking about activities, even in the absence of programming skills. This idea is not new in computing education. Scholars in computing education have identified key concepts in CT, such as *abstraction*, *efficiency*, and *heuristics*, which are also common themes in human activities. For example, Lu and Fletcher (2009) provided examples including an assembly line for automobile manufacturing and English grammatical rules. Research conducted by Lewandowski, Bouvier, McCartney, Sanders, and Simon (2010) showed that students employed CT in one daily activity: selling concert tickets. They asked students who had no prior knowledge about programming to figure out solutions. Students' plans for selling the tickets provided information that was useful to assess their CT performance on objectives related to algorithm design.

## 1.2 Overarching Research Goal

Existing instruments to measure students' CT reflect the range of instructional approaches implemented. There is no widely used assessment for CT across classrooms (or research-based programs). In the CT community, different assessment approaches highlighted different aspects of CT. A number of assessments have been developed, including artifacts-based interviews (e.g., Brennan & Resnick, 2012), coding projects (e.g., Grover, 2014), tasks in multiple-choice format (e.g., Yadav, Zhou, Mayfield, Hambrusch, & Korb, 2011), or tasks in open-ended format (e.g.,

Tran, 2019). Most of them are content-specific, in that they require disciplinary knowledge (e.g., Bers, Flannery, Kazakoff, & Sullivan, 2014). Some are curriculum-specific, in that they require programming instruction (e.g., Mouza, Marzocchi, Pan, & Pollock. 2016). A few of them are stand-alone assessments (or at least contain some stand-alone questions, e.g., Gouws, Bradshaw, & Wentworth, 2013). (A comprehensive description of assessment will be provided later in the literature review section.) Among them, either a recommended scoring system was not available, or a single score was used to reflect students' CT competency.

In this research, a new instrument, Computational Thinking Competency Assessment (CTCA), is introduced. After specifying the test design, a set of multiple-choice (MC) items are developed for middle school students to solve. In the CTCA, the assessment design process, the analysis of students' response data, and *subscore* reporting are all tied together by cognitive diagnostic modeling (CDM). CDM is a cognitive-psychometric modeling approach to measure a human performance domain (i.e., ability, trait, or competency). Throughout this research, subscore refers to the profile scores estimated through CDM representing students' mastery status. It gives information to teachers about students' strengths and weaknesses, who can in turn further influence students' learning. In a broader sense, subscore reporting has the capacity to offer feedback to teachers while organizing the ongoing or next lesson.

The main design feature for the CTCA is that it does not depend on any programming language context or specific disciplinary knowledge. So teachers across different classrooms can use it. The content used to develop items is drawn from everyday scenarios or graphs/tables commonly seen by middle school students, "positioning CT as a link to cognitive competencies involved not only in science and engineering but also in everyday life" (Yasar, 2018, p. 33). Yasar further suggested that a cognitive model, such as an *information processing model,* can possibly

be used to identify the characteristics of CT. Other than the information processing model (see Embretson & Wetzel, 1987), another widely used general cognitive model is the *cognitive model of domain mastery*. As Svetina, Gorin, and Tatsuoka (2011) described, "the domain mastery model is typically specified as a list of discrete skills, knowledge, or attributes assessed by the items on the test" (p. 3). For example, the model of domain mastery in *reading comprehension* constitutes multiple discrete attributes including location (locating the information in the text) and vocabulary (deploying his or her vocabulary while reading).

In this research, the assessment design process involves establishing the formal cognitive structure of CT competency, namely, a CT model of domain mastery. Then, the formal cognitive structure and scoring model of the response data are tied together with the CDM approach. The CTCA also possesses the following design features: (1) CT researchers and teachers (practitioners) can use them without cost. (2) The quality of the assessment is evaluated by a series of validity evidence including psychometric properties of students' response data. (3) The scoring system, along with score meaning, is explicitly articulated and presented. After students' response data are collected, in this research, the following overarching questions will be answered:

(1) *What are the characteristics of mastery status in CT competency among middle school students?*

(2) *Do female and male groups have different CT competency?*

## 1.3 General Considerations in Assessment Design

Developing a stand-alone CT assessment does not indicate that CT should only be taught as a stand-alone course. It solely means that CT competency can be displayed without pre-requisites (e.g., programming ability). The tasks designed to elicit CT competency are not related

to any content knowledge (e.g., the disciplinary knowledge in biology).This section is an overview of assessment design in order to answer the question below:

As the literature in instructional practice has suggested, students' CT can be fostered through various context or content knowledge. The reciprocal relationship is also true; that is, CT competency can deepen the understanding in disciplinary knowledge (e.g., biology or physics, e.g., Weintrop et al., 2016). In the CTCA, why the goal is to develop a stand-alone CT assessment rather than a content-specific CT assessment (e.g., programming or biology)?

A stand-alone assessment was chosen because of the following factors: (1) reasonable cost of assessment development, (2) reasonable testing time, (3) relatively precise estimation of students' latent scores, and (4) an explicit interpretation of scoring meaning. These factors are the important design decisions to be made by assessment developers (Gorin & Mislevy, 2013). They were considered in this research. Consider a case where CT instruction has been integrated into a six-week biology lesson plan in a middle school, and the disciplinary knowledge in question relates to *molecules and organisms*. Compared with a stand-alone CT competency assessment, a CT assessment for a substantive content, for example, an *assessment of CT for scientific modeling in molecules and organisms*, is likely to capture multiple constructs. An individual measurement task therefore is expected to measure all of the following areas: (1) CT (e.g., the ability to identify parallel processing), (2) scientific modeling (e.g., the ability to evaluate the evidence used to support a scientific claim), and (3) biology disciplinary knowledge (e.g., the ability to describe that a subsystem is composed of groups of cells interacting with other subsystems in the body). The defined construct may require *complex scenario-based assessment*, which has the following drawbacks, as stated by Gorin and Mislevy (2013):

> "Current research on complex scenario-based assessment suggests that each task could require 30 to 45 minutes, which clearly places a limit on the number of independent tasks that can be administered and used to estimate student abilities. Without more information from a broader range of tasks, the possibility of any psychometric model producing generalizable results is low". (p. 21)

Moreover, the traditional items (tasks) on an assessment, along with existing psychometric models, require an explicit mapping between items and the latent variable (i.e., ability or competency). In contrast, the *complex scenario-based tasks* lack clear boundaries between the constructs assessed (Baker, Martin, & Rossi, 2016). Given current psychometric modeling development, the scoring results generated from these complex assessment fail to offer an explicit score interpretation. However, such interpretation is often what classroom teachers and multiple stakeholders seek.

In a broader sense, scoring system and score interpretation are very important for an assessment because they are closely related to validity. This is found in the most recent *Standards for Educational and Psychological Testing*, jointly published by the American Educational Research Association, American Psychological Association, and National Council on Measurement in Education (hereafter, "the Standards;" AERA, APA, & NCME, 2014). The design methodology of the CTCA (see Chapter 3) will follow the *Standards*. In order to design a coherent assessment, multiple design decisions should be explicitly articulated. Validation, termed *validity by design* by Mislevy (2007), is not mechanically conducting a series of analyses after assessment administration, but rather intentionally structuring design activities "in such a way that validity evidence emerges" (p. 467).

In an assessment design language, the *argument-based approach to validation*, as termed by Kane (2013), refers to "the proposed score interpretations and uses that are validated not the test or test scores" (p. 1). Throughout this research, the validation process of *the proposed score use* and *score interpretation* involves both a systematic assessment design process and parametric psychometric modeling of students' responses on the assessment. The proposed score use for CTCA is that practitioners (e.g., classroom teachers) will be informed about students' CT competency. Students' CT competency may often be of interest to teachers. For example, in a computing course, the instructors wonder, "What relevant knowledge and abilities do students without prior computing instruction ('beginners') bring to their first class?" (Lewandowski et al., 2010, p. 60). Score interpretation for the CTCA will be described later in Chapter 3.

The remainder of this study is outlined as following. In Chapter 2, the literature on CT in K-16 (most of the literature focuses on K-12) education is identified and synthesized regarding two areas: (1) how CT is implemented in teaching and learning; and (2) what kind of CT assessments have been developed. In Chapter 3, the design methodology of the CTCA is presented. Cognitive interviews with 10 middle school students are reported with the purpose that this *item-tryout* procedure may contribute to the validity evidence. In Chapter 4, results from the analysis of both pilot data (*79 students*) and operational data (*564 students*) are reported. The scoring system is also described. In Chapters 5 and 6, discussions and conclusions are drawn in connection with both existing literature and the research questions.

# CHAPTER 2. LITERATURE REVIEW

This chapter consists of three sections synthesizing the literature in CT. The first section summarizes studies that emphasize the role of computing environments (e.g., programming instruction or computer interfaces) in fostering CT. The second section summarizes studies of how non-computing related courses or activities may foster students' CT; studies in various contexts (e.g., science) are presented. The third section presents a set of CT assessments identified in the existing literature.

## 2.1 CT and Digital Artifacts

Many researchers have highlighted the role of programming skills in CT development. Some literature has blurred the line between CT competency and programming skills because these skills or the programming environment context can greatly foster students' CT development. Brennan and Resnick (2012) arguably proposed the first CT framework for young learners. They described CT as having three strands: (1) *computational concepts* such as loops or iteration; (2) *computational practices* such as how young learners use these computational concepts to debug and remix work with block-based language; and (3) *computational perspectives*, that is, how young computational thinkers view the world around them and themselves. Such perspectives possess long-term value for young learners. The K-12 education community expects students to acquire knowledge and skills with long-term value, not only immediate applications to classroom assignments (Czerkawski & Lyman, 2015), and the computing community also aims to achieve this goal.

The CT framework proposed by Brennan and Resnick is promising because it highlights the importance of computational perspectives. The perspectives enable young learners to use

computation as the medium to self-express or to create. The assessment associated with this framework can only be implemented when young learners engage with programming. The specific formats for assessment include project portfolio analysis as well as digital artifact-based interviews. Throughout their study, they used the Scratch (https://scratch.mit.edu/) programming interface; Scratch is an open-source, agent-based modeling platform, and block-based programming language. It allows users to translate blocks of code into storytelling on the screen.

In keeping with the idea that programming environments can benefit CT development, Grover, Pea, and Cooper (2015) outlined that computing education as a new type of discipline-based research in K-12 education, which differs from science, math, or technology education (e.g., Grover & Pea, 2013). They argued that early exposure to programming may enhance students' CT competency, which in turn may further spur their interests to pursue computing-related professions (Cooper, Grover, Guzdial, & Simon, 2014). Grover (2014) developed a computing course for middle school students. The author highlighted that computer science instruction should be effectively structured in a way that students' CT is deliberately fostered. The assessment to measure middle school students' CT in that study will be described in detail later in this chapter.

Consistent with the idea that *early exposure to programming* may enhance students' CT competency, many empirical studies involving programming have been conducted at elementary schools. The underlying measurement principle for this type of research relies heavily on the ability to create digital artifacts (e.g., coding fluency). Bers, Flannery, Kazakoff, and Sullivan (2014) explored how a TangibleK robotics curriculum may help students develop CT. The research participants included 53 students, 23 of whom were enrolled in a religious-based private school, and the remaining students were enrolled in an urban public school. The curriculum covered six lessons of increasing difficulty: (1) the engineering design process; (2) robotics; (3) choosing and

sequencing programming instructions; (4) looping programs (control flow instructions-1); (5) sensors; and (6) branching programs (control flow instructions-2). They found that most children (75%) did well on the first three lessons, but that fewer children (56%) did well on the second half of the lessons. The learning outcomes they measured were related to programming concepts (i.e., debugging, control flow) rather than robotics knowledge (i.e., technological devices' function). Children's work products (e.g., the program or robot made by the child) were rated on a scale from 0-5, with a 0 indicating that children did not attempt the task, and a score of 5 indicating the child had fully and effectively completed the tasks.

Complementary to efforts undertaken in technology classes (which mainly focus on robotics), researchers also have used science and art classrooms to help students' CT development. The learning outcome in such contexts has often been measured by students' understanding of certain programming concepts. Sáez-López, Román-González, and Vázquez-Cano (2016) described the implementation of design-based research (DBR) in science and art classrooms. 107 students in the $5^{th}$ and $6^{th}$ grades participated in the research. This 2-year intervention used Scratch and found that Scratch instruction was able to improve performance on *computational concepts* and *computational perspectives* within the experimental group. The assessment tool used in the research contained five scales. Scale 1 measured students' active learning. Scale 2 measured the contents of art history. Scale 3 measured computational concepts. Scale 4 measured the perceived usefulness of the content to students. Scale 5 measured students' enjoyment. All of these scales were attitudinal or affective scales except Scale 3, which was a competency scale related to CT. The content of the scale includes sequences, loops, conditional statements, parallel execution, coordination, event handling, and keyboard input. The scale development process or sample items was not reported in the publication.

In another application of DBR to CT of elementary students, Jun, Han, and Kim (2016) examined programming instruction using both Scratch and the Creative Computing Guidebook developed by MIT Media Lab. The curriculum they designed for the experimental group included the following content: (a) introduction of Scratch and art projects; (b) storytelling; (c) game programming; and (d) making game projects and reflecting through presentations. They defined CT competencies as (1) *CT problem-solving ability*; (2) *self-efficacy*; (3) *interest*; (4) *self-CT*; and (5) *self-CT: computing perspectives*. The Self-CT test included eight items, where "five items on the programming aspect-understanding codes, construction ability, enthusiasm for programming, and operating principles, three items on the computing aspect" (p. 47). They found the curriculum was an effective approach to improve the learning experience in CT for students. Details of the test content were not provided.

Other than formal instruction in K-12 classrooms, as mentioned above, informal education programs (e.g., after-school programs) have also been designed to develop students' CT using a programming environment. Mouza, Marzocchi, Pan, and Pollock (2016) designed an after-school computer science program for middle school students; they found the program had positive learning outcomes in both programming skill and attitudes toward computing. During the 9-week instruction, they combined college field experience with a partnership of local schools, in which undergraduate students offered one-to-one scaffolding to middle school students. Because CT was characterized as consisting of *computer science* (*CS*) *concepts*, *computational practices*, and *attitudes toward computing*, the learning outcomes were measured by using a Scratch coding project, attitude survey, and a CS concept test containing 10 Scratch coding knowledge items.

Forty-one students received instruction, and completed both the attitude survey and CS concept test. In this test, the first item required students to choose the category for a Scratch block.

For questions 2 to 9, students were required to match a Scratch code to one of the following CS concepts: parallelism, variables, conditionals, loops, and knowing what a block of code does. Question 10 required student to predict the outcome given a block of code. No sample items were provided in their study. Gain scores calculated by using pre-and-post sum scores on the test indicated that students had gained understanding in CS concepts; no statistically significant difference was found between boys and girls on the attitude measure, the CS test, or the Scratch coding project.

Also related to informal education, Fields, Giang, and Kafai (2014) explored an informal online programming community for 2,225 young Scratch users where the median age is 14. Compared with traditional classrooms, an informal learning environment may enhance *computing participation*, which, according to the authors, includes: (1) social participation, such as commenting on a coding project with peers; and (2) personal expression. They defined computing participation as part of CT; therefore, the informal programming environment may be a natural platform for fostering computing participation. After examining online users' programming profiles, they found no association between a respondent's level of participation and the level of programming sophistication. Only a small group of users who were highly engaged with programming tasks tend to use more complex programming concepts such as *variables and operators*. The authors noted that programming is easy for beginners but becomes exponentially difficult later. They pointed out that the evidence of learner engagement with advanced programming tasks is lacking. For example, girls made more comments about playing games rather than discussing programming in the online community. They recommended that future research should develop a mechanism of more meaningful computing participation to engage learners with more sophisticated tasks.

As noted in these research studies, the learning outcomes of CT are most often measured by students' ability to produce digital artifacts or their understanding of programming concepts. Consequently, this indicates that the authors of various studies have implicitly endorsed the idea that *teaching programming is teaching CT*. But not all members of the CT community endorse this idea. Czerkawski and Lyman (2015), on behalf of the editorial board of *Issues and Trends in Educational Technology*, stated that "while CT and computer science are linked, computational thinking is not simply computer science; teaching CT should not be reduced merely to teaching Scratch or Alice" (p. 1). Therefore, in the next section, a different set of studies are presented. The learning outcome of interest is no longer about understanding computing concepts or ability in producing digital artifacts, although some studies may use computer interfaces (e.g., Scratch) as instruction-delivery tools.

## 2.2 CT without Digital Artifacts

In this section, a set of studies are presented. Some studies are position papers, whereas some are empirical studies. The studies presented in the previous section used the ability to produce digital artifacts as the learning outcome of CT. In contrast, the studies presented here stressed the following learning outcomes: (1) the understanding in disciplinary knowledge (e.g., biology) and (2) ability to think computationally in daily-life scenarios.

### Robotic Programming

Chen et al. (2017) developed a robotic programming curriculum for elementary students. They endorsed the idea that teaching CT is more than teaching students to use a set of digital commands to make a robot turn left or right. Students' CT competency is displayed not merely through coding fluency, but in various contexts. By using a pre-test and post-test design, they aimed to assess the effectiveness of the curriculum in enhancing CT for 5$^{th}$ grade students and, in

addition, provided psychometric properties of two different sets of measurement tasks. Based on students' gain scores, they found that students did better in programming robotics tasks than daily-life scenarios tasks and suggested that the future instruction should focus on (1) strengthening the connection between coding examples and daily life examples, and (2) encouraging students to deliberately convert everyday tasks to coding problems.

Similar to the approach adopted by Chen et al. (2017), Berland and Wilensky (2015) studied robotic programming instruction in middle schools. For 78 students in the 8$^{th}$ grade, they explored whether a *physical* robotic-related simulation instructional unit and a *virtual* robotic-related simulation instructional unit may lead to different levels of understanding for middle school students. Both simulations were embedded in the computer-based learning environment (VBOT), a graphical programming language. Learning outcomes were measured by four open-ended items; two human raters graded the questions on a scale of 0-3, each of which measured exactly one construct. The CT question, Question 2, was a stand-alone question that was not contingent on programming or robotics knowledge, which measured CT in a daily-life scenario. Other questions measured complex systems thinking, robotic knowledge (technology content knowledge), which required students to write an essay, and students' programming skill with VBOT in four scenarios. The pre- and post-test results suggested that two instructional units led to similar learning gains for these 4 target constructs. .

In addition, Kazimoglu, Kiernan, Bacon, and Mackinnon (2012) also explicitly articulated that teaching CT should not be reduced to teaching programming concepts in robotics. They designed an educational computer game, *Program Your Robot*, as a framework to foster CT. Their study addressed the question: What tools can make CT accessible to *everyone*? Compared to many computer interfaces designed as *programming tools*, a *CT tool* should allow users to develop

solutions for CS related challenges where users have little or minimal programming backgrounds. In these authors' view, CT consists of cognitive skills such as *creating and applying algorithms for a particular problem* or *detecting logical errors*. These skills are not equal to the ability to use computing concepts to create digital artifacts. However, many existing computer interfaces often make programming skills a prerequisite in order for users to display CT. The authors described the development of the framework and acknowledged that empirical evidence had not been collected. But the framework elaborates a clear distinction between CT competency and programming ability.

**Science Education**

As mentioned above, content knowledge is conceptually separate from CT competency. CT has been integrated not only in computing or technology education, but also in science education. Science is expected to be a natural context to foster CT. The current national K-12 science standards—the New Generation Science Standards (NGSS, 2013)—considers CT a part of *science practices*, because scientists often practice debugging and testing. Such *computational practice* among scientists is captured as an emerging category to inform the new science standards (NGSS Lead States, 2013). Weintrop et al. (2016) provided a framework to define CT in science and math classrooms in high schools. The authors provided a taxonomy of the core areas of CT competencies in math and science, that is, *data practices, modeling and simulation practices, computational problem-solving practices,* and *systems-thinking practices.* Students must possess disciplinary knowledge (for example, in climate change) in order to display CT competencies. The instructional delivery is based on human-computer interactive virtual modules. In physics classes, for example, virtual modules about energy require students to complete a series of activities to build a roller coaster; a set of CT competencies therefore can be elicited or displayed, such as *understanding the relationships within a system*. When students keep track of energy

measurements four times, *data practices* are used. (No empirical evidence appears in the paper because its purpose is to propose a framework).

Along this line, Sengupta, Kinnebrew, Basu, Biswas, and Clark (2013) developed an agent-based computer instructional tool, *Computational Thinking in Simulation and Modelling* (CTSiM). The underlying idea is that a computer environment may foster CT through modeling and simulation while participants simultaneously acquire knowledge in physics and biology. Three modules were developed to provide instruction on kinematics in physics, and ecology in biology: *The Construction World, The Enactment World,* and *The Envisionment World*. The activities embedded in these modules introduced students to basic programming concepts (e.g., conditionals or logical operators). The participants were 24 sixth-grade students in a racially diverse middle school in the US, 15 of whom received one-on-one scaffolding guidance from research members while working on the CTSiM modules; the remaining students (the control group) received regular classroom instruction. To assess the effectiveness of the scaffolded CTSiM modules, content tests for both kinematics and ecology were used to measure students' disciplinary knowledge. The test scores showed learning gains in both ecology and kinematics, particularly in the former.

Subsequently, the same group of authors (i.e., Basu, Biswas, Sengupta, Dickes, Kinnebrew, & Clark, 2016) produced another empirical study using a *learning-by-modeling* paradigm of attaining disciplinary knowledge in middle school science classrooms. Students acquired disciplinary knowledge (e.g., kinematics and ecology) along with domain-general computational concepts (e.g., conditionals or logical operators) in the CTSiM user interface. Their study promoted the idea that a CT-based open-ended learning environment could facilitate students' learning in science when in-person scaffolds were available to students.

**Other Contexts**

The existing literature in robotic programming and science has emphasized students' CT acquisition. But they are not the only contexts identified in the literature emphasizing students' CT acquisition. This section highlights how other contexts are able to foster CT. Sanford and Naidu (2016) argued that spreadsheets are useful in connecting CT in algebra, and considered mathematical modeling as the core of CT. In their later position paper, they provided a few concrete examples in algebra for elementary mathematical concepts, and they detailed examples that are suitable for high school or early college (Sanford & Naidu, 2017). Each solution acts as a template that can be used for solutions of similar problems. By using spreadsheets to manipulate the parameters, students are guided to answer a "what if" question, given the template.

The Scratch programming system is also a commonly used instruction-delivery tool. Wolz, Stone, Pearson, Pulimood, and Switzer (2011) wrote a research report describing how middle-school level language arts content may foster CT using through *poetry projects* and *research report projects*. The tasks required students to write news stories, edit video, and develop animations in Scratch to support their storylines. Students and teachers in a New Jersey middle school were recruited to be part of the integration of CT in journalism; in total, 36 students participated in the project. The measurements used in the study included a computer usage survey, a computer efficacy test, a career decision-making survey, a math efficacy test, and a math interest survey. A story-telling/problem-solving activity was used to measure CT (the details of this activity are unavailable). By using non-parametric statistics, Wolz et al. found the experimental group (i.e., the students who completed the summer school) had no difference from the control group in CT, but they did find that both teachers and students gained positive attitudes about CT and programming by participating in the project.

## 2.3 Existing Assessments in CT

In the previous two sections, the discussion of the literature centered around how CT is displayed and fostered. Namely, multiple meaningful contexts were used to foster students' CT. In this section, CT assessment tools are reviewed in order to further examine the characteristics of CT. Table 1 below shows the identified 8 assessment tools in CT. One additional study, conducted by Korkmaz, Çakir, and Özden (2017), measured undergraduate students' self-reported *perception and confidence level* on five CT competencies: creativity, algorithmic thinking, cooperativity, critical thinking, and problem solving, rather than CT competency itself, so was excluded here.

Table 1 Eight Published Assessment Tools in Computational Thinking

| Study ID | Authors | Participants' population |
|---|---|---|
| 1 | Grover (2014) | Middle school |
| 2 | Yadav, Zhou, Mayfield, Hambrusch, & Korb (2011) | Pre-service teachers |
| 3 | Werner, Denner, Campe, & Kawamoto (2012) | Middle school |
| 4 | Román-González, Moreno-León, & Robles (2017) | 5$^{th}$ grade to 10$^{th}$ grade |
| 5 | AP Computer Science Exam by College Board (2017) | High School |
| 6 | Tran (2019) | 3$^{rd}$ grade |
| 7 | Zhong, Wang, Chen, & Li (2016) | 6$^{th}$ grade |
| 8 | Gouws, Bradshaw, & Wentworth (2013) | First-year college students |

Study 1 was published by Grover (2014). The author designed a computer science course for middle school students and a set of items embedded in the course. One of the instructional objectives for the course was to increase students' CT competency. The items can be summarized as three types. The first type is *CT vocabulary*. Students are required to write down the definitions of the key terms in programming (e.g., loop or algorithm). The second type of item is code execution. Two programming interfaces are used in this context: block-based code and text-based code. The item format includes both open-ended format and multiple-choice format. For example, a block of code is given, but one or more lines are missing from the code. Students are required to

choose one option from the given options to complete the block of code. Another example is that a complete block of code is given, and students are asked to choose one of five options to indicate what the result would be after executing the code. As for the open-ended items, a block of code is given, and students are required to write in plain English what the code does. The third type of item is a stand-alone item where no programming experience is needed. Such items are based on daily-life scenarios (e.g., card game), in which students are required to write down the card order after following a series of instructions.

Study 2 was published by Yadav, Zhou, Mayfield, Hambrusch, and Korb (2011). It described a core course required for undergraduate students majoring in primary or secondary education. A CT module was part of the course and it aimed to introduce (1) computing concepts and (2) integration of CT across various disciplines. The main idea motivating the course design was that future teachers would be aware that CT instruction can be built upon activities that require no computers. The CT concepts covered in the course included: (a) abstraction, (b) logical thinking, (c) algorithms, and (d) debugging. A set of items were developed to align with the instruction. Most items measured teachers' perceptions or attitude towards CT, such as the questions "*In your view, what is computational thinking*?" In addition, some items about CT concepts were based on daily scenarios (e.g., serving pizza). They did not depend on any programming knowledge or content areas.

Study 3 was published by Werner, Denner, Campe, and Kawamoto (2012). It described the details of the *Fairy assessment*, a performance assessment in the 3D *Alice* environment. As part of computer game programming instruction, it was offered either in after-school programs or elective classes. CT competency was defined as including (1) *thinking algorithmically* as well as (2) *making effective use of abstraction and modeling*. *Use-Modify-Create*, a widely used programming

instruction, was used to teach students how to control the characters in *Alice*. The Fairy assessment gives learners incomplete blocks of code (i.e., 3D scenarios) and requires them to implement the solutions to achieve the outcomes. It contains three tasks, each of which is graded by two human raters on a 0-10 scale. The authors also promoted the idea that paired programming (two students coding a project together) is a useful approach to engage students in CT. They acknowledged that the generalizability of the results might be low for the studies that do not use the *Alice* environment, and recommended that the Fairy assessment could be easily adapted by other CT researchers.

Study 4 was published by Román-González, Moreno-León, and Robles (2017). It described the development of a CT test consisting of 28 multiple-choice items for students from Grade 5 to Grade 10. The items were generated from seven CT competencies: *basic directions and sequences*, *loops-repeat times*, *loops-repeat until*, *if-simple conditional*, *if/self-complex conditional*, *while-conditional*, *simple functions*. Students were required to understand block-based programming language in order to solve CT items.

Study 5 was published by the College Board (2017). The AP exam, a national testing program, is a 2-hour long exam that accounts for 60% of the AP Computer Science Principles final score. The item format contains (a) single-select multiple-choice items and (b) multiple-select multiple-choice items (students select two answers from four options). The content of the exam depends on the AP Computer Science curriculum framework, which is organized using seven "Big Ideas": *creativity, abstraction, data and information, algorithms, programming, the Internet,* and *global impact* (College Board, 2017). Items were generated from each Big Idea (p. 84). Moreover, each item measures one of six areas in *computational thinking practices*: (1) connecting computing; (2) creating computational artifacts; (3) abstracting; (4) analyzing problems and artifacts; (5) communicating; and (6) collaborating. Among all the sample items listed on the document, most

items elicit students' programming knowledge (i.e., text-based language) as required by the curriculum framework. Two items are the exception where no programming skills or experience are needed to solve the items; both were generated from (4) analyzing problems and artifacts. Because this exam is a commercial product, it is not freely available for use by teachers.

Study 6 was published by Tran (2019). It describes how third-grade students' CT competency changed after a 10-week computer science curriculum. The curriculum provided instruction in a block-based language. By working with university computer scientists, the author developed a *Computational Thinking Assessment*. This tool has 10 stand-alone items that were based on daily-life scenarios, such as exercising in the gym or the sequence of planting tomatoes. The item formats include (a) true/false selection format, (b) fill-in-the-blank, and (c) single or multiple multiple-choice. 10 items are evenly distributed across five computational concepts addressed in the assessment tool: *sequence*, *algorithm*, *looping*, *debugging*, and *conditionals*. For each item in *sequence*, a set of discrete stages of actions is given (i.e. "wake up" or "drive to school"), students are required to circle the wrong steps in the sequence first and then to rewrite the actions in the correct order.

Study 7 was published by Zhong, Wang, Chen, and Li (2016). It describes the development of the Three-Dimensional Integrated Assessment (TDIA) framework in the 3D *Alice* environment. The assessed dimensions are *computational concepts*, *computational practices*, and *computational perspectives*. The TDIA contains six tasks to measure these dimensions. The tasks are written in two formats: (1) code completion and (2) error correction. For the first format, students are given the incomplete code and the goal. For the second format, a 3D interface is provided with a storytelling scenario (e.g., rabbits in the garden) along with the reflection report. On the report, a selection of errors are listed; for example, "the rabbit's jumping is not looping". Students are

required to fill out the report regarding (a) why the error occurs and (b) the solution to fix the error. In addition, detailed task-specific rubrics were developed. For tasks 1 to 4, the code is judged based on a 0-5 scale. For tasks 5 to 6, the code is judged based on a scale from 0-20. Specific criteria measured by tasks 5 and 6 included content, creativity, artistry, and technology. Based on the students' scores, the findings indicated that students' performance on the two different task formats did not vary much.

Study 8 was published by Gouws, Bradshaw, and Wentworth (2013). It described the development of an assessment tool for first-year college students in an introductory computer science course. The score use had two purposes. The first purpose was that instructors can "assess the raw skills that students possess before they have learned anything as part of the formal academic course" (p. 273). The second purpose was that the same group of students took the test in the third term in college to observe score changes as they progressed through their degree program. In the assessment design process, design decisions were explicitly made before they characterized the content of the test. These decisions were: (a) the test should be a stand-alone test where no programming knowledge is required; (b) testing length should be reasonable, so it is easy to administer the test to classrooms; (c) items should be easy to score (i.e., no rubrics are needed). They defined CT as "how to think like a computer scientist, with a strong emphasis on problem solving" (p. 271).

In their assessment, six areas were measured: (1) Processes and Transformations, (2) Models and Abstractions, (3) Patterns and Algorithms, (4) Tools and Resources, (5) Inference and Logic, (6) Evaluations and Improvements. 25 stand-alone items (e.g., frog jumps) are in the final test form. The item format includes multiple-choice and fill-in-the-blank modules. Five sample items are provided in the paper. Among 83 students, they found that high achievers (top 33%) on

the test also had good performance on the course exam (i.e., programming knowledge test). Although the psychometric properties of items (e.g., parametric modeling) were not reported in their paper, as the pioneering work in stand-alone CT assessment, this test highlighted the clear distinction between programming ability and CT competency

# CHAPTER 3. ASSESSMENT DESIGN METHODOLOGY

In this chapter, the design process for the CTCA is explicitly articulated. Simply put, as Mislevy (2007) has argued, *validity by design* requires that assessment developers think of validating scores not as a process of mechanically conducting a series of analyses after assessment administration, but as several assessment design activities intentionally structured "in such a way that validity evidence emerges" (p. 467).

## 3.1 Domain Analysis

In assessment design methodology, *domain analysis* is a design activity in which assessment developers make decisions on how to obtain substantive information within the domain of interest (Mislevy & Riconscente, 2015). Currently, research on CT is limited, and its definitions are ambiguous and vary considerably from one study to another (Mouza, Marzocchi, Pan, & Pollock, 2016). As Kalelioglu, Gülbahar, and Kukul (2016) reported, "CT literature is at an early stage of maturity, and is far from either explaining what CT is, or how to teach and assess this skill" (p. 583). According to the Computer Science Teachers Association (CSTA) K-12 Computer Science Standards (2011), "developing an approach to computational thinking that is suitable for K-12 students is especially challenging in light of the fact that there is, as yet, no widely agreed upon definition of computational thinking" (p. 10).

Previous research has explored characteristics of CT relating to communication, collaboration, and participation. To the authors of those studies, CT is more than mental operations required to solve problems, it is also about dispositions or practices. In a computer-based simulation model for high school science class, CT practices included collecting data and visualizing data about physics on computer (Weintrop et al, 2016). In contrast, in the CTCA, the

viewpoint that *CT is a thought process* is adopted; dispositions are not considered in this research. Also, as Lee (2016) described, CT is different from *CT practices*. For example, communicating results to peers, gathering data using computational tools (e.g., to copy and paste files on computers), and collaborating on computing activities are considered *CT practices* rather than CT. "While these are…valuable practices in CS education, they are not necessarily part of 'formulating a problem and its solution so that the solution can be carried out a by computer'" (p. 4).

The next design decision is to examine the details regarding *CT is a thought process*. Table 2 (below) summarizes the definitions found in the existing CT literature. It includes (a) the source of the definition and (b) the main aspects emphasized in the definition. The definition proposed by Barr and Stephenson (2011), which was adopted for use in the national standards published by Computer Science Teacher Association (CSTA, 2011). By examining the definitions, a CT definition is proposed and used throughout this research (see below). This definition resembles the one proposed by Barr and Stephenson (2011).

> *CT is a series of mental operations used to solve a problem, so the problem may be automated on a computer. CT competency is not equal to programming competency, although a programming environment may benefit a CT learner in terms of displaying CT competency. Programming knowledge is not a prerequisite for solving CT problems.*

Table 2 Extracted Computational Thinking Definitions

| Source | Definition | Aspects |
|---|---|---|
| Aho, 2011, p. 2 | "We consider computational thinking to be the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms." | (1) mental operations |
| Denning, 2009, p. 28 | A mental orientation to formulating problems as conversions of some input to an output and looking for algorithms to perform the conversions." | (1) mental operations |
| Barr & Stephenson, 2011, p. 51 | "A problem-solving methodology that can be automated and transferred and applied across subject." "An approach to solving problems in a way that can be implemented with a computer." | (1) mental operations (2) not content specific (3) the solutions can be carried out by a computer |
| Lee, 2016, p. 3 | "CT refers to the human ability to formulate problems so that their solutions can be represented as computational steps or algorithms to be carried out by a computer." | (1) mental operations (2) the solutions can be carried out by a computer |
| Voogt, Fisser, Good, Mishra, & Yadav, 2015, p. 716 | "There is a history in both research and popular press of the use programming as a way to develop thinking skills. CT focuses on developing these thinking skills while within subjects beyond computer science. CT does not necessarily require the use of programming nor are CT scholars making the claim that programming has to be the context in which these skills are developed." | (1) mental operations (2) not content specific |
| Lu & Fletcher, 2009, p. 260 | "We argue that in the absence of programming, teaching CT should focus on establishing vocabularies and symbols that can be used to annotate and describe computation and abstraction, suggest information and execution, and provide notation around which mental models of processes can be built." | (1) mental operations (2) not content specific |
| CSTA & ISTE, 2011, p. 57 | "(a) Formulating problems in a way that enables us to use a computer and other tools to help solve them; (b) Logically organizing and analyzing data; (c) Representing data through abstractions such as models and simulations; (d) Automating solutions through algorithmic thinking (a series of ordered steps); (e) Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources; Generalizing and transferring this problem solving process to a wide variety of problems." | (1) mental operations (2) not content specific |

*Note*: CSTA stands for the Computer Science Teachers Association; ISTE stands for the International Society for Technology in Education.

Based on the definition proposed in this research, the ability to troubleshoot a computer hardware problem, for example, does require a series of mental operations, but this problem solving process *per se* fails to be automated. Thus, this ability is not CT. Likewise, many other types of problem solving are not CT competency. For example, the Programme for International Student Assessment (PISA), a large-scale testing program across multiple countries, defined *problem-solving competency* and created an assessment to measure it (OECD, 2003; OECD, 2014). The resulting definition is "an individual's capacity to engage in cognitive processing to understand and resolve problem situations where a method of solution is not immediately obvious" (OECD, 2014, p. 30). The problem-solving competency required in PISA highlights students' ability for detecting faults.

Also, mathematical problem-solving ability and CT competency share an emphasis on problem solving, but mathematics ability is not the same as CT competency. A tremendous amount of research has examined the characteristics of mathematical thinking. Researchers have explored the importance for mathematical thinking of *problem representation skill*, which Brenner et al (1997) defined as "constructing and using mathematical representations in words, graphs, tables, and equations" (p. 666). The same authors designed a ten-item test to measure students' ability to convert a word problem to a table, equation, and graph. Blanton et al. (2015), who developed an algebra intervention for third-grade students, defined that representational skill as the "ability to navigate between different representations in a math expression" (p. 68). STEM experts, though, describe CT as different from engineering design thinking or mathematical thinking (NRC, 2010).

The purpose of presenting both PISA Problem Solving Competency test and the empirical research in math is to highlight CT as a unique way of thinking. After describing *what CT is* as well as *what CT is not*, the subsequent design activity for CTCA is to extract the key testable

elements in CT competency. Assessment developers must collaborate with subject-matter experts (SMEs) to identify the information that can be used for assessment design (Mislevy & Riconscente, 2015). Throughout this research, in various design activities, multiple panel reviews were conducted to collect feedback or input from SMEs for item revisions. For each panel review, the meeting with SMEs was in person and lasted about 60-70 minutes. The decisions from each review took place by consensus. The review was structured as a meeting with each SME. The prompts were verbally given to the SMEs. For example, “Does this item confuse you” “Does this item require you to think computationally?” The prompts were not scripted. If possible, the SMEs also provided the feedback for revisions.

**The SMEs in CT**

Throughout this research, the expert panel consisted of three members in CT. A short biography of each member is shown below:

(1) Mr. Troy Fisher earned a BS degree from Indiana State University in Industrial Technology Education. His first job was as an educator at Attica High School in 1985. He completed an MS degree in Industrial Technology Education in 1989, and a second MS degree in Mechanical Computer-aided design (CAD) Drafting in 1991. In 2016, he took a position at Lafayette Tecumseh Junior High School as the inaugural Project Lead The Way (PLTW) Gateway instructor and robotics club sponsor. He received a PLTW Gateway Certificate in the summer of 2017 from Purdue University. He has taught technology, engineering, and integration of CT at Grade 7 and Grade 8.

(2) Dr. Ali Alshammari earned his PhD in 2018 in Learning Design and Technology at Purdue University. His research interests include instructional design theory for serious games and computer science pedagogy. Throughout his years at Purdue University, he worked on research projects in the Games Innovation Laboratory, as well as teaching undergraduate courses in computing.

(3) Mr. Shengwei An is a doctoral student in the Department of Computer Science at Purdue University. He is a recipient of the Purdue Ross Fellowship, awarded to Purdue students for high academic achievement. He received his bachelor's and Master's degrees in Computer Science at Nanjing University, China. His research interests include dynamic software update and robust machine learning.

**The Performance Objectives Document**

National (or state) standards reflect what students should know and be able to do at certain grade(s), but are not a curriculum (Polikoff, 2017). Patel and Herick (2016) gave the specific steps to ensure instructional practices, assessment, and the standards are aligned. They used a sixth grade math standard selected from Common Core State Standards as the starting point to develop the classroom-based assessment.

In the CTCA, in order to identify learning goals for students at middle school grades, the assessment design started by examining national standards. Computer Science Teacher Association (CSTA) K-12 Computer Science Standards (2011) is used. I chose the CSTA standards as a basis for the domain for several reasons. (1) The CSTA standards seemed popular among teachers (Chen, Shen, Barth-Cohen, Jiang, Huang, & Eltoukhy, 2017). Empirical research on CT conducted in K-12 has often used CSTA as the starting point in characterizing the curriculum or assessment (e.g., Chen et al., 2017; Tran, 2019). (2) The standards contain three

different expectation levels: Level 1 is for elementary school grades, Level 2 for middle school grades, and Level 3 for high school grades.

Other national standards (e.g., K-12 Computer Science Framework developed by K-12 Computer Science Framework Steering Committee, 2016) offer general statements about CT, which only contain one level of expectation. Also, International Society for Technology in Education (ISTE, 2016) standards and CSTA standards share many commonalities in CT learning goals. But ISTE standards heavily emphasize the use of technological devices for students to display CT. In this research, due to the design decision made in the previous stage, the standpoint held in this research is that technological devices are unnecessary for students to display their CT competency. Thus, the ISTE standards were not chosen as the performance objective document.

The CSTA standards have five strands: (1) *computational thinking*; (2) *collaboration*; (3) *computing practice*; (4) *computers and communication devices*; and (5) *community, global, and ethical impact*. The CTCA is developed to measure Level 2 in *computational thinking* strand, which contains 15 standard statements. Among 15 statements, nine of them require either (1) the use of a computer (e.g., "*Analyze the degree to which a computer model accurately represents the real world*"), or (2) the integration of CT in other content areas (e.g., "*Interact with content-specific models and simulations to support learning and research*"). These nine statements are not considered in this assessment design because they are not aligned with the design decisions made previously in the CTCA.

The next design decision is: Among the remaining six standard statements, which ones should be selected to guide the next design activities (e.g., item development) for the CTCA? It is possible to use all six standard statements. However, more standards statements covered on the assessment will require more items. Considering the tradeoff between reasonable testing time and

a comprehensive test (i.e., more standards being covered), a literature review was hence conducted. The purpose of conducting the literature review is to ensure the *content coverage*, namely, to select standard statements that can reasonably represent the domain of CT in the design process of the CTCA. The summary of the literature review results is presented in Figure 1.The results showed that three of the CSTA standard statements (see below) were often emphasized in existing CT literature:

> Statement 1: use visual representations of problem state, structures, and data.
>
> Statement 2: describe and analyze a sequence of instructions being followed.
>
> Statement 3: examine connections between elements of mathematics and computer science including binary numbers, logic, sets, and functions. (p. 16)

These three selected statements are further structured into *claims*, as shown below. In an achievement test design, a claim about score interpretation typically reflects students' standing with respect to specific learning standards that have been put forth (Kane, 2013). For example, one possible claim in math is that "Students are able to identify when two expressions are equivalent" (e.g., $x + x + x = 3x$). The score interpretation on the math assessment, therefore, involves claims about students having some fluency on *algebraic manipulation*. In an argument-based approach to validation (Kane), what is valued in instruction (or standards), what the items are designed to elicit, and what their scores on the assessment represent are all tied together by such claims.

In this research, the main claims about score interpretation hence refer to expected CT competency for students in middle schools:

*Claim 1: Students are able to use visual representations of problem state, structures, and data.*

*Claim 2: Students are able to describe and analyze a sequence of instructions being followed.*

*Claim 3: Students are able to examine connections between elements of mathematics and computer science including binary numbers, logic, sets, and functions.*

## 3.2 Domain Modeling

In the previous design activity, decisions were made in order to explicitly determine (a) what CT is and what CT is not; and (b) the claims used as the proposed score interpretation for the CTCA. In this section, a *domain modeling* design activity is carried out. The purpose of *domain modeling* is to examine how each claim is unfolded. At the end of this design activity, fine-grained testable elements (categories) are extracted from the claims. The existing literature is used in this design activity. Figure 1 is a summary of the literature. Each testable element was then treated as a latent attribute representing a type of CT competency, labeled below as A1, A2, and A3.

**A1:** Students are able to identify the underlying corresponding pattern (e.g., trend or relation) in a given stimulus material. The material can include graphs, letters of the alphabet, or maps.

**A2:** Students are able to execute steps in an algorithm. That is, a series of ordered steps is given in an algorithm in order to generate its output, in which none of the steps can be skipped.

**A3:** Students are able to evaluate variables in an algorithm by examining the pre-defined conditions.

The first round of panel review was conducted. The SMEs reviewed the summary of the literature review (i.e., the literature presented in Figure 1) and the three selected standard statements. They agreed that the selected standard statements can represent the domain of CT for the sake of content coverage. Namely, three claims can be reasonably used to represent what students can do and know in terms of CT competency. The SMEs also agreed that the extracted testable elements can reflect how claims are unfolded in terms of developing an assessment.

| Source | Wording | Other Similar Literature | Claim | Testable Element |
|---|---|---|---|---|
| Gouws, Bradshaw, & Wentworth (2013, p. 272) | "The ability to recognize patterns and work with repetitive structures such as loops, recursion or functions is an invaluable skill for computer scientists. This class also includes the ability to think algorithmically and the practical skills of recognizing classes of problems, generalizing solutions, capitalizing on repetitive structures and considering boundary cases of a problem." | • Brennan & Resnick (2012)<br>• Román-González, Pérez-González, & Jiménez-Fernández (2017) | 1 | Identify patterns |
| Chen et al., (2017, p. 164) | "A Conceptualizing and generating solutions through algorithms (a series of ordered steps)." | • Grover (2014)<br>• Wolz, Stone, Pearson, Pulimood, & Switzer (2011) | 2 | Follow steps |
| Bers, Flannery, Kazakoff, & Sullivan (2014, p. 150) | "While there are currently no curriculum frameworks explicitly addressing control flow, these activities connect to mathematics, by reinforcing number sense and estimation, or to natural science, by comparing human and animal sensory functions with robot sensors. Children are also able to compare and contrast repeating or looping programs to patterns, cyclical events in the natural world, and calendar time." | • College Board (2017)<br>• Lewandowski, Bouvier, McCartney, Sanders, & Simon (2010) | 3 | Evaluate variables |

Figure 1 The literature used to identify claims and testable elements

## 3.3 Item Prototype Writing

Next, item prototypes are developed based on the testable elements. A few design decisions are made in this design activity. First, in what assessment mode, the questions should be presented to students. In daily instructional practice, teachers can use (1) observation and questioning strategies, or (2) paper-and-pencil test to measure CT competency. In the CTCA design, the mode is paper-and-pencil based. The reason to choose this is simply because students' responses can be easily stored which in turn the larger scale of data can be collected. The answer a student writes down or the notes he/she takes on the answer sheet while solving CT problems can all be considered as the artifacts he/she produced.

The next design decision is about the item format on the CTCA. Both open-ended (OE) items and multiple-choice (MC) items are reasonable formats. The decision made in this design

decision that all the items should be MC items. The rationale is found in the measurement literature. MC items can be easily and quickly scored (e.g., Haladyna, Downing, & Rodriguez, 2002). So, choosing MC items over OE items in this research is due to the fact that the scoring procedure will not take considerable time from teachers, the end-users of the assessment. Some sources have suggested that OE items can elicit higher-order thinking more than MC items can. However, as Attali, Laitusis, and Stone (2016) pointed out, "there are many reasons to believe that open-ended (OE) and multiple-choice (MC) items elicit different cognitive demands of students. However, empirical evidence that supports this view is lacking" (p. 787). The authors used math questions on a computer-based assessment where immediate feedback (i.e., right versus wrong) was available after each response was given by a student. The student was required to provide an answer until he/she reached the correct answer. This method is termed *answer-until-correct*. They found that the difference between MC items and OE items mainly relates to test-takers' effortful engagement, rather than a different level of cognitive demand being captured. Specifically, students had more effort invested in the revisions for OE questions. "OE questions may provide a more conducive context for this effort by forcing the student to generate the response instead of selecting one" (p. 799).

Hohensinn and Kubinger (2011) provided empirical evidence that no item format-specific dimension was found in students' response data, indicating both OE items and MC items measured one targeted domain. Such a finding is not new. Decades ago, the findings in measurement literature have indicated that both OE and MC functioned similarly in terms of reflecting students' latent ability. By using the GRE Quantitative Reasoning test, Bridgeman (1993) has compared the OE and MC item formats and found both "formats appeared to be comparable. Both formats ranked the relative abilities of students in the same order" (p. 25).

In order to develop quality items, the item development for the CTCA follows the guidelines provided by Haladyna, Downing, and Rodriguez (2002). As the authors underlined, "despite the increased emphasis on performance testing, the MC format continues to play an important role in classroom and large-scale assessments, and this importance is evident in all 27 textbooks on educational measurement reviewed for this study" (p. 310). Their guidelines detail agreed best practices in MC item writing, for example, (1) readability of item stems indicates that language should be simple; (2) no opinion-based or trick items should be used; and (3) the option "All-of-the-above" or "None-of-the above" should not be used.

At the end of this design activity, thirty-four item prototypes were developed. Each item was designed to measure one or more latent attributes. The CTCA design methodology is in Figure 2 below. The assessment design started from the definition of CT, and three selected claims to represent CT competency. The brackets indicate the resources used in various design activities, for which moving to the right is synonymous with moving from *general* to *specific*. Three fine-grained categories extracted from the literature were used to represent how claims are elaborated. Thus, the formal cognitive structure of CT is a list that contains these three discrete attributes (categories). In the third and fourth layers, the scoring model (parametric psychometric modeling) and CT competency are tied together by a "Q-matrix," an alignment table between items and attributes. A Q-matrix is necessary for the specification of a cognitive diagnostic model to connect students' item responses to the posited attributes of a domain (Tatsuoka, 1990). By convention, if an attribute is relevant to an item, 1 is used to indicate the relevance. Otherwise, 0 is used to indicate irrelevance. In other words, there is no quantity (e.g., 0, 1, 2,…) to indicate to what degree an attribute is relevant to each item.

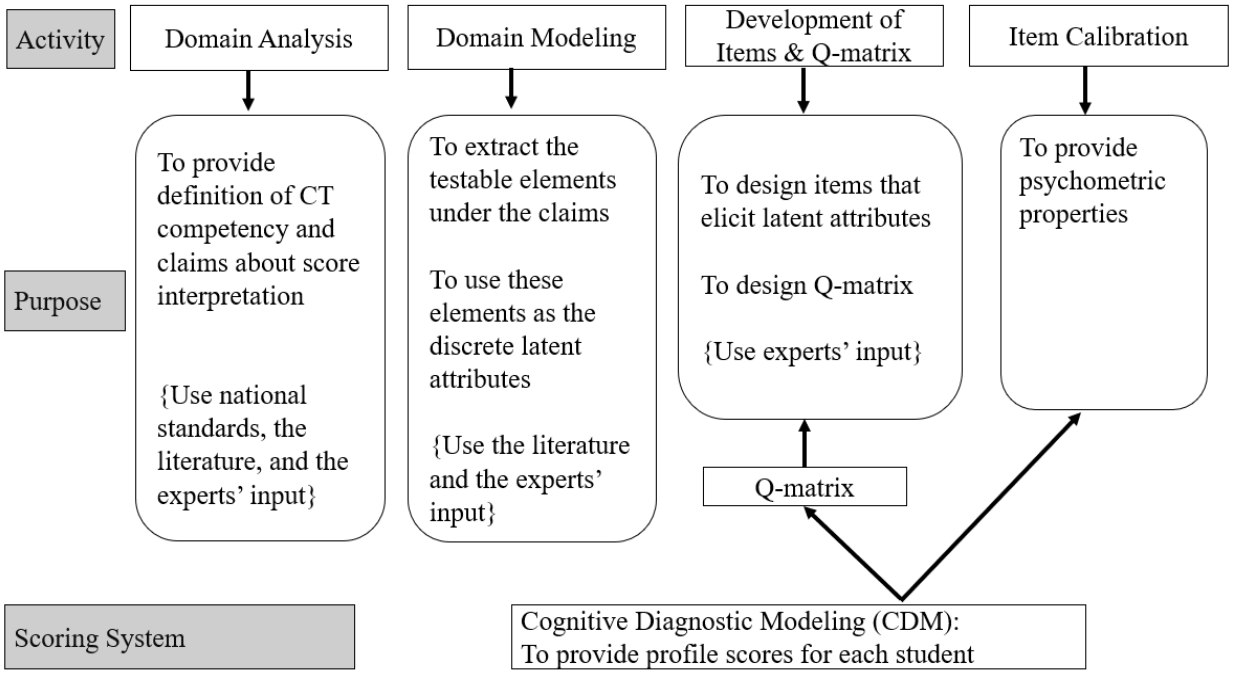



Figure 2 CTCA design methodology

A second round of panel review was conducted in this design stage. By examining the item prototypes and the CTCA design process (Figure 2) along with the draft Q-matrix, the SMEs were required to judge (1) the quality of Q-matrix, that is, the alignment between items and attributes, and (2) the quality of item prototypes. Table 3 below shows the panel's feedback on how item prototypes should be revised. Category 1 indicates that some item prototypes should be deleted. Category 2 indicates that some should be re-structured or re-written. Five items are in Category 1; ten items are in Category 2.

Table 3 Panel Decision on Item Prototypes Revision

| Category | Problem | Action needed |
|---|---|---|
| 1 | Items fail to elicit CT competency. Very little room to improve. | Delete |
| 2 | Re-writing/reformulating the questions is needed, although the design rationale is prominent. | Modify |

For demonstration purposes, Table 4 below contains specific comments provided by the SMEs in terms of Category 1 and Category 2. Category 1 refers to items that fail to possess clear design rationale for CT. Category 2 refers to the items that possess a promising CT-related design rationale, but need to be better orchestrated. After receiving the feedback from the SMEs, the item prototypes were revised. Twenty-nine item prototypes were assembled and used in the next design activity, the cognitive lab protocol.

## 3.4 Cognitive Lab Protocol

A cognitive lab protocol can provide empirical evidence for item revisions (Winter, Kopriva, Chen, & Emick, 2006). Namely, verbal data collected during interviews, along with notes taken by students during the interviews can be used to guide item revisions. As Johnstone, Thompson, Bottsford-Miller, and Thurlow (2008) outlined,

> "Statistics can highlight *which* items may be problematic on a particular assessment, but provide little insight into *why* such items may be problematic. Think aloud techniques are an approach that allows assessment professionals to understand the thinking and problem-solving approaches students use when attempting test items. Insights gained from such approaches provide insights into how the language, presentation, or tasks associated with an item may distract, confuse, or appropriately challenge students." (p. 32)

Table 4 Specific Comments Given by the Subject-matter Experts

| | |
|---|---|
| A list has 3 words and 3 numbers: **{Cat, Tree, Building, 6, 9, 14}**<br>Taking the actions below step-by-step. What does the new list look like?<br><br>Step 1. If the 1st item has fewer letters than the 3rd item, switch two smallest numbers.<br>Step 2. If the 3rd item has more letters than the 2nd item, switch these two items<br>Step 3. If the 2nd item has more letters than the 1st item, no action is needed<br>Step 4. End<br><br>A. {Tree, Cat, Building, 14, 6, 9}<br>B. {Cat, Building, Tree, 6, 9, 14}<br>C. {Building, Tree, Cat, 14, 6, 9}<br>D. {Cat, Building, Tree, 9, 6, 14} | The design rationale is prominent. But it is rarely the case that sorting algorithm considers the number and letters simultaneously. |
| Here is a story. Each number is an event.<br>Please use only FIVE events in a correct order.<br><br>1. Mike wanted to buy a bike.<br>2. Mike got on a bus.<br>3. Mike did not find a green color bike.<br>4. Mike arrived at a store. He will buy a bike if he can find a green color bike and the price is less than 90 dollars. Otherwise, he will leave the store.<br>5. Mike went to a different store.<br>6. Mike bought a bike and went back home.<br><br>A. 24165 B. 12346 C. 41325 D. 12435 | This is more like daily sequential events. It does require students to think logically. But thinking logically is not equal to think computationally. This is more like a generic problem solving question. This item should be deleted. |

The *Standards* (AERA, APA, & NCME, 2014) explicitly state that validation processes should include *evidence of response processes* (e.g., verbal data, students' artifacts). Subsequently, this piece of validity evidence enables assessment developers to capture the extent to which they can adequately model students' response processes. The degree of certainty for this question can help CTCA preserve the proposed scoring interpretation.

**Participants**

The students who participated in this research were drawn from the intended test-taker population for the CTCA - students in the middle school grades. With permission from Purdue University's Institutional Review Board (IRB), recruitment flyers were posted across Purdue University West Lafayette campus. The IRB permission document is attached as Appendix A at the end of this research. Ten students voluntarily participated in the cognitive interviews. Table 5 (below) contains the demographic information. In the first column, pseudonyms are used to label the students. All of them speak English as their first language. Five of them are in 7th grade; five of them are in 8th grade. In terms of self-identified race/ethnicity, seven of them identified as Asian, one as Kosovar, one as Brazilian, and one as White.

Table 5 Demographic Information for 10 Subjects

| Name | Gender | Self-reported Race/ethnicity | Grade | First language |
|---|---|---|---|---|
| Bryan | M | Asian | 8th | English |
| Brandon | M | Brazilian | 8th | English |
| Bruce | M | White | 7th | English |
| Bruno | M | Asian | 7th | English |
| Bennet | M | Asian | 7th | English |
| Bob | M | Kosovo | 8th | English |
| Gwen | F | Asian | 8th | English |
| Grace | F | Asian | 8th | English |
| Brayden | M | Asian | 7th | English |
| Ben | M | Asian | 7th | English |

**Procedures**

Each individual interview session was video recorded to ensure that problem-solving steps were captured in detail. These sessions were about 70 to 90 minutes in duration. The data were collected "in real time" by asking students to talk aloud while he/she was solving the problems; if students stopped verbalizing their thoughts for about four or five seconds, the researcher reminded

them to "keep talking." Follow-up questions were unscripted, which is a common practice for cognitive lab protocols (e.g., Nitko, 2004). A sample follow-up question is that "Was there anything that confused you?" Before the interview session started, the instructions and examples were provided to the students so that he/she could gain familiarity with the cognitive interview format (see the material on Table 6 below). The researcher (i.e., the author of this research) explained to the student the following details: (1) the task that he/she was required to complete, and (2) his/her performance on the test was not being judged.

Table 6 *Instruction and Practice Examples*

You need to solve some questions. This is not a math test. No tricky questions. You can use a pencil to take notes on the test. Whatever you verbally tell me, is very important to me. I want to capture everything you say, so the recording device will be used. I do NOT judge how well you do on the test because getting the problems right or wrong is not the purpose of this research. And your performance on the test will not be graded. The purpose of this research is to capture your thought process. So please tell me everything going on your mind while you solve these problems. If you stop talking for like 4 or 5 seconds, I will remind you to "keep talking". I understand you are not used to this sort of activity. You always keep silence when you take a test or solve problems, right? So, let's practice two examples.

*Example 1*: 1+8÷4×3-2=?

*Example 2*:
A list has 3 words and 3 numbers: **{Cat, Tree, Building, 6, 9, 14}**
Taking the actions below step-by-step. What does the new list look like?

Step 1. If the 1st item has fewer letters than the 3rd item, switch two smallest numbers.
Step 2. If the 3rd item has more letters than the 2nd item, switch these two items
Step 3. If the 2nd item has more letters than the 1st item, no action is needed
Step 4. End

A. {Tree, Cat, Building, 14, 6, 9}
B. {Cat, Building, Tree, 6, 9, 14}
C. {Building, Tree, Cat, 14, 6, 9}
D. {Cat, Building, Tree, 9, 6, 14}

**Results**

The results based on the verbal data and the artifacts provided by the students during the interviews showed that test-takers' response processes were adequately modeled by most of the item prototypes. Most of the prototypes were able to elicit students' CT competency. However, the results also indicated that nine item prototypes should be revised because they had problems in one or more of three aspects: *Aspect 1* involved reducing unnecessary information in item stems. *Aspect 2* involved deleting problematic items. *Aspect 3* involved restructuring the item stems. Table 7 below is a sample question, with artifacts are provided by three students. The artifacts show the item can elicit the intended attributes, because students consistently followed the intended step-to-step procedures to reach the correct answer.

Table 7 Digital Copies of Artifacts for a Sample Question (a)

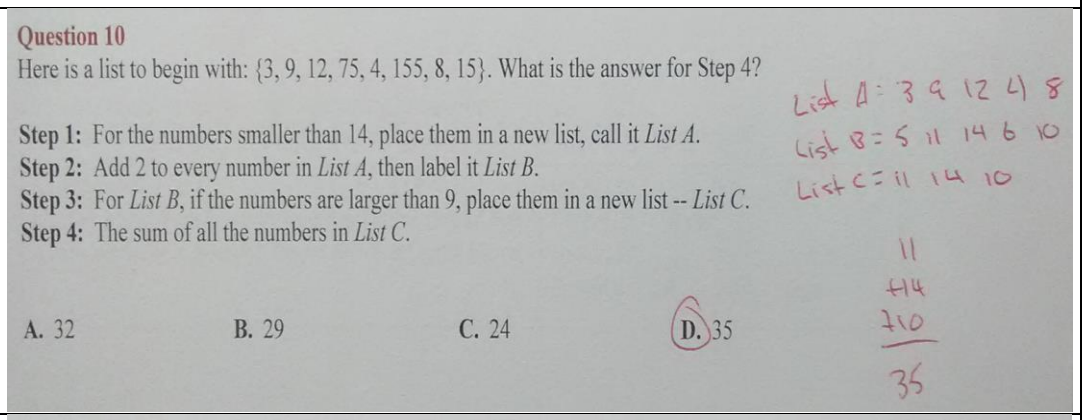
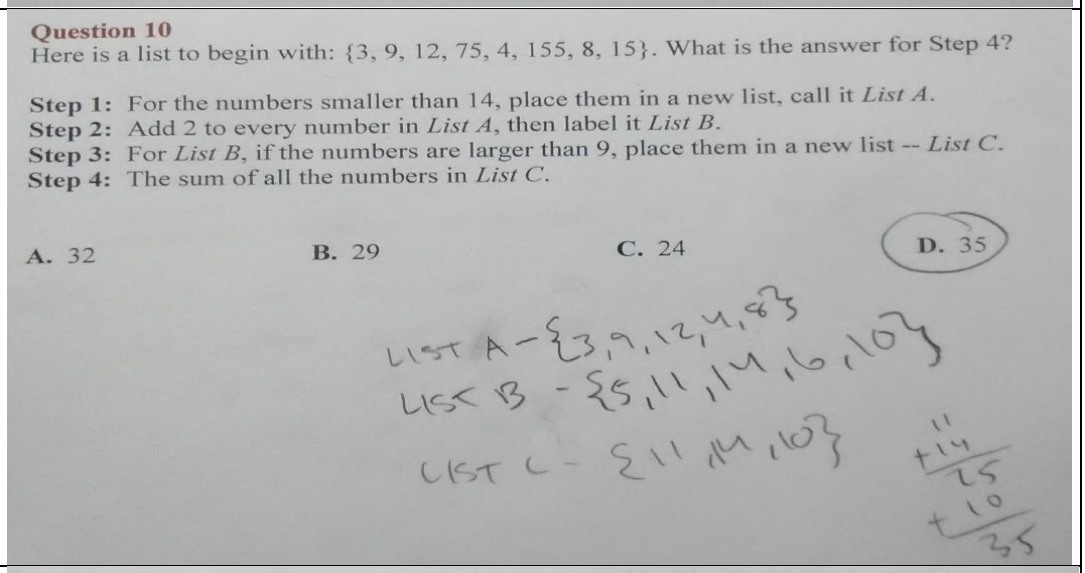
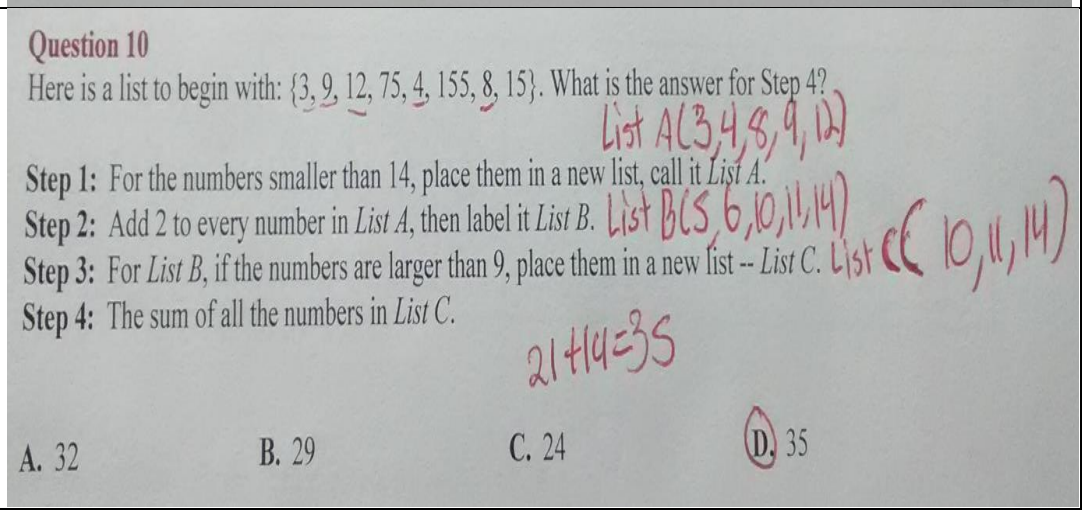
| Subject | Artifact |
|---|---|
| Bruce |  |
| Gwen |  |
| Ben |  |

Table 8 is an example where the item is designed to elicit students' CT. It requires students to recognize the corresponding pattern horizontally, then, moving down to the next step. No step can be skipped in order to find the corresponding pattern for the final step.

Table 8 *Digital Copies of Artifacts for a Sample Question (b)*

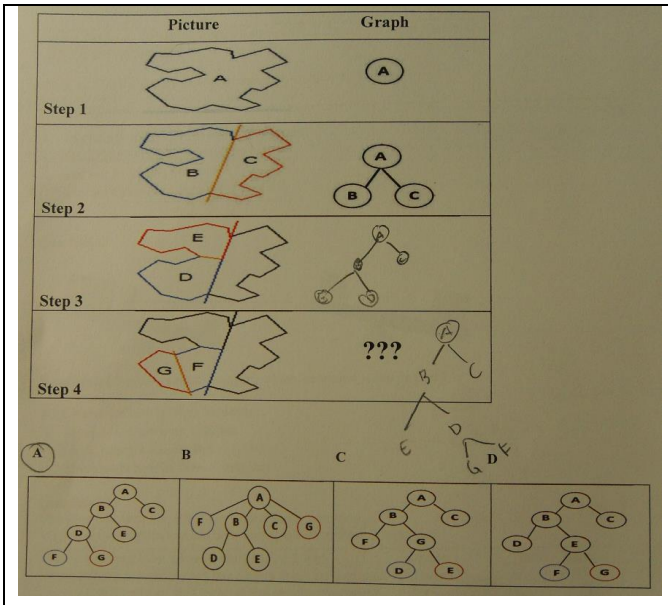
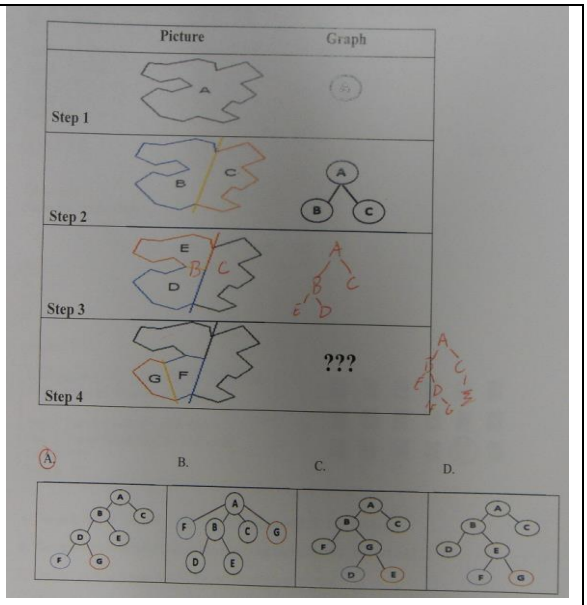
| Bruno | Bryan |
| --- | --- |
|  |  |



Table 9 below shows an item revision in *Aspect 1*. The item stem contained too much information, so the revised version only contained the reduced information. Participants also commented, for example:

"Round 1 and round 2 are not necessary"

"What do round 1 and round 2 do for solving the problem?"

Therefore, in the revised form, the item stem was shortened. Future test-takers can use the time allotted to solve the problem rather than to comprehend the information provided in the item stem.

Table 9 Another Sample Item in Aspect 1

**Item used during the interview**

Maria draws one card at a time in a game. If the first 4 cards she draws meet the following requirement she will receive 2 points. Please fill out the table.

*The first card she draws must be a yellow card.*
*The second card she draws must be a blue card.*
*The third card she draws must NOT be a blue card.*
*The fourth card she draws must NOT be yellow card.*

| | Is the first card yellow? | Is the second card blue? | Is the third card blue? | Is the fourth card yellow? | She receives 2 points |
|---|---|---|---|---|---|
| Round1 | Yes | Yes | No | Yes | False |
| Round2 | No | Yes | Yes | No | False |
| Round3 | Yes | Yes |  | No | True |
| Round4 | Yes | No | No | No |  |

A.  = No  = False

B.  = No  = True

C.  = Yes  = False

D.  = Yes  = Ture

**Revised Item**

Maria draws one card at a time in a game. She will receive 2 bonus points if the first 4 cards meet the following requirements:

*The first card she draws must be a yellow card.*
*The second card she draws must be a blue card.*
*The third card she draws must NOT be a blue card.*
*The fourth card she draws must NOT be yellow card.*

She plays 2 rounds. What do green and purple triangles represent?

| | Is the first card yellow? | Is the second card blue? | Is the third card blue? | Is the fourth card yellow? | Result |
|---|---|---|---|---|---|
| Round1 | Yes | Yes |  | No | She receives 2 points |
| Round2 | Yes | No | No | No |  |

**A.**  =No  = She receives 2 points

**B.**  = No  = She does Not receive 2 points

**C.**  =Yes  = She receives 2 points

**D.**  = Yes  = She does Not receive 2 points

Table 10 shows an example in *Aspect 2* where a distractor has been revised. In a multiple-choice (MC) item, at least two options are given to students. One option is *the answer key*; all other options are *distractors*. Table 10 contains the artifact of Bob, who erred because he skipped one comparison in sorting. After Bob realized his answer was not one of the options, he solved the problem again and reached the correct answer. For *Aspect 3*, two types of items are deleted: (1) ones which students found confusing; and (2) ones for which the problem-solving process used by the students was different than the hypothesized process in the CTCA.

Table 10 *An Example in Aspect 2*

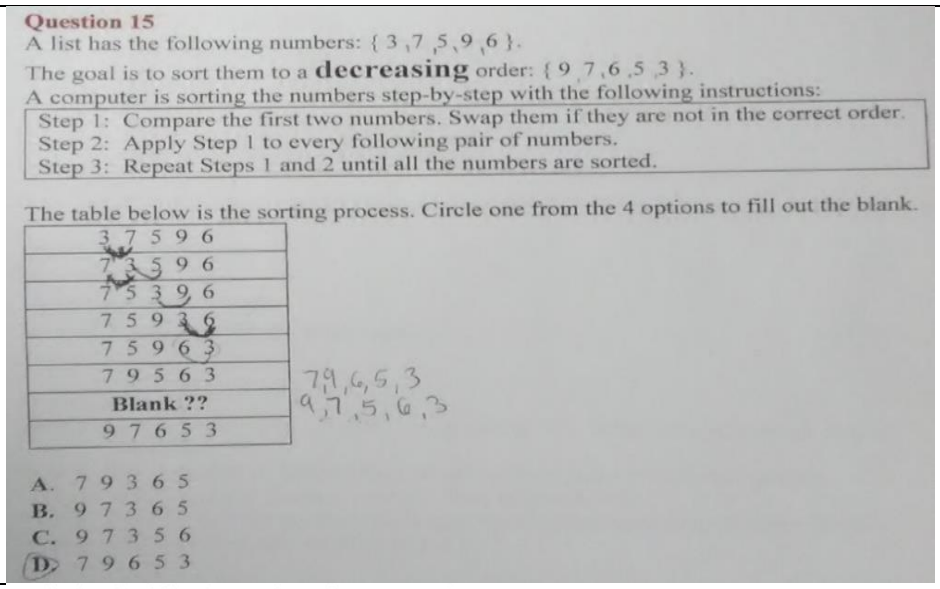
<table>
<tr>
<td>Bob's artifact</td>
<td> 
</td>
</tr>
<tr>
<td>Item has been revised based on Bob's mistake</td>
<td>A list has the following numbers: { 3 7 5 9 6 }.<br>The goal is to sort them to a decreasing order: { 9 7 6 5 3 }.<br>This is the instructions:<br>Step 1: Compare the first two numbers. Swap them if they are not in the correct order.<br>Step 2: Apply Step 1 to every following pair of numbers.<br>Step 3: Repeat Steps 1 and 2 until all the numbers are sorted.<br><br>The table below is the sorting process. Circle one from the 4 options to fill out the blank.<br>3 7 5 9 6<br>7 3 5 9 6<br>7 5 3 9 6<br>7 5 9 3 6<br>7 5 9 6 3<br>7 9 5 6 3<br>Blank ??<br>9 7 6 5 3<br><br>A. 7 9 3 6 5 B. 9 7 3 6 5 C. 9 7 5 6 3 D. 7 9 6 5 3</td>
</tr>
</table>

Table 11 below is an example of an item that has been deleted. Based on the cognitive interview results, the students were confused by this item. They made the comments such as “Which way is the car turning?”

Table 11  Deleted Item: Sample A

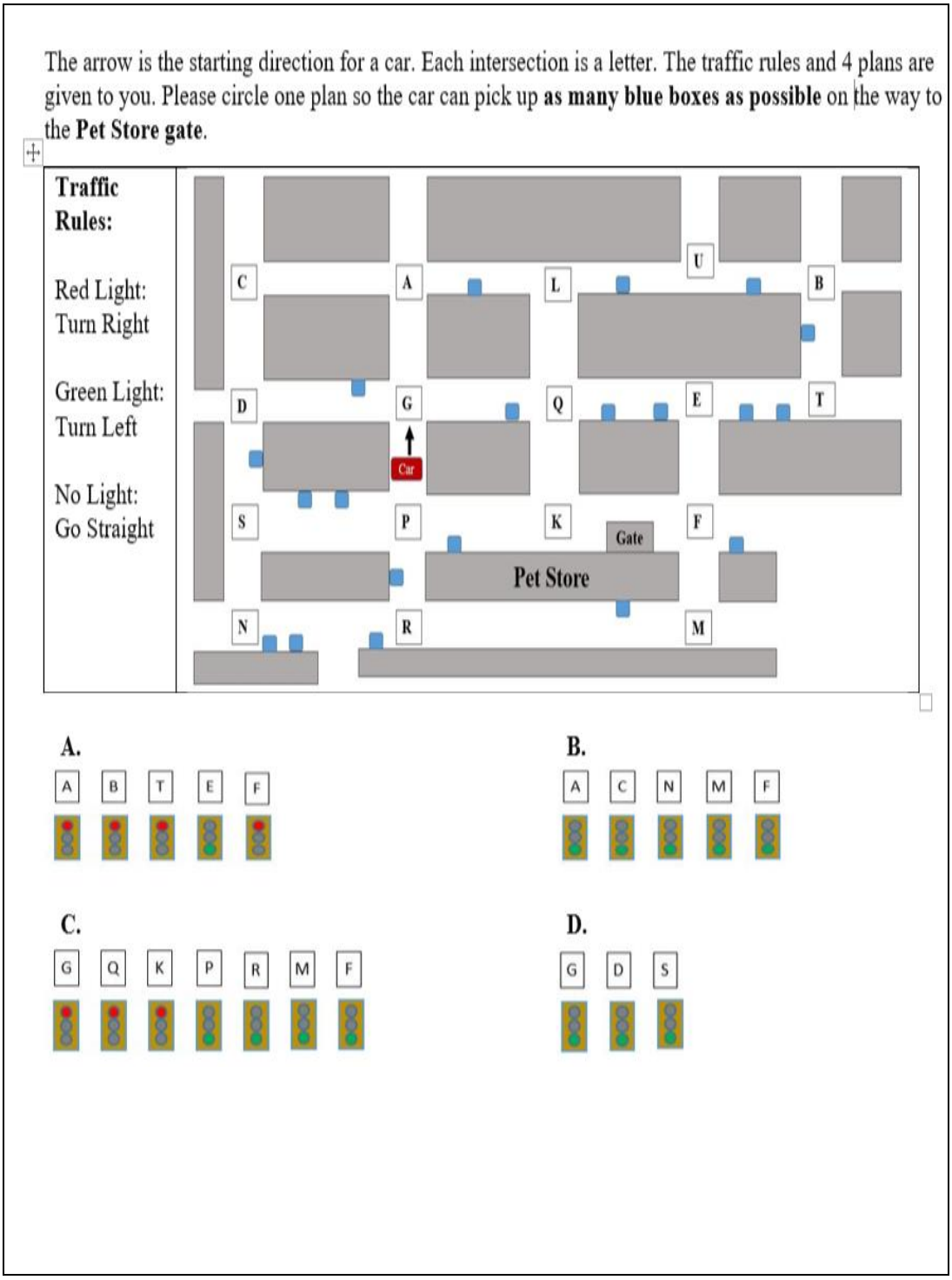


Table 12 below is an item that has been deleted. The design rationale for this item is that students will examine how 1 is changing row by row. The empirical evidence based on students' verbal data showed the students used a short-cut to solve the item. Namely, they only examine vertically on the very right column (see the red color circle). They noticed the number is changing based on 0,1,0,1 ordering. So the next element should end with 0.

Table 12 Deleted Item: Sample B

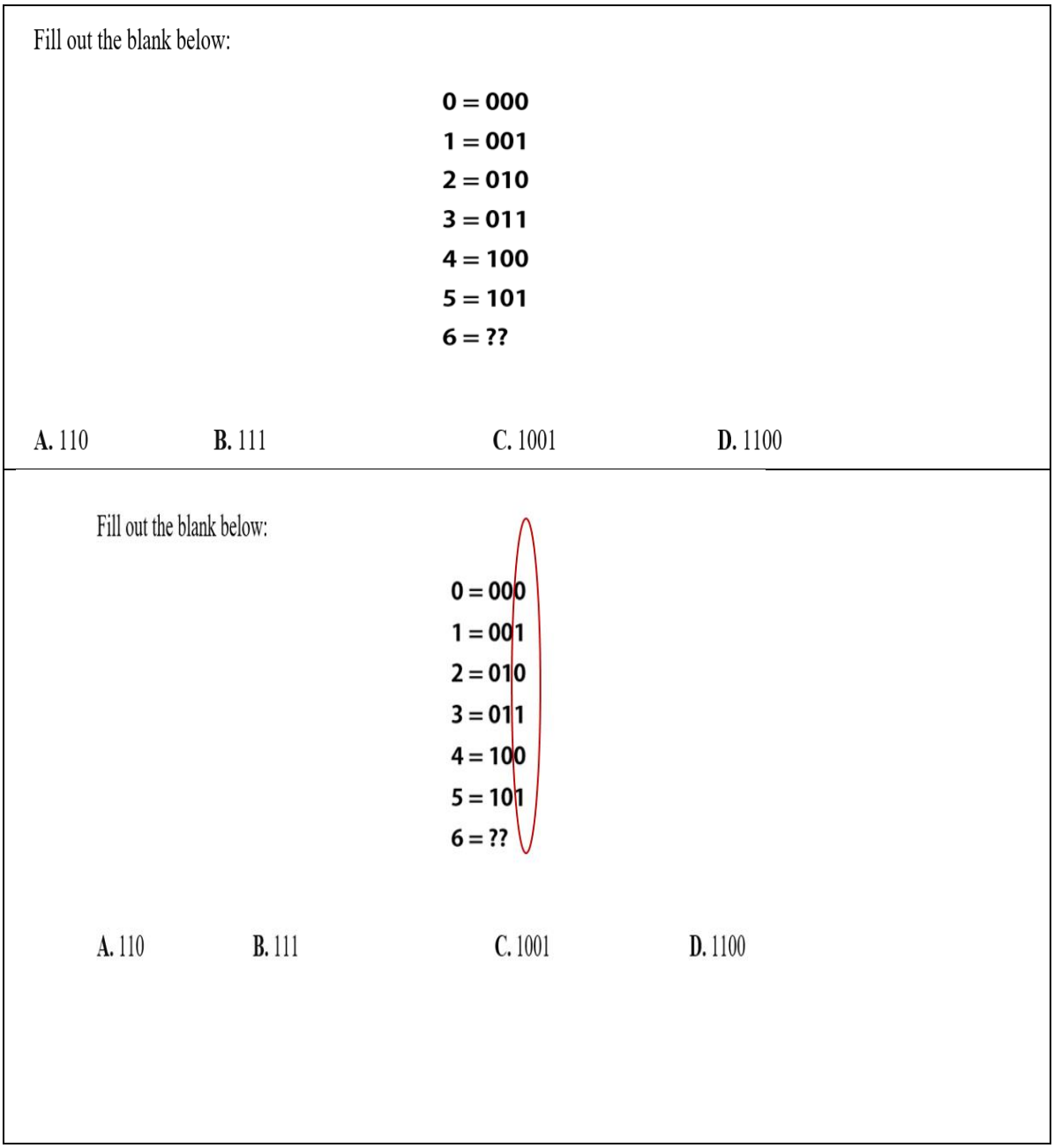
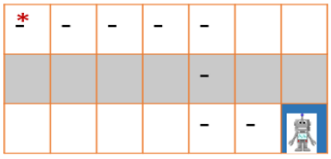


Table 13 below shows an example of *Aspect 4*. The item used during the interviews did not specify at which cell the robot would stop. Students tended to guess where the robot stops. In the revised version, the starting and stopping cells are explicitly given on the map.

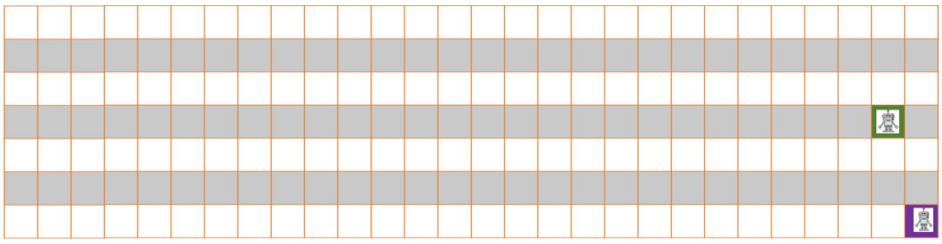
Table 13 Sample Item in Aspect 4

**Item used during the cognitive interview**

It takes a robot **one second** to travel **one cell.** A robot departs from blue cell with the command **L2 U2 L4**.
That means the robot will travel 2 cells left, 2 cells up, and 4 cells left, see the map on right. At the 8th second, the robot will arrive at the red star cell.



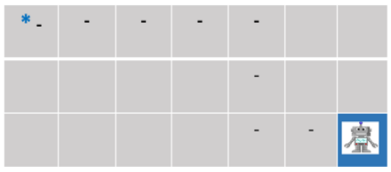
On the map below, one robot departs from green cell with the command **L5 U1 L5 U1.**
Another robot departs from purple cell with the command **L3 U2 L4 L3 U2 L4**.
**They depart at same time**, at which second, they will arrive at the **same cell**?



A. **14th** B. **10th** C. **18th** D. **Never**

**Revised Item**

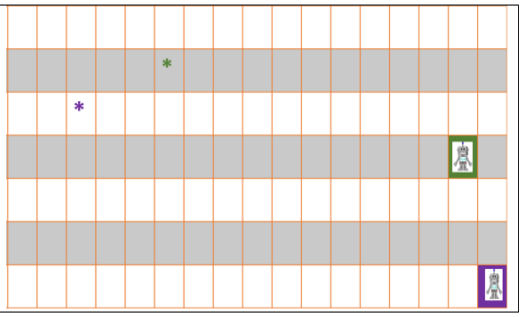
It takes a robot **one second** to travel **one cell.** A robot departs from blue cell with the command **L2 U2 L4**.
That means the robot will travel 2 cells left, 2 cells up, and 4 cells left, see the map on right. At the **8th** second, the robot will arrive at the blue star cell.



One robot departs from green cell with the command **L5 U1 L5 U1.** It will arrive at the green star cell.

Another robot departs from purple cell with the command **L3 U2 L4 L3 U2 L4**. It will arrive at the purple star cell.

**They depart at same time**, at which second, they will meet at the **same cell**?



A. **14th** B. **10th** C. **18th** D. **Never**

The item prototypes revision was completed based on the results of the cognitive interviews. Then, a third round of panel reviews was conducted. The SMEs received the following material: (1) twenty-seven revised items; (2) the definitions of three latent attributes representing CT competency proposed previously in this research; and (3) the alignment table (Q-matrix) between items and latent attributes. *Latent*, in a sense, refers to a variable that cannot be measured directly. The table showed which items were designed to elicit each attribute; some items targeted multiple attributes. The SMEs were asked to judge (a) whether the wording of items was precise; (b) whether the answer key for each item was accurate and absolute; and (c) the alignment between items and the attributes. Finally, the SMEs reached a decision: no further revision was needed. The CTCA specification seemed reasonable, and the items seemed to possess desirable alignment with the Q-matrix. The final design of the CTCA yielded 27 MC items, which were ready to be tried out in a middle-school classroom.

# CHAPTER 4. DATA ANALYSIS RESULTS

In this chapter, two rounds of data collection are documented. The first round of data collection, with a sample size of 79 students, was a pilot test. The psychometric properties (e.g., statistical indices) were reported. The second round of data collection was conducted on a much larger scale, with a sample size of 564 students in two different school districts. Both data collections were conducted with permission from Purdue University's Institutional Review Board (IRB). The IRB permission document is attached as Appendix A at the end of this document. Through cognitive diagnostic modeling (CDM), subscore reporting will be reported. Other psychometric properties from CDM are also reported.

## 4.1 Pilot Test

### Data Collection Context

The CTCA assessment contained 27 multiple-choice items that were pilot tested. The purpose of the pilot test was to empirically determine: (1) the testing time; (2) any non-functioning options across all items; and (3) the presence of bias in any item. The assessment was administered to 79 seventh-grade students in a Financial Literacy classroom at a middle school in the Midwestern US. The sample contained 44 male and 35 female students. The classroom teacher monitored the pilot test.

### Results

According to the SMEs, the hypothesized testing time was 45 minutes for 27 items. Based on verbal feedback provided by the classroom teacher, this testing time was reasonable. A 'non-functioning option' means no students chose that distractor in a given item. By examining the pilot data, all distractors appeared to be functioning well; namely, every distractor on the CTCA was

chosen by a certain number of students, meaning no further revision was needed. The total score was the observed sum score for each student. Figure 3 below shows the distribution of total scores across 79 students. Table 14 shows the item analysis statistics. The indices include the proportion correct for each item and the item-total score correlation, represented by a point biserial correlation coefficient. This set of indices provides an initial quantitative analysis of the items. It seemed the items are useful, and they are not too easy or too difficult. Among all the items, no negative item-total correlation coefficient has been found. A negative coefficient would be a warning message that responses on a particular item are inconsistent with the averaged behavior of the other items. A package – psych, written by Revelle (2010) in R software (R Core Team, 2013) was used for conducting the analysis.

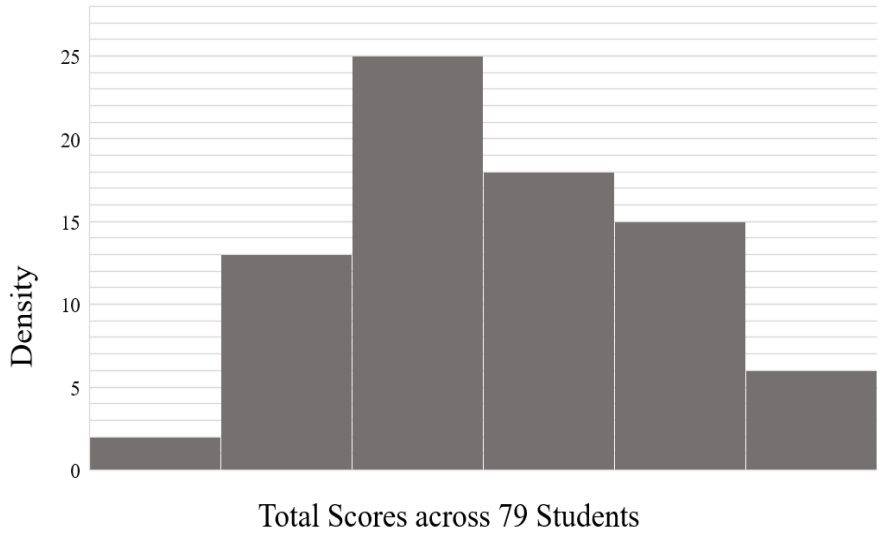



Figure 3 Distribution of total scores

A *t*-test was conducted using the vart.test function in R software to compare the mean of total scores for the female group and the male group, with a test for homogeneity of variance

conducted beforehand. According to Levene's test, the null hypothesis is that the variance is equal across the two groups. The results show the null hypothesis should be retained. Then, the *t*-test is conducted. The results show that the difference of total scores between the two groups is not statistically significant: female (M=13.0, SD=5.44), and male (M=14.5, SD=4.70),

$t(77) = 1.38$, $p = 0.17$

Table 14 Item Analysis Statistics

| Item ID | Item-Total Correlation | Proportion Correct |
|---|---|---|
| 1 | 0.25 | 0.63 |
| 2 | 0.49 | 0.29 |
| 3 | 0.61 | 0.62 |
| 4 | 0.37 | 0.32 |
| 5 | 0.52 | 0.71 |
| 6 | 0.39 | 0.14 |
| 7 | 0.45 | 0.76 |
| 8 | 0.38 | 0.48 |
| 9 | 0.15 | 0.62 |
| 10 | 0.13 | 0.28 |
| 11 | 0.21 | 0.53 |
| 12 | 0.48 | 0.42 |
| 13 | 0.43 | 0.62 |
| 14 | 0.40 | 0.39 |
| 15 | 0.53 | 0.35 |
| 16 | 0.47 | 0.78 |
| 17 | 0.30 | 0.27 |
| 18 | 0.64 | 0.47 |
| 19 | 0.46 | 0.63 |
| 20 | 0.42 | 0.52 |
| 21 | 0.62 | 0.42 |
| 22 | 0.42 | 0.38 |
| 23 | 0.23 | 0.85 |
| 24 | 0.40 | 0.32 |
| 25 | 0.33 | 0.90 |
| 26 | 0.33 | 0.44 |
| 27 | 0.46 | 0.71 |

Furthermore, in order to examine whether any item showed bias towards either gender group, a differential item functioning (DIF) procedure was used. The DIF procedure involves statistically identifying the content of some items that is not central to the measured attribute and that may be less familiar to a particular group of students. Consequently, students with the same level of competency might receive different scores because of their group membership. Thus, the presence of DIF on any item may suggest that the item should be revised or deleted before future

use of the instrument (Walker, 2011). In the CTCA, a package in R software (psychotree, written by Zeileis, Strobl, Wickelmaier, & Kopf, 2012) was used to conduct the DIF procedure. This package employs a tree-based method with the recursive partitioning approach. This method provides a global test for DIF as well as an overall model fit test for the Rasch model. Unlike the parametric regression approach to detect DIF, in this method, the guide to interpretation suggests that if there is more than one terminal node in the tree, DIF may exist on a particular item. For this sample in the pilot test, only one terminal node was found. Thus, no DIF was found in any of the items.

**Design Decision from Pilot Test**

As mentioned above, the testing time for these 27 items was about 40-45 minutes. It seemed necessary to reduce the number of items, so that the assessment could easily be completed within about half of a typical class period (i.e., less than 25 minutes). It is common to shorten a classroom-based assessment (e.g., Gogol et al., 2014) to make it more suitable for practical use. A panel review was conducted, and the Q-matrix was provided to SMEs to review. The SMEs made recommendations about reducing the number of items in the assessment. Finally, 15 items were selected because SMEs judged that these items possessed the strongest alignment to the testable elements. In other words, what the items were designed to elicit (shown in the Q-matrix) reflected what SMEs thought they were measuring. Figure 4 below is the Q-matrix associated with these 15 items. Each cell is marked as 1 if the attribute was measured by that item, and 0 otherwise. These 15 items are attached as Appendix B.

| Item ID | A1 | A2 | A3 | *Note:* **A1:** Students are able to identify the underlying corresponding pattern (e.g., trend or relation), given material. The material can be graphs, letters, or maps.<br><br>**A2:** Students are able to execute steps in an algorithm. That is, a series of ordered steps is given in an algorithm, in order to generate its output, none of the steps can be skipped.<br><br>**A3:** Students are able to evaluate variables in an algorithm by examining the pre-defined conditions. |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | |
| 2 | 1 | 0 | 0 | |
| 3 | 0 | 1 | 1 | |
| 4 | 0 | 0 | 1 | |
| 5 | 0 | 1 | 1 | |
| 6 | 1 | 0 | 1 | |
| 7 | 1 | 1 | 0 | |
| 8 | 0 | 1 | 1 | |
| 9 | 1 | 1 | 1 | |
| 10 | 1 | 1 | 0 | |
| 11 | 1 | 0 | 0 | |
| 12 | 0 | 1 | 1 | |
| 13 | 1 | 1 | 0 | |
| 14 | 0 | 1 | 1 | |
| 15 | 1 | 1 | 0 | |

Figure 4 Q-matrix associated with the 15 items

## 4.2 Operational Testing

### Data Collection Context

The CTCA assessment contains 15 MC items, delivered to students in a pencil-and-paper format. To prevent fatigue or item order effects from influencing validation of the Q-matrix, six test forms with different item orders were produced. A sample of students from two middle schools in two different school districts in the Midwestern US completed the items on the CTCA. The percentages of students in the two schools who had free or reduced lunch plans were 75% and 50%. Six teachers used about 25 minutes of their regular class time to administer the CTCA items. The subjects they teach include math (3), technology (1), English (1), and computer science (1). Because teachers were monitoring their students' test-taking as well as collecting students' answer sheets in person, the data collection process was constructed to avoid missing data. If a teacher

found that an answer sheet contained missing items, the teacher asked the corresponding student to complete the item (or items) and re-submit.

In this research context, the six participating teachers reached a consensus that if students completed the test within five minutes, it indicated the students were not making serious effort, so the response data were not meaningful. To avoid gathering rapid guessing response data, teachers discarded any answer sheets submitted in less than five minutes. In total, 67 answer sheets were discarded. Furthermore, 19 students had an Individualized Education Program (IEP), so assistant teachers read the test aloud to them. Finally, this data collection procedure produced response data from 564 students as well as two variables provided by classroom teachers: (1) gender, a categorical variable taking on two values, and (2) current classroom academic percentage grade, a continuous variable on a scale from 1-100.

**Scoring System**

Once data were collected, they were entered into spreadsheet software on a computer. The scoring system was that a student received a score of 1 if he/she answered the item correctly, and a score of 0 otherwise. Item calibration and score report generation require psychometric modeling, in this case CDM. In this section, a short review of CDM psychometric models is presented.

A traditional (one-dimensional) model in item response theory (IRT) locates a student's ability on a latent continuous scale. Consider a two-parameter logistic (2PL) model below in Equation 1. Suppose that an assessment contains *J* items, where *j = 1, 2, … J*. The response data produced by student *i* on *j* items is denoted as $y_{ij}$ , where $y_{ij}$ = 1 is the scoring system for answering the item correctly, and $y_{ij}$ = 0 for answering incorrectly. The following parameters are estimated in the model: the student's latent variable, that is, the *latent variable score*, denoted as $\theta_i$; the item difficulty parameter, $\lambda_{0k}$; and the item discrimination parameter, $\lambda_1$.

$$P(y_{ij} = 1|\theta, \lambda_{0k}, \lambda_1) = \frac{exp[1.7\ \lambda_1(\theta - \lambda_{0k})]}{1+exp[1.7\ \lambda_1(\theta - \lambda_{0k})]} \quad (1)$$

A traditional IRT model estimates a single quantity associated with each student to represent a student's ability score. In contrast, in CDM, a vector is estimated to represent the profile scores for a student. Specifically, the CDM uses a vector of 0 or 1 to express a student's mastery status on each of a set of latent attributes. It is often termed as *a profile of mastered attributes* (Gierl, Cui, & Zhou, 2009). Recall that, in the previous chapter, the proposed purpose for CTCA is that teachers gain information about students' CT competency. Thus, in CDM, the estimated profile scores as subscores are reported along with the observed total score for individual students.

An attribute in CDM is a generic term; it simply describes an identified thinking skill or knowledge state in educational assessment. In a simple algebra example, described below in Table 15, these attributes are *addition, subtraction, multiplication,* and *division*. The implementation of CDM requires the implementation of a Q-matrix (Tatsuoka, 1990). A Q-matrix has *J* rows and *K* columns. The *jk* element is unity (i.e., $q_{ik} = 1$) if the *k*th attribute is necessary to be mastered for answering *j*th item correctly. By the time that students' response data are ready to be estimated, the Q-matrix should be relatively sound. Table 15 is an illustration example from the mathematics domain where the relevant Q-matrix is the formal cognitive model to describe the qualitative relationship among items. It specifies what attributes are needed for solving each item, because not all items measure all attributes in CDM (Rupp & Templin, 2008).

Table 15 A Toy Example in Math: Q-matrix in CDM

| Item ID \ Attribute | | Addition | Subtraction | Multiply | Division |
|---|---|---|---|---|---|
| 001 | 6+2-3=? | 1 | 1 | 0 | 0 |
| 002 | 10/4=? | 0 | 0 | 0 | 1 |
| 003 | (6+3)*(1/4)+2=? | 1 | 0 | 1 | 1 |

In CDM terminology, each assessment is designed to measure a particular number of attributes. The number of attributes measured by an assessment is denoted as *K*, where *k* = 1, 2, …*K*. Compared with one quantity, $\theta_i$, estimated from an IRT model (Equation 1), the vector $\alpha_i$ is estimated in CDM, because the student's score now is indexed with *k* attributes, where $\alpha_i = \{\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik}\}'$. Consider the illustration example in Table 15, where the number of latent attributes that are being estimated is four. Thus, all possible combinations of attribute patterns are 16 ($2^4 = 16$). In CDM, the number of attribute patterns is finite. Each pattern is sometimes referred to as the latent group. In a broader sense, CDM is a restricted latent class model (Haertel, 1989). Correspondingly, the estimated *profile of scores* for a student who took the test are expressed in Table 16 below, to illustrate the goal of subscore reporting. The information conveyed from Table 16 serves as diagnostic information about this student's mastery status on four pre-defined attributes, revealing information about his/her strengths and weaknesses. The scores suggest the student will need to practice *multiplication* and *division* for future learning. Teachers may use such information to adjust their ongoing instruction. The capacity for providing richer information about students' latent competency is the main advantage in the use of CDM in educational assessment. A few models in CDM are introduced below.

Table 16 A Student's Score File in Cognitive Diagnostic Modeling

| Student ID | Addition | Subtraction | Multiplication | Division |
|---|---|---|---|---|
| 001 | 1 | 1 | 0 | 0 |

The *deterministic inputs, noisy "and" gate* (DINA; Junker & Sijtsma, 2001) model is a conjunctive model, meaning that students must possess all the required attributes in an item *j*, in order to maximize their probability of getting a correct answer on item *j*. A student's *latent response* $\eta_{ij}$ is a deterministic term through equation $\eta_{ij} = \prod_{k=1}^{K} \alpha_{ik}{}^{q_{ik}}$. If $\eta_{ij}$=1, it means that student *i* possess all the required attributes for item *j*. If $\eta_{ij}$=0, student *i* has one or more attributes missing for successfully solving item *j*. Table 15 is an illustration of three items. For item 1: $\eta_{i1}$= $\alpha_{i1}{}^{1} \times \alpha_{i2}{}^{1} \times \alpha_{i3}{}^{0} \times \alpha_{i4}{}^{0} = \alpha_{i1} \times \alpha_{i2}$. For student *i*, $\eta_{i1}$ is equal to 1 when both attributes are present. The conditional distribution of **y** in relation to $\alpha$ is Equation 2:

$$P(\boldsymbol{y}|\, \alpha\,) = \prod_{j=1}^{J} P(\, y_i|\alpha) \tag{2}$$

Moreover, in practice, a student may guess an item correctly when he or she does not possess the required attributes. Alternatively, he or she may get an answer wrong when he or she possesses all the required attributes for that item. Therefore, two item parameters are needed to categorize item *j* in the DINA model. The parameter $s_j$ is the probability of slipping on the answer, defined as $s_j$= P ($y_{ij}$= 0| $\eta_{ij}$=1). The parameter $g_j$ is the probability of guessing the answer, which is defined as $g_j$= P ($y_{ij}$= 1| $\eta_{ij}$= 0). The probability of getting item *j* correctly for person *i* is Equation 3:

$$\mathrm{P}(y_{ij}= 1|\, \alpha_i) = (1 - \, s_j)^{\eta_{ij}} (g_j)^{1-\eta_{ij}} \tag{3}$$

The likelihood function of the response vector of student *i* is Equation 4:

$$L\,(y_i|\, \alpha_i) = \sum_{j=1}^{J} P(y_{ij} = 1|\alpha_i)^{y_{ij}}\, [1 - P\,(y_{ij} = 1|\alpha_i)]^{1-y_{ij}} \tag{4}$$

When using a logit function in the guessing parameter, the logit that a student answered an item correctly, but without the required attributes, becomes:

$$\text{logit P}(y_{ij}= 1|\, \eta_{ij}= 0)= f_j \quad (5)$$

Using the logit function, a correct answer given by a student thus becomes:

$$\text{logit P}(y_{ij}= 1|\, \eta_{ij}= 1)= f_j + d_j \quad (6)$$

The parameter $d_j$ is the capacity of classification for item *j*, which is the item discrimination index; an item with greater magnitude on this index indicates a higher power to detect the students with or without the required attributes. Equation 5 and Equation 6 can be written in one model (see Equation 7):

$$\text{logit P}(y_{ij}= 1|\, \eta_{ij})= f_j + d_j\eta_{ij} \quad (7)$$

Equation 7 is a logistic regression model with latent classes where item *j* is the *detector* of the required attributes. Equation 3 and Equation 7 are equivalent.

After describing the conditional distribution of $y_{ij}$, given $\alpha_i$, the probability distribution of $\alpha_i$ needs to be considered in the DINA model. The assumption is that all $\alpha_k$ are independent from one another with $2^K$ possible values for the saturated model. In the example provided in Table 15, under the DINA model, $K$ = 4 can produce 16 possible attribute vectors. The attributes are unstructured, so all eight vectors are permissible as profile score patterns. However, cognitive scientists have pointed out that some cognitive processes should not be investigated in isolation (Kuhn, 2001). Such an approach makes practical sense, because the model can be understood as

measuring one's general proficiency, along with measuring a set of specific attributes. In many assessment design situations, the SMEs define not only the latent attributes but also the latent relationship among them, for example, to explicitly put the constraints among the attributes in Q-matrix construction. A higher-order DINA (HO-DINA) model is constructed by adding a high-order continuous proficiency term in estimating students' profile scores. The advantage of HO-DINA is that it reduces the number of attributes in combination (de la Torre & Douglas, 2004). Figure 5 below shows an example higher-order structure when $K$ = 3 latent attributes, along with the latent structure among them, are defined in the assessment.

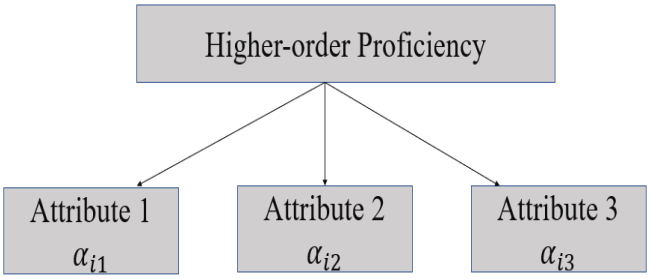



Figure 5 A graphical representation for higher-order DINA model.

The higher-order latent proficiency term being added is denoted as $\theta_i$. The elements of $\alpha_i$ can be conditionally assumed as independently distributed given the high-order term $\theta_i$, where $P(\boldsymbol{\alpha}|\,\boldsymbol{\theta}) = P(\alpha_1, \alpha_2, \ldots, \alpha_k|\,\boldsymbol{\theta}) = \prod_k P(\boldsymbol{\alpha}_k|\,\boldsymbol{\theta})$. In the DINA model, the independence assumption related to $\alpha$ is $P(\alpha) = P(\alpha_1, \alpha_2, \ldots, \alpha_k) = \prod_k P(\boldsymbol{\alpha}_k)$. The higher-order proficiency term determines the probability that the student possess each of the attributes. The parameterization of HO-DINA can be thought of as similar to the relationship between items and latent proficiency in the traditional IRT model shown in Equation 1. The formal HO-DINA model is shown in Equation

8 below. Students who have greater higher-order proficiency tend to possess lower-level attributes, where the attributes with larger magnitudes on $\lambda_{0k}$ are regarded as more difficult to master. According to de la Torre, Hong, and Deng (2010), a one-parameter logistic (1PL) model can be employed, which is termed a *restricted high-order DINA model*, because $\lambda_1$ is assumed equal across all attributes.

$$P(\alpha_k|\,\theta) = \frac{exp[1.7\,\lambda_1(\theta - \lambda_{0k})]}{1+exp[1.7\,\lambda_1(\theta - \lambda_{0k})]} \tag{8}$$

Junker and Sijtsma (2001) proposed the *noisy-input, deterministic-and-gate* (NIDA) model, shown in Equation 9. It shares many similarities with the DINA model, except the slipping parameter and guessing parameters now are indexed with attribute *k* rather than with item *j*. (Recall that the $\eta_{ij}$ is a deterministic term in DINA models). The same concept is used in the NIDA model to define guessing parameter and slipping parameter. The interpretation has changed to the attribute level. That is, the slipping parameter: $s_k$= P ($\eta_{ijk}$=0| $\alpha_{ik}$=1), represents the probability of incorrectly applying attribute *k* for item *j* while the attribute *k* is actually mastered. The guessing parameter is correctly applying attribute *k* for item *i* while the attribute is not mastered, which is expressed as $g_k$= P ($\eta_{ijk}$ = 1| $\alpha_{ik}$=0).

$$P(y_{ij}= 1|\mathbf{s, g, \alpha})= \prod_{k=1}^{K} P(\eta_{ijk} = 1|\boldsymbol{\alpha}_k) = \prod_{k=1}^{K}[\,(1 - s_k)^{a_{jk}} {g_k}^{1-a_{jk}}]^{q_{jk}} \tag{9}$$

Templin and Henson (2006) proposed the DINO (deterministic input; noisy "or" gate) model. It is a disjunctive model, meaning that students can solve the item if they possess only one attribute required for the item; having more than one required attribute does not increase the probability of getting an correct answer. Compared with the DINA model, the deterministic term

is now denoted as $w_{ij}$, where $w_{ij}$ = 1- $\prod_{k=1}^{K}(1-\alpha_{ik})^{q_{ik}}$. Unlike $\eta_{ij}$ in the DINA model, $w_{ij}$ can only become zero when none of the attributes is present. The slipping parameter and guessing parameter are defined as following: $s_j$= $P(y_{ij}$= 0| $w_{ij}$=1); $g_j$= P($y_{ij}$= 1| $w_{ij}$=0). The DINO model, which can be compared with Equation 3 (i.e., DINA model). It is defined in Equation 10 as:

$$P(y_{ij}= 1|w_{ij}) = (1 - s_j)^{w_{ij}}(g_j)^{1-w_{ij}} \quad (10)$$

Henson, Templin, and Willse (2009) proposed a cognitive diagnosis model using a log-linear model (i.e., LCDM). For item 1 in Table 16, two attributes are required: addition and subtraction, $\alpha_k$= $\{\alpha_1, \alpha_2\}$. Equation 11 shows the logit of answering item 1 correctly where three generic terms in log-linear model are defined in LCDM. If a student *i* possesses both $\alpha_1$ and $\alpha_2$, the logit for answering item 1 correctly can be defined with one main effect associated with $\alpha_1$, one main effect associated with $\alpha_2$, and the interaction between $\alpha_1$ and $\alpha_2$. Note that if an item requires more attributes, for example, item 3 requires three attributes, then, the higher-order interaction will be needed. So more parameters will be estimated.

$$\text{logit}(y_{ij}= 1|\alpha_i) = \lambda_{i,0} + \lambda_{i,1,(1)}\alpha_{i1} + \lambda_{i,1,(2)}\alpha_{i2} + \lambda_{i,2,(1,2)}\alpha_{i1}\alpha_{i2} \quad (11)$$

The main effect for attribute $\alpha_1$ is $\lambda_{i,1,(1)}$. It represents *the increase* in the logit for mastering attribute $\alpha_1$ in student *i* who has not mastered attribute $\alpha_2$.The main effect for attribute $\alpha_2$ is $\lambda_{i,1,(2)}$. It represents *the increase* in the logit for mastering $\alpha_2$ in student *i* who has not mastered attribute $\alpha_2$. The additional change in the logit for mastering both attributes is $\lambda_{i,2,(1,2)}$. The intercept, $\lambda_{i,0}$, represents the logit for the reference group, which is usually the students who have both attributes missing (that is, $\alpha_{i1}$= $\alpha_{i2}$= 0).

According to the preceding discussion, $2^k$ is the total number of attribute vectors. If the example in Table 16 is used, there are 16 possible attribute vectors. Instead of having $2^k$attribute vectors, the notion of reduced number of attribute vectors, $2^{K_j^*}$ , is introduced, where $K_j^* = \sum_{k=1}^{K} q_{ik}$ is the number of required attributes for item *j* (de la Torre, 2011). The reduced attribute vector for item *j* can be denoted as $\alpha_{ij}^*$, where the elements $\alpha_{ij1}$, $\alpha_{ij2}$, $\ldots.\alpha_{ijK_j^*}$, show the required attributes for item *j*. The purpose of using the term $\alpha_{ij}^*$ is to express the reduced number of latent attribute vectors from $2^k$ to $2^{K_j^*}$in the model. The notion of strict inequality describes the model, that is:

$\alpha_{ij}^* < \alpha_{i'j}^*$, if and only if the elements in the vectors satisfy $\alpha_{ik}^* < \alpha_{i'k}^*$. This implies the sum of the two vectors has the following property: $\sum_{k=1}^{K_j^*} \alpha_{ijk}^* < \sum_{k=1}^{K_j^*} \alpha_{i'jk}^*$. This property does not imply that $\alpha_{ik}^* < \alpha_{i'k}^*$. The probability that a student with attribute pattern $\alpha_{ij}^*$ correctly answers item *j* is $P(X_j = 1 | \alpha_{ij}^*) = P(\alpha_{ij}^*)$.

Unlike the DINA model, the generalized DINA (G-DINA) model contains $2^{K_j^*}$ parameters that need to be estimated for item *j*, with the constraint that $P(\alpha_{ij}^*) \leq P(\alpha_{i'j}^*)$. Equation 12 describes the G-DINA model, where $P(\alpha_{ij}^*)$ is the sum of effects of attributes and the interactions among the attributes. When using the logit link function, the G-DINA model becomes a log-linear model described in Equation 13.

$$P(\boldsymbol{\alpha}_{ij}^*) = \delta_{jo} + \sum_{k=1}^{K_j^*} \delta_{jk}\alpha_{ik} + \sum_{k'=k+1}^{K_j^*} \sum_{k=1}^{K_j^*-1} \delta_{jkk'}\alpha_{ik}\alpha_{ik'} \ldots + \delta_{j12\ldots K_j^*} \prod_{k=1}^{K_j^*} \alpha_{ik} \quad (12)$$

$$\text{logit}[P(\boldsymbol{\alpha}_{ij}^{*})] = \lambda_{jo} + \sum_{k=1}^{K_j^*} \lambda_{jk}\,\alpha_{ik} + \sum_{k'=k+1}^{K_j^*} \sum_{k=1}^{K_j^*-1} \lambda_{jkk'}\,\alpha_{ik}\alpha_{ik'} \ldots + \lambda_{j12\ldots K_j^*} \prod_{k=1}^{K_j^*} \alpha_{ik} \quad (13)$$

The intercept for item $j$ is $\delta_{jo}$. The main effect coming from source $\alpha_k$ is $\delta_{jk}$. The interaction between two attributes, $\alpha_k$ and $\alpha_{k'}$, is denoted as $\delta_{jkk'}$. The higher order interaction among more than two attributes is denoted as $\delta_{j12\ldots K_j^*}$.

The interpretation of the terms is the following: $\delta_{jo}$ is the probability of answering item *j* correctly when all the required attributes for item *j* are missing. The term, $\delta_k$, is the change in the probability of answering item *j* correctly when one attribute (i.e., trait *k*) is present. The term, $\delta_{jkk'}$, is the change in the probability of answering item *j* correctly due to both attribute *k* and attribute $k'$ being mastered. The G-DINA model will become the DINA model when the following constraints are applied: estimating $\delta_{jo}$ and $\delta_{j12\ldots K_j^*}$ but setting all other parameters at zero. This estimates a smaller number of parameters per item, that is, two parameters for each item. The guessing parameter $g_j$ in DINA is now $\delta_{jo}$; 1-$s_j$ is now equal to $\delta_{jo}$+ $\delta_{j12\ldots K_j^*}$.

**Summary Statistics**

In the CTCA, the response data collected from operational testing came from 564 students, 283 of whom were male and 281 of whom were female. Figure 5 (below) shows how the total scores are distributed across all students using a box plot.

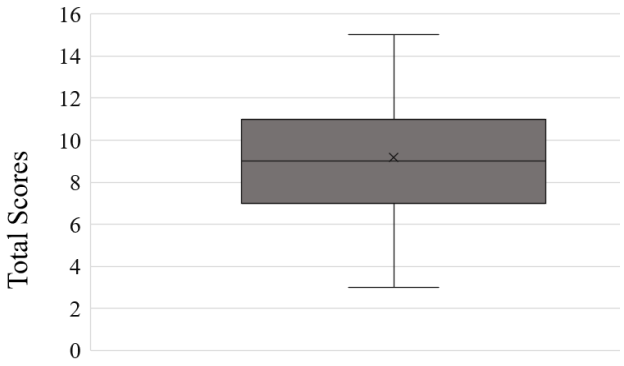



Figure 6 The distribution of total scores

A *t*-test is conducted for this sample. The results show that in terms of total scores, the difference between the two groups (male vs. female) is a marginally statistically significant: female (M=8.9, SD=2.89), male (M=9.4, SD=2.88), t(562) = 2.00, $p$ = 0.05. In addition, for the sample in the operational testing, no DIF was found in any of the items.

**Parametric Psychometric Modeling**

The DINA model is chosen as the scoring model for the response data. In the CTCA, as described in the previous design activity, SMEs reviewed the Q-matrix and the content of items and hypothesized how students will correctly answer an item: (1) a student must master all the attributes required in that item in order to answer it correctly, and (2) attributes are equally important. To this end, the DINA model is chosen, given that Lee, Park, and Taylan (2011) have emphasized "when attributes required for an item are considered equally important, the DINA model is deemed appropriate (p. 149). Moreover, the "DINA model is a parsimonious and interpretable model that requires only two parameters for each item regardless of the number of attributes being considered" (de la Torre, 2009, p. 117).

In the CTCA, the estimation was conducted in the CDM package (Robitzsch, Kiefer, George, & Uenlue, 2018) in R. For the model-data fit criteria, the Standardized Root Mean Squared Residual (SRMSR) is 0.037. The SRMSR is an index for absolute fit statistic. The criterion is that if the magnitude is less than 0.09, it indicates a reasonable model-data fit (Maydeu-Olivares & Joe, 2014). Thus, for this sample, the model-data fit is reasonably good.

Total score reliability measure (Cronbach's alpha or omega index) commonly seen in the measurement literature is not applicable in CDM-based instruments. Such a measure is calculated based on the true-score theory, assuming some proportion of observed-score variance is traceable to a continuous underlying true-score. In contrast, CDM conceptualizes an examinee's "true-score" as a categorical mastery profile.

The CDM package provides two estimated item parameters of the DINA model: the guessing ($g$) and the slipping ($s$) probabilities for each item (see Equation 5 and Equation 6 above). In terms of interpretation, these two parameters are often used to reflect the *item discrimination index* (IDI, denoted as $d$), that is, the probability of correctly solving an item without the influence of guessing and slipping (i.e., $d = 1 - g - s$). There is no cut-point to judge the parameters' magnitudes in the measurement literature. Table 17 below shows these three magnitudes across 15 items on the CTCA. Item 1 has high magnitude on the guessing parameter. Item 10 has high magnitude on the slipping parameter. Thus, these two items possess very low IDI values, which may be used as indicator of poor quality of the assessment (e.g., items or the Q-matrix).

Table 17 Item Discrimination Index

| Item ID | Guessing Parameter | Slipping Parameter | IDI |
|---|---|---|---|
| 1 | 0.88 | 0.03 | 0.09 |
| 2 | 0.73 | 0.01 | 0.26 |
| 3 | 0.48 | 0.06 | 0.46 |
| 4 | 0.35 | 0.05 | 0.59 |
| 5 | 0.20 | 0.47 | 0.33 |
| 6 | 0.69 | 0.05 | 0.26 |
| 7 | 0.53 | 0.16 | 0.31 |
| 8 | 0.27 | 0.26 | 0.48 |
| 9 | 0.38 | 0.21 | 0.41 |
| 10 | 0.37 | 0.57 | 0.06 |
| 11 | 0.66 | 0.08 | 0.26 |
| 12 | 0.41 | 0.24 | 0.35 |
| 13 | 0.35 | 0.26 | 0.39 |
| 14 | 0.31 | 0.25 | 0.44 |
| 15 | 0.13 | 0.55 | 0.32 |

In the existing literature, one CDM-based instrument (Kuo, Chen, Yang, & Mok, 2016) that was developed using a DINA model. On that instrument, the item-level magnitudes on slipping parameter, guessing parameter or IDI were not reported. In the literature regarding retrofitting CDM, most studies also failed to provide such information (e.g., Garcia, Olea, & de la Torre, 2014; Chen & Chen, 2016). One article did provide these three indices which was written by Lee, Park, and Taylan (2011). They used DINA model to analyze the samples from Massachusetts, Minnesota, and the U.S. national sample for the 2007 the Trends in International Mathematics and Science Study (TIMSS) mathematics test. They reported these 3 indices for the three samples across 25 items. For the guessing parameter and slipping parameter, they ranged from 0.00 to 0.95. The IDI magnitude ranged from 0.00 to 0.95. The authors used the guessing parameter and slipping parameter to compare the differences among samples on a given item.

Other than the model-data fit and item-level statistics reported above, scoring results for the CTCA provide additional psychometric properties. Because subscore reporting is part of an

instrument, psychometric properties about the distinctiveness of subscores are needed (Wainer, Sheehan, & Wang, 2000). The tetrachoric correlations among latent attributes were positive and moderate: the coefficient between A1 and A2 is 0.71; the coefficient between A2 and A3 is 0.46; the coefficient between A3 and A1 is 0.78. These positive correlations indicate that mastering one attribute heightens the possibility of mastering others. In the CDM literature, no criterion has been specified to judge the magnitude of the tetrachoric correlations for evaluating attribute separation. Jurich and Bradshaw (2014) reported the tetrachoric correlations in their newly developed CDM-based instrument in higher education. The coefficients among four latent attributes ranged from 0.56 to 0.94. They further pointed out that if the correlation coefficient is (nearly) perfect (e.g., 0.94), it suggests that two attributes are inseparable. Bradshaw, Lzsak, Templin, and Jacobson (2014) reported tetrachoric correlations among the attributes in their newly developed CDM-based instrument. The range of the coefficients was from .626 to .781. Thus, the attributes in the CTCA seem to be reasonably separated because none of the two attributes was highly correlated.

Figure 6 shows the distribution of pattern scores, which represents the mastery status across this sample of students. Recall that, in the DINA model, the number of latent patterns depends on how many latent attributes the test aims to elicit. In the CTCA, three latent attributes were extracted and defined as elements of students' CT competency. Thus, a student can be classified in any one of eight latent groups to indicate his or her mastery status: {000}, {100}, {010}, {001}, {110}, {101}, {011}, or {111}. Mastery status on a particular attribute is indicated by a 1, and non-mastered status by a 0. For example, the pattern score {100} represents the group of students who mastered A1, but failed to master the rest of the attributes.

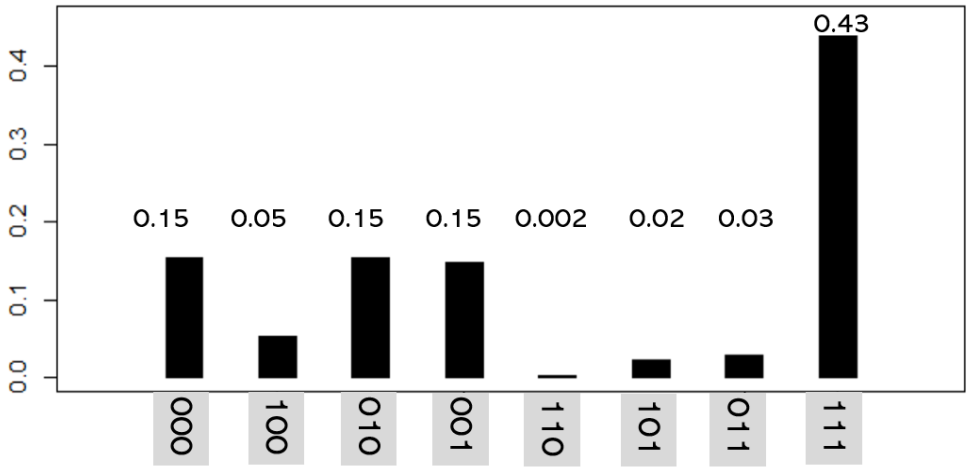



Figure 7 Classification on latent pattern scores

Because CDM uses a probability model to estimate mastery status, the estimated latent scores are expressed in a 0-1 scale, where magnitudes below 0.5 indicate an attribute has probably not been mastered. The results can also be expressed in a percentage fashion. For example, according to Figure 6, about 43% of students have mastered all three attributes of CT competency. For the remaining students, about 15% of students will need the most instructional support in their CT learning because they mastered none of the attributes; about 35% of students only mastered one attribute. Figure 7 shows the attribute-level mastery status across this sample. Among all students, about 50% of students mastered A1, 62% of them mastered A2, and 64% of them mastered A3.

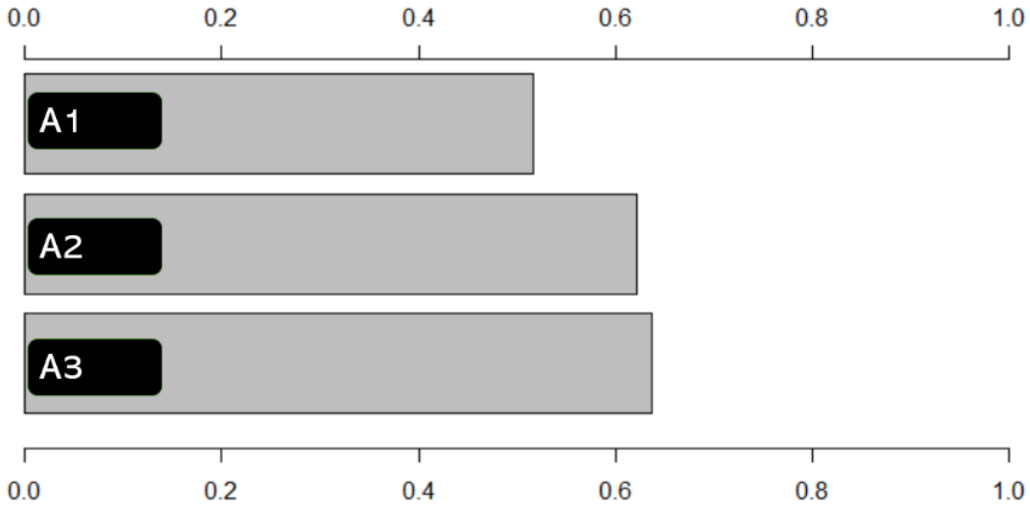
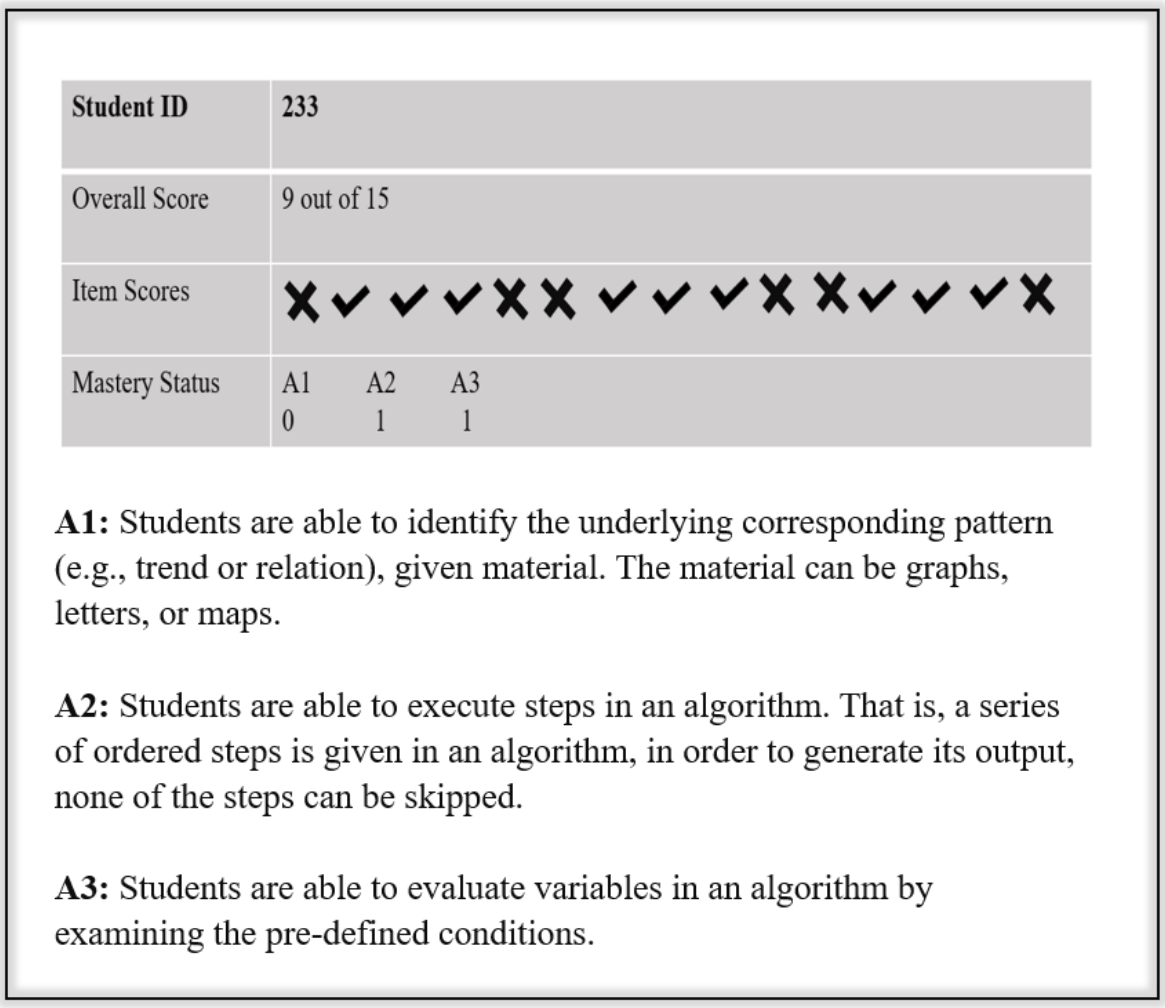



Figure 8 Mastery status on each attribute

For individual score reporting, the subscore, the latent mastery probability estimated by the DINA model, is reported to represent each student's mastery status along with the total score (i.e., the observed sum scores). Figure 8 below is an example of a score report. This student is from the sample in operational test. The first row of the score report is the student ID, which the teacher is able to link it to the student's name. The second row is the total score the student received. The third score includes further detail about the wrong and right answers, where 1 indicates the correct answer, 0 otherwise. The last row displays the indicators of mastery status provided by the DINA model estimation. The results shown in the example suggest that the student should practice more on A1 in future learning.



| Student ID | 233 |
|---|---|
| Overall Score | 9 out of 15 |
| Item Scores | ✗ ✓ ✓ ✓ ✗ ✗ ✓ ✓ ✓ ✗ ✗ ✓ ✓ ✓ ✗ |
| Mastery Status | A1 A2 A3<br>0 1 1 |

**A1:** Students are able to identify the underlying corresponding pattern (e.g., trend or relation), given material. The material can be graphs, letters, or maps.

**A2:** Students are able to execute steps in an algorithm. That is, a series of ordered steps is given in an algorithm, in order to generate its output, none of the steps can be skipped.

**A3:** Students are able to evaluate variables in an algorithm by examining the pre-defined conditions.

Figure 9 Score report card for an individual student

In this research, score reports were not provided to the classroom teachers. However, the score report should be particularly useful, especially if a large-scale study is conducted in the future. For example, when a school district (which involves multiple classrooms or schools) adopts a CT curriculum, the CTCA can be administered to pre-assess students' CT competency, and score reports can be distributed to teachers so that teachers may gain the understanding about the CT of their students.

# CHAPTER 5. DISCUSSION

## 5.1 Discussion on the Development of the CTCA

The emerging field of computational thinking (CT) has received international attention. A *Computing at School* (CAS) curriculum has been introduced to classrooms in the United Kingdom, with a focus on CT (Curzon, Dorling, Ng, Selby, & Woollard, 2014). In 2012, the Israeli Ministry of Education initiated a program to improve science and technology curricula through the addition of a CT component (Gal-Ezer & Stephenson, 2014). In the United States, National Science Foundation's *STEM+C* (science, technology, engineering, mathematics and computing) program aims to incorporate CT research in K-12 education. The hope is that the computational preparation comes into play in cultivating the future generation for the nation's prosperity (National Science Foundation, 2018).

Other than cultivating citizens in general, improved CT research and practices can also benefit the computing profession in particular. In the computing field, consistent low participation rates among particular subgroups are problematic. According to the College Board (2016), only 21.9% of test-takers on the 2015 Advanced Placement (AP) Computer Science exam were female students, the lowest female participation among all AP exams. The same test recorded the following participation rates for racial or ethnic minority subgroups: 3.9% of Black or African American students, 9% of Hispanic or Latino students, and 0.4% of American Indian students. The possible explanation for this is that K-12 classrooms' activities or tasks fail to promote computational participation. As Margolis, Ryoo, Sandoval, Lee, Goode, and Chapman (2012) commented, "Especially in schools with high numbers of African American and Latino/a students, computer classes too commonly offer only basic, rudimentary user skills rather than engaging

students with the problem-solving and computational thinking practices that are the foundation of computer science" (p. 73).

Indeed, effective CT instruction (e.g., meaningful classroom activities) is the foundation to increase students' interests in learning computer science or choosing computer science as a major. As Lu and Fletcher (2009) have argued, in CT instruction, "efforts must be made to lay the foundations of computational thinking long before students experience their first programming language" (p. 3). Along this line, many scholars in the CT community have underlined how CT instructions impact on students' CT learning. Hsu, Chang, and Hung (2018) in their review paper provided a few of directions for future CT research. One of the directions is related to knowing about students' learning status. "Teachers need to consider students' learning status when teaching so as so guide the students through the CT training courses, or offer proper aid or feedback regarding different students" (p. 308).

This research on the Computational Thinking Competency Assessment (CTCA) has yielded results that echo back to this direction. The results produced from this research can inform teachers about students' CT competency. Specifically, the model-based classification has reflected students' mastery status on pre-defined attributes of CT (see Figure 8), with 0 or 1 was used as indicators to convey the mastery status. The pre-defined three attributes were:

**A1:** Students are able to identify the underlying corresponding pattern (e.g., trend or relation), given material. The material can be graphs, letters, or maps.

**A2:** Students are able to execute steps in an algorithm. That is, a series of ordered steps is given in an algorithm, in order to generate its output, none of the steps can be skipped.

**A3:** Students are able to evaluate variables in an algorithm by examining the pre-defined conditions.

For the sake of brevity, these variables are: Attribute 1 refers to *pattern recognition* (A1), Attribute 2 refers to *sequence of instructions* (A2), and Attribute 3 refers to *evaluation of variables* (A3).

To answer Research Question 1, for middle school students, my findings suggest that mastery status in CT competency has the following characteristics:

(1) About 15% of students showed struggles in CT competency because they mastered none of attributes - pattern recognition (A1), sequence of instructions (A2), or evaluation of variables (A3). This group of students may need the most instructional support.

(2) Across all students, the most effort should be made on pattern recognition (A1). These efforts may include students studying this topic more or direct instruction on A1 being planned. A few examples of relevant instructional activities are suggested below (to see a comprehensive list of examples about lesson plans and classroom activities, see the book edited by Kong and Abelson, 2019).

(3) The mastery status on sequence of instructions (A2) or evaluation of variables (A3) is better than on A1. For A2, about 40% of students did not master it. A similar finding was evident for A3.

(4) The three points mentioned above are described in detail at individual student level. Each student has his or her profile scores reporting about their mastery status.

Turner (2014) presented details about how to create an assessment-centered classroom for middle school teachers. The goal is to both measure and *promote* learning. Turner pointed out that

effective instruction started by surveying what students bring in to the classroom. Teachers answer the questions before instruction begins. For example, (1) "What are students' current levels of knowledge about this topic/unit? (2) Have some students mastered all or some of the topic/unit goals?" (p. 4). The answers for these questions can create more effective instruction. This research has the capacity to inform instruction by equipping teachers with one tool to pre-assess existing CT competency. The proposed score use for the CTCA is that teachers may gain information about students' current CT competency before instruction begins.

Moreover, this tool differs from other CT assessments reviewed and synthesized earlier in this research. Other assessments possess the capacity to measure CT learning. The CTCA is able to inform instruction because it can both measure and *promote* learning in CT. For example, the findings used to address Research Question 1 are the mastery status for each student, as well as classification rates on the three attributes. The findings may help teachers facilitate their students *mastery orientations*, that is, to address the gap between what students know and what students need to know in order to meet desired learning goals.

When teachers address the gap, they communicate descriptive feedback to clarify learning for students. Feedback is goal-referenced information and actionable information (Hattie & Timperley, 2007). It is not an advice or praise from a teacher. Feedback is the foundation of effective instruction (Fund, 2010). Imagine that a piece of information provided by a teacher to his/her student: "your grade is B+", or "out of 12 questions, you got 4 wrong." This information is not feedback because this information fails to provide what *specifically* the student do next time with what goal in mind. But the results from the CTCA can be a basis for establishing a learning profile for each student where learning goal-referenced and actionable information are conveyed to the student.

Not only can each individual student benefit from the CTCA by receiving learning goal-referenced and actionable information, but also teachers can benefit from the CTCA by using its results to adjust their ongoing instruction. For example, instead of giving same problem-solving sheets (activities) to all the students collectively in the class, a teacher can assign different problems (activities) for different students to practice. Teachers can do this with a deliberate focus. For example, for students who fail to master pattern recognition (A1), teachers can use English language grammar rules as the context for students to practice (see examples provided by Rich & Hodges, 2017). For students who fail to master sequence of instructions (A2), Yadav, Stephenson, and Hong (2017) have suggested that teachers can use computer-free classroom activities to highlight the steps involved, for example, during a lab experiment. Understanding a set of precise steps is a basis of understanding sequence of instructions. As another example, teachers can require students to write down the instruction about a cookie-making sequence (Barr & Stephenson, 2011).

For students who fail to master evaluation of variables (A3), teachers can develop some activities for students to practice. For example, the activities can resemble Item 5 on the CTCA; students are required to evaluate how variable Y and variable Z change over time (A3). Teachers can also give students computer-based exercises (e.g., Scratch) to recognize that variables do not hold a value (data); variables point to it. Previous research (Hambrusch, Hoffmann, Korb, Haugan, & Hosking, 2009) suggested students had difficulty evaluating variables in CT because they treated variables in CT like variables in mathematics. In mathematics, variables (e.g., in an equation) are bound to a given value only once and then keep that value. In contrast, when evaluating a variable in CT, the variable should be treated like a storage box with a name. Its value can be changed over time.

To answer Research Question 2, the pilot and operational test results may suggest somewhat different conclusions. At the pilot test, the female and male groups had no statistical difference in total scores on the CTCA. At the operational test, the two groups had a marginally statistically-significant mean difference, where male group was slightly better. Existing literature generally suggests CT competency (or ability) is very similar across gender groups, although differences have been found in non-cognitive aspects (e.g., confidence). Weintrop et al. (2016) reported on integration of CT into science and math classrooms in high schools. The non-cognitive questions included students' attitudes towards CT and their confidence in solving CT problems in math and science contexts. Compared with the male group, the female group felt that CT was less important, and showed less confidence in solving CT problems. However, on the score on the CT competency questions, the between-group difference was not statistically significant. Similarly, Hutchins, Zhang, and Biswas (2017) studied high school students' CT development. They found that "while girls have a significantly lower confidence in CT applications, there is no difference in their ability to perform CT tasks when compared to their male counterparts" (p. 38). Mouza, Marzocchi, Pan, and Pollock (2016) designed an after-school computer science program for middle school students, where CT consisted of *computer science concepts*, *computational practices*, and *attitudes toward computing*. No statistical difference was found between gender groups for any of these areas.

## 5.2 Limitations and Future Directions

Some weaknesses in the current CTCA specification should be taken into consideration before future revisions or use of the instrument. As Mislevy (2007) has articulated, validation is an open-ended process. Validity evidence is constantly being collected, even for a test that has existed for years (e.g., Liu, Bridgeman, Gu, Xu, & Kong, 2015; Kleiger, Bridgeman, Tannenbaum,

& Olivera-Aguilar, 2018). Similarly, future empirical data should be collected for the CTCA to further validate the score meaning and possibly revise the items (and Q-matrix).

Recall the rising concern from the item parameters in Table 17. Two items possessed very low item discrimination index magnitudes because of high slipping and guessing parameter values. A consequence of high guessing and slipping parameters on the CDM-based instrument is that the classification accuracy is affected. An ideal model-based classification is that the mastery status provided by the scoring model can truly represent a student's mastery of certain attributes.

Multiple reasons can explain the presence of a high guessing parameter. First, students may answer the item correctly solely because of randomness (e.g., luck). Second, item design flaws may give away the answers (e.g., the stimuli on the item stem). Thirdly, students may have solved the item correctly because they used a different set of strategies (e.g., mental operations) that were not captured by the Q-matrix (or SMEs' hypothesis), as described by de la Torre (2009). Two reasons can explain the presence of a high slipping parameter. First, the item design has flaws that caused confusion for students. Students who possess the required attributes then may not answer the item correctly. Secondly, if students were not careful, the slipping may occur because of random measurement error.

By examining the CTCA items, no problematic patterns (e.g., stimuli) in the content of these two items were found. A possible explanation is that the item parameters are related to the students' characteristics. If this is the case for the CTCA, future revisions should incorporate the sample's characteristics. Using DINA model as the scoring model, each sample may have a different set of estimated item parameters (Lee, Park, & Taylan, 2011). An alternative scoring model, for example, could be the random-effect DINA model (developed by Huang and Wang, 2014). This approach argues that slipping and guessing parameters should not be fixed values,

but rather depend on persons' (students') characteristics. "The level of slipping may depend on the person's degree of caution, and the probability of a correct guess may be determined by the person's ability to eliminate distractors among the options included in an item" (p. 75). This model approach emphasizes the estimation of variation in slipping and guessing across the entire sample of students. Thus, these two parameters would no longer be considered solely as item characteristics.

In addition, according to the Q-matrix (Figure 4), no item solely measured A2. In future, if the testing time is still in a reasonable range, one or two items can be added in the CTCA to measure A2 specifically. Moreover, two open-ended items could also be added because the students' responses to the open-ended items can be considered as educational artifacts to be further analyzed, which in turn may help researchers and practitioners understand students' thinking process. In contrast, MC items provide little concrete evidence of students' cognitive process.

The CDM approach embedded in the CTCA is a tool to collect empirical evidence that may help the CT community better understand the domain of CT competency. Future classroom activities should be structured in a way to provide students with opportunities to think computationally. Future studies in CT instruction and CT learning should collect more empirical evidence about relevant cognitive aspects while students solve CT problems. For example, future studies should examine how CT competency is changing on a learning spectrum, perhaps particularly examining the trajectories of CT acquisition from sixth grade to eighth grade. After all, assessment and instruction should inform each other through empirical evidence, especially in light of the fact that the CT field is relatively new compared with traditional STEM domains.

# CHAPTER 6. CONCLUSION

The research reported here is a computational thinking competency assessment (CTCA) specification. A set of multiple-choice items were developed that can be used to assess students' computational thinking competency by classroom teachers in middle schools. The main highlight for CTCA is *subscore reporting*, made available to practitioners because the scoring system harnessed the assessment design to provide information about students' CT competency. The integration of CT into different contexts such as science or programming is rather common. Subsequently, different sets of measurement tasks are needed to most directly assess students' substantive area and CT competencies. The CTCA is designed for wide usage. The reasonable testing time also strengthens its practical usability across middle school classrooms.

This section underlines a few of the design features of CTCA. Firstly, throughout the assessment design, multiple types of validity evidence were collected, including using (1) national standards documents, (2) input from subject matter experts (SME), and (3) the existing literature. The design's initial stages included an examination of national standards to ensure the feasibility of establishing a CTCA specification. Then, by examining the existing literature in CT, the standard statements were adapted into more fine-grained and latent attributes. Finally, by revising and refining the items and the testable elements, the CTCA specification produced subscores through CDM to reflect students' CT competency. The alignment table (Q-matrix) was developed to indicate the explicit relationship between items and attributes. Multiple rounds of panel review were conducted to ensure the quality of the item prototypes and the soundness of Q-matrix. Also, by examining the summary statistics, the total scores were reasonably spread out.

Secondly, empirical evidence was used to provide the *validity evidence of response processes*. This piece of validity evidence enables assessment developers to capture the extent to

which they can adequately model students' response processes. In this context, the degree of certainty for this question can help CTCA preserve the proposed scoring interpretation. The evidence was extracted from the verbal data and artifacts produced by students during the cognitive interviews. Multiple item prototypes were deleted or revised because empirical evidence indicated they failed to elicit the hypothesized response process (i.e., the mental operations). Thirdly, other validity evidence also emerged through psychometric properties of the item response data, such as the correlation between the total scores and the academic grades for students. Differential item functioning (DIF) was used to examine the quality of items in terms of gender bias. The tetrachoric correlations among latent attributes indicated that the internal structure of the observed response data appeared consistent with the hypothesis that the assessment was measuring multiple discrete latent CT competency attributes.

The subscore reporting design feature is highly likely to benefit the end-users of the assessment – classroom teachers. The CTCA design methodology took advantage of cognitive diagnostic modeling (CDM) approach to explore CT competency, which in turn can inform instructional practices in CT. The traditional measurement is usually a single reported score; for example, a total score or a single latent score estimated from item response theory (IRT) is useful for ranking students on a predefined continuum. In contrast, CDM provided model-based classifications as well as attribute-level mastery status (i.e., weakness or strength) which in turn involves placing instructional targets on students' areas of weakness.

The CTCA specification seeks to characterize CT competency in terms of improving students' learning. Subscore reporting does not encourage a teacher merely to tell a student that he/she needs to answer two more items correctly in order to attain mastery. Subscore reporting is a tool (assessment) for teachers (instructional practice) to help students embody the meaning of

good work (learning) with the extracted learning goals (attributes or claims). By connecting instructional practices and assessment to inform each other to enrich students' learning experience, the CTCA has enabled a modest but important innovation in the CT filed.

# REFERENCES

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

Attali, Y., Laitusis, C., & Stone, E. (2016). *Differences in reaction to immediate feedback and opportunity to revise answers for multiple-choice and open-ended questions Educational and Psychological Measurement, 76*(5), 787-802.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48- 54.

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology, 38*(6), 20-23.

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J., & Clark, S. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning, 11*(1), 1-35.

Baker, R. S., Martin, T., & Rossi, L. M. (2016). Educational data mining and learning analytics. In A. A. Rupp & J. P. Leighton (Eds.), *Wiley handbook of cognition and assessment: frameworks, methodologies, and applications* (pp. 379-396). Oxford, UK: John Wiley & Sons.

Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology, 25*(5), 628-647.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*(1), 145-157.

Blanton, M., Stephens, A., Knuth, E., Gardiner, A. M., Isler, I., & Kim, J. (2015). The development of children's algebraic thinking: The impact of a comprehensive early algebra intervention in third grade. *Journal for Research in Mathematics Education, 46* (1), 39-88

Bradshaw, L., Izsák, A., Templin, J., & Jacobson, E. (2014). Diagnosing teachers' understandings of rational number: Building a multidimensional test within the diagnostic classification model framework. *Educational Measurement: Issues and Practice, 33(*1), 2-14.

Brenner, M., Mayer, R., Moseley, B., Brar, T., Durán, R., Reed, B., & Webb, D. (1997). Learning by understanding: The role of multiple representations in learning algebra. *American Educational Research Journal, 34*(4), 663-689.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada. Retrieved from: http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf.

Bridgeman, B. (1993). *A comparison of open-ended and multiple-choice question formats for the quantitative section of the graduate record examinations general test* (GRE Board Professional Report No. 88-13P). Princeton, NJ: Educational Testing Service.

Briggs, D. C. (2017). Learning theory and psychometrics: room for growth. *Assessment in Education: Principles, Policy & Practice, 24*(3), 351-358.

Briggs, D. C., & Alonzo, A. C. (2012). The psychometric modeling of ordered multiple-choice item responses for diagnostic assessment with a learning progression. In A. Alonzo & A. W. Gotwals (Eds.), *Learning progressions in science: Current challenges and future directions* (pp. 293–316). Boston, MA: Sense Publishers.

Baker, R. S., Martin, T., & Rossi, L. M. (2016). Educational data mining and learning analytics. In A. A. Rupp & J. P. Leighton (Eds.), *Wiley handbook of cognition and assessment: frameworks, methodologies, and applications* (pp. 379-396). Oxford, UK: John Wiley & Sons.

Carlson, M., Oehrtman, M., Engelke, N. (2010). The precalculus concept assessment (PCA) instrument: A tool for assessing reasoning patterns, understandings and knowledge of precalculus level students. *Cognition and Instruction, 28*(2), 113-145.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education, 109*(1), 162-175.

Chen, H., & Chen, J. (2016). Retrofitting non-cognitive-diagnostic reading assessment under the generalized DINA model framework. *Language Assessment Quarterly, 13*(3), 218-230.

Cooper, S., Grover, S., Guzdial, M., & Simon, B. (2014). A future for computing education research. *Communications of the ACM, 57*(11), 34-36.

College Board (2017). *AP course and exam description. AP Computer Science principles including the curriculum framework computer science principles including the curriculum framework.* Retrieved from https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf

Computer Science Teachers Association. (2011). *K-12 computer science standards.* New York, NY: CSTA.

CSTA & ISTE (2011). Operational definition of computational thinking for K-12 education. Retrieved from: http://csta.acm.org

Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). Developing computational thinking in the classroom: A framework. Retrieved from http://eprints.soton.ac.uk/id/eprint/369594

Czerkawski, B. C. & Lyman, E.W. (2015). Editorial: Educational computing and computer science. *Issues and Trends in Educational Technology, 3*(2), 1-2.

de la Torre, J. (2009). DINA model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics, 34*(1), 115-130.

de la Torre, J. (2011). The generalized DINA model framework. *Psychometrika, 76*(2), 179-199.

de la Torre, J., Hong, Y., & Deng, W. (2010). Factors affecting the item parameter estimation and classification accuracy of the DINA model. *Journal of Educational Measurement, 47*(2), 227-249.

Denning, P. J. (2009). The Profession of IT: Beyond computational thinking. *Communications of The ACM, 52*(6), 28-30.

Deschryver, M. D., & Yadav, A. (2015). Creative and computational thinking in the context of new literacies: working with teachers to scaffold complex technology-mediated approaches to teaching and learning. *Journal of Technology and Teacher Education, 23*(3), 411-431.

Duschl, R., Maeng, S., & Sezen, A. (2011). Learning progressions and teaching sequences: A review and analysis. *Studies in Science Education, 47*(2), 123-182.

Embretson, S. E., & Wetzel, C. D. (1987). Component latent trait models for paragraph comprehension. *Applied Psychological Measurement, 11*(2), 175–193.

Etkina, E., Karelina, A., Ruibal-Villasenor, M., Rosengrant, D., Jordan, R., & Hmelo-Silver, C.E. (2010). Design and reflection help students develop scientific abilities: learning in introductory physics laboratories. *Journal of the Learning Sciences, 19*(1), 54-98.

Fields, D. A., Giang, M. T., Kafai, Y. B. (2014). Programming in the Wild: Patterns of Computational Participation in the Scratch Online Social Networking Forum. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (WiPSCE '14). (pp. 2-11). New York, NY: ACM.

Fund, Z. (2010). Effects of communities of reflecting peers on student-teacher development including in-depth case studies. *Teachers and Teaching: Theory and Practice, 16*(1), 679-701.

Gal-Ezer, J., & Stephenson, C. (2014). A tale of two countries: Successes and challenges in K–12 computer science education in Israel and the United States. *ACM Transactions on Computing Education, 14*(2), 1-18.

Garcia, P. E., Olea, J., & de la Torre, J. (2014). Application of cognitive diagnosis models to competency-based situational judgment tests. *Psicothema, 26*(3), 372–377.

Gobert, J. D., & Clement, J. J. (1999). Effects of student-generated diagrams versus student-generated summaries on conceptual understanding of causal and dynamic knowledge in plate tectonics. *Journal of Research in Science Teaching, 36*(1), 39-53.

Gogol, K., Brunner, M., Goetz, T., Martin, R., Ugen, S., Keller, U., Fischbach, A., & Preckel, F. (2014). "My Questionnaire is Too Long!" The assessments of motivational-affective constructs with three-item and single-item measures. *Contemporary Educational Psychology, 39*(3), 188-205.

Gorin, J. S., & Mislevy, R. J. (2013). *Inherent measurement challenges in the Next Generation Science Standards for both formative and summative assessment*. Commissioned paper presented at the K-12 Center at ITS Invitational Research Symposium on Science Assessment, Washington DC.

Gouws, L., Bradshaw, K., & Wentworth, P. (2013, October 7–9). First year student performance in a test for computational thinking. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (pp. 271–277). East London, South Africa.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education, 25*(2), 199-237.

Grover, S. & Pea, R. D. (2013). Computational Thinking in K–12: A Review of the state of the field. *Educational Researcher, 42*(1), 38-43.

Grover. S. (2014). *Foundations for Advancing Computational Thinking: balanced designs for deeper learning in an online computer science course for middle school students* Retrieved from https://searchworks.stanford.edu/view/10592902

Haertel, E. H. (1989). Using restricted latent class models to map the skill structure of achievement items. *Journal of Educational Measurement, 26*(4), 301-321.

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), 309-334.

Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *ACM SIGCSE bulletin, 41*(1), 183-187.

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research, 77*(1), 81–112.

Huang, H.-Y., & Wang, W.-C. (2014). The random-effect DINA model. *Journal of Educational Measurement, 51*(1), 75–97.

Henson, R., Roussos, L., Douglas, J., & He, X. (2008). Cognitive diagnostic attribute-level discrimination indices. *Applied Psychological Measurement, 32*(4), 275-288.

Henson, R., Templin, J., & Willse, J. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika, 74*(2), 191–210.

Hohensinn, C., & Kubinger, K. D. (2011). Applying item response theory methods to examine the impact of different response formats. *Educational and Psychological Measurement, 71*(4), 732 - 746.

Hsu, T. - C., Chang, S. - C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*(1), 296-310.

Hutchins, N. M., Zhang, N., & Biswas, B. (2017). The role gender differences in computational thinking confidence levels plays in STEM applications. In Proceedings of The International Conference on Computational Thinking Education (CTE'17). (PP. 34-38). The Education University of Hong Kong: Hong Kong.

International Society for Technology in Education. (2016). ISTE national educational technology standards (NETS). Eugene, OR: International Society for Technology in Education.

Jacobson, E., Lobato, J., & Orrill, C. H. (2018). Middle school teachers' use of mathematics to make sense of student solutions to proportional reasoning problems. *International Journal of Science and Mathematics Education, 16*(8), 1541–1559.

Johnstone, C., Thompson, S., Bottsford-Miller, N., & Thurlow, M. (2008). Universal design and multimethod approaches to item review. *Educational Measurement: Issues and Practice, 27*(1), 25-36.

Jun, S., Han, S., & Kim, S. (2016). Effect of design-based learning on improving computational thinking. *Behaviour & Information Technology*, *36*(1), 1-11.

Junker, B. W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, *25*(3), 258-272.

Jurich, D. P., & Bradshaw, L. P. (2014). An illustration of diagnostic classification modeling in student learning outcomes assessment. *International Journal of Testing*, *14*(1), 49-72.

K-12 Computer Science Framework Steering Committee. (2016). K–12 computer science framework. Retrieved from http://www.k12cs.org

Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, *4*(3), 583-596.

Kane, M. T. (2013). Validating the interpretations and uses of test scores. *Journal of Educational Measurement, 50*(1), 1-73.

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia - Social and Behavioral Sciences, 47*(1), 1991-1999.

Kleiger, D. M., Bridgeman, B., Tannenbaum, R., & Olivera-Aguilar, M. (2018). *The validity of GRE General Test scores for predicting academic performance at U.S. law schools*. (Research Report ETS RR-18-26). Princeton, NJ: Educational Testing Service.

Kong, S., & Abelson, H. (Eds.). (2019). *Computational thinking education*. Singapore: Springer.

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the Computational Thinking Scales (CTS). *Computers in Human Behavior, 72*(1), 558-569.

Kuo, B., Chen, C., Yang, C., & Mok, M. (2016). Cognitive diagnostic models for tests with multiple-choice and constructed-response items. *Educational Psychology, 36*(6), 1115-1133.

Lee, I. (2016). Reclaiming the roots of CT. *CSTA Voice Magazine* (March). 3-5.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads, 2*(1), 32–37.

Lee, Y-S, Park, Y, & Taylan, D. (2011). A cognitive diagnostic modeling of attribute mastery in Massachusetts, Minnesota, and the U.S. national sample using the TIMSS 2007. *International Journal of Testing, 11*(2), 144-177.

Lewandowski, G., Bouvier, D. J., McCartney, R., Sanders, K., & Simon, B. (2010). Commonsense understanding of concurrency: computing students and concert tickets. *Communications of The ACM, 53*(7), 60-70.

Linn, M. C. (1985). The cognitive consequences of programming instruction in classrooms. *Educational Researcher, 14*(5), 25-29.

Liu, O., Bridgeman, B., Gu, L., Xu, J., & Kong, N. (2015). Investigation of response changes in the GRE revised general test. *Educational and Psychological Measurement, 75*(6), 1-19.

Lu, J. J. & Fletcher, G. H. (2009). Thinking about computational thinking. *ACM SIG CSE Bulletin, 41*(1), 260–264.

Margolis, J., Ryoo, J. J., Sandoval, C. D. M., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. *ACM Inroads, 3*(4), 72-78.

Maydeu-Olivares, A., & Joe, H. (2014). Assessing approximate fit in categorical data analysis. *Multivariate Behavioral Research, 49*(4), 305–328.

Mislevy, R. J. (2007). Validity by design. *Educational Researcher, 36*(8), 463–469.

Mislevy, R. J. & Riconscente, M. M. (2015). Evidence-centered assessment design. In Lane, S., Raymond, M. R., & Haladyna, T. M. (Eds.), *Handbook of test development* (pp. 61-90). Mahwah, NJ: Lawrence Erlbaum Associates.

Mouza, C., Marzocchi, A., Pan, Y., & Pollock, L. (2016). Development, implementation, and outcomes of an equitable computer science after-school program: Findings from middle-school students. *Journal of Research on Technology in Education, 48*(2), 84-104.

National Science Foundation (2018). *STEM + Computing K-12 Education (STEM+C).* Retrieved from https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=505006

National Research Council. (2010). Committee for the Workshops on Computational Thinking: Report of a workshop on the scope and nature of computational thinking. Washington, DC: National Academies Press.

National Research Council. (2011). Committee for the Workshops on Computational Thinking: Report of a workshop of pedagogical aspects of computational thinking. Washington, DC: National Academies Press.

NGSS Lead States (2013). *Next Generation Science Standards: For States, By States*. The National Academies Press, Washington, DC.

Nitko, A. (2004). *Educational assessment of students* (4th ed.). Upper Saddle River, N.J.: Merrill/Prentice Hall.

OECD (2014). *The PISA 2012 Results: Creative Problem Solving (Volume V): Students' Skills in Tackling Real-Life Problems*, PISA, OECD Publishing, Paris, http://dx.doi.org/10.1787/9789264208070-en.

OECD (2003). *The PISA 2003 Assessment Framework: Mathematics, Reading, Science and Problem Solving Knowledge and Skills*, PISA, OECD Publishing, Paris, http://dx.doi.org/10.1787/9789264101739-en.

Patel, N. H., & Herick, D. L. (2016, January). How can you ensure your assessments provide accurate feedback? *Association for Middle Level Education Magazine,* 8-12.

Seoane-Pardo, A. M. (2016). Computational thinking beyond STEM: an introduction to "moral machines" and programming decision making in Ethics classroom. In F. J. García-Peñalvo (Ed.), *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16) (Salamanca, Spain, November 2-4, 2016)* (pp. 37-44). New York, NY: ACM.

Polikoff, M. S. (2017). Is Common Core "working"? And where does Common Core research go from here? *AERA Open, 3*(1), 1-6.

Revelle, W. (2010). Package "psych." Retrieved March 2, 2019, from http:// personality-project.org/r/psych_manual.pdf

Robitzsch, A., Kiefer, T., George, A., C., & Uenlue, A. (2018). *CDM: Cognitive diagnosis modeling. R package version 6.3*. Retrieved March 2, 2019, from https://CRAN.R-project.org/package¼CDM

Román-González, M., Pérez-González, J., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72*(1), 678-691.

Rupp, A. A., & Templin, J. (2008). Unique characteristics of cognitive diagnosis models: A comprehensive review of the current state-of-the-art. *Measurement: Interdisciplinary Research and Perspectives, 6*(1), 219–262.

Sáez-López, G., Román-González, M, & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education, 97*(1), 129-141.

Sanford, J. F., & Naidu, J. T. (2016). Computational thinking concepts for grade school. *Contemporary Issues in Education Research (Online), 9*(1), 23.

Sanford, J. & Naidu, J. (2017). Mathematical modeling and computational thinking. *Contemporary Issues in Education Research (online), 10*(2), 159-168.

Sengupta, P., Kinnebrew, J., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351-380.

Strobl, C., Kopf, J., & Zeileis, A. (2015). Rasch trees: A new method for detecting differential item functioning in the Rasch model. *Psychometrika, 80*(2), 289-316.

Svetina, D., Gorin, J. S., & Tatsuoka, K. K. (2011). Defining and comparing the reading comprehension construct: A cognitive-psychometric modeling approach. *International Journal of Testing, 11*(1), 1-23.

Tatsuoka, K. K. (1990). Toward an integration of item response theory and cognitive error diagnosis. In N. Frederiksen, R. Glaser, A. Lesgold, & M. G. Shafto (Eds.), *Diagnostic monitoring of skill and knowledge acquisition* (pp. 453–488). Hillsdale, NJ: Lawrence Erlbaum Associates.

Templin, J., & Henson, R. A. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods, 11*(3), 287-305.

Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research, 57*(1), 3-31.

Turner, S. L. (2014). Creating an assessment-centered classroom: Five essential assessment strategies to support middle grades student learning and achievement. *Middle School Journal, 45*(5), 3-16.

Vandegrift, T. (2010). Commonsense understanding of concurrency: Computing students and concert tickets. *Communications of The ACM, 53*(7), 60-70.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715-728.

Wainer, H., Sheehan, K., & Wang, X. (2000). Some paths toward making praxis scores more useful. *Journal of Educational Measurement, 37*(2), 113-140.

Walk, C. M. (2011). What's the DIF? Why Differential Item Functioning analyses are an important part of instrument development and validation. *Journal of Psychoeducational Assessment, 29*(4), 364-376.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016).Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147.

Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The Fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12)*, 215-220. New York, NY: ACM.

Wing, J. M. (2011). Research Notebook: Computational thinking what and why? *The magazine of the Carnegie Mellon University School of Computer Science*. Retrieved from http://www.cs.cmu.edu/link/research-notebookcomputational-thinking-what-and-why.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), doi.org/10.1145/1118178.1118215.

Winter, P. C., Kopriva, R. J., Chen, C-S., & Emick, J. E. (2006). Exploring individual and item factors that affect assessment validity for diverse learners: Results from a large-scale cognitive lab. *Learning and Individual Differences, 16*(4), 267–276.

Wise, S. (2017). Rapid-guessing behavior: Its identification, interpretation, and implications. *Educational Measurement: Issues and Practice, 36*(4), 52-61.

Wolz, U., Stone, M., Pearson, K., Pulimood, S., & Switzer, M. (2011). Computational thinking and expository writing in the middle school, *ACM Transactions on Computing Education, 11*(2), 1-22.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 80*(4), 55-62. doi:10.1145/2994591

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. In Proceedings of ACM Special Interest Group on Computer Science Education, Dallas, TX.

Yaşar, O. (2018). A new perspective on computational thinking. *Communications of the ACM, 61*(7), 33-39. DOI:10.1145/3214354

Zeileis, A., Strobl, C., Wickelmaier, F., & Kopf, J. (2012). psychotree: Recursive partitioning based on psychometric models. R package version 0.12-2. Retrieved from http://CRAN.R-project.org/package=psychotree.

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research, 53*(4), 562-590.

# APPENDIX A. INSTITUTIONAL REVIEW BOARD (IRB) PERMISSION

PURDUE
UNIVERSITY

HUMAN RESEARCH PROTECTION PROGRAM
INSTITUTIONAL REVIEW BOARDS

---

| | |
|---|---|
| **To:** | |
| **From:** | |
| **Date:** | |
| **Committee Action:** | **Expedited Approval - Category**(6) (7) |
| **IRB Approval Date** | 01/09/2018 |
| **IRB Protocol #** | 1709019685 |
| **Study Title** | Development of Assessment Framework in Computational Thinking |
| **Expiration Date** | 01/08/2019 |
| **Subjects Approved:** | 12 |

The above-referenced protocol has been approved by the Purdue IRB. This approval permits the recruitment of subjects up to the number indicated on the application and the conduct of the research as it is approved.

The IRB approved and dated consent, assent, and information form(s) for this protocol are in the Attachments section of this protocol in CoeusLite. Subjects who sign a consent form must be given a signed copy to take home with them. Information forms should not be signed.

PURDUE
UNIVERSITY

HUMAN RESEARCH PROTECTION PROGRAM
INSTITUTIONAL REVIEW BOARDS

| | |
|---|---|
| **To:** | |
| **From:** | |
| **Date:** | 11/26/2018 |
| **Committee Action:(1)** | Determined Exempt, Category **(1)** |
| **IRB Action Date:** | 11 / 20 / 2018 |
| **IRB Protocol #:** | 1809021063 |
| **Study Title:** | Computational Thinking Assessment |

# APPENDIX B. CTCA ITEMS

Instructions: Please circle only **one** answer for each question. Feel free to mark on the answer sheet.

**Question 1**

The chart on the right shows that letters in blue can be represented by the letters or numbers in red. Can you fill in the blank?

| | |
|---|---|
| A=B | G=H |
| B=C | H=I |
| C=D | I=5 |
| D=E | J=6 |
| E=F | K=7 |
| F=G | L=8 |

????? = EFD7

| A. | B. | C. | D. |
|---|---|---|---|
| DESK | DECK | HIKE | HIGH |

**Question 2**

Which pair of words below have the same relationship as **Pigeon--Bird**?

| A. | B. | C. | D. |
|---|---|---|---|
| **Rose--Flower** | **Stove--Kitchen** | **Cat--Dog** | **Fish--Lake** |

**Question 3**

***List A*** contains five numbers: { 3, 7, 10, 15, 20 }.

**Step 1:** Move each number in *List A* that is smaller than 14 to a new list. Then, call the new list *List B*.

**Step 2:** Add 2 to every number in *List B*, then call it *List C*.

**Step 3:** In *List C*, if the numbers are larger than 6, place them in a new list, then call the new list *List D*.

**Step 4:** The sum of all the numbers in *List D*.

What is the result for **Step 4**?

| A. | B. | C. | D. |
|---|---|---|---|
| 20 | 21 | 22 | 23 |

**Question 4**

A school requires students to pass BOTH a math exam AND an English exam to become a member of its choir. This table contains the information for four students. Can you fill in the blanks?

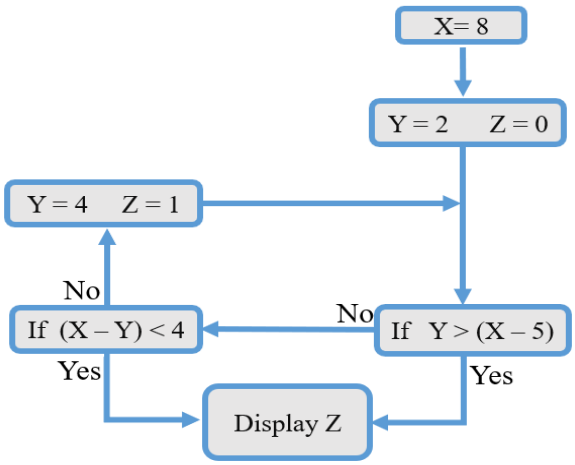
| Student ID | Passed math exam? | Passed English exam? | Is the student a member of the choir? |
|---|---|---|---|
| 001 | No | No | No |
| 002 | No | Yes | **Blank 1??** |
| 003 | Yes | No | **Blank 2??** |
| 004 | Yes | Yes | Yes |

| A. | B. | C. | D. |
|---|---|---|---|
| **Blank 1**: Yes | **Blank 1**: Yes | **Blank 1**: No | **Blank 1**: No |
| **Blank 2**: No | **Blank 2**: Yes | **Blank 2**: No | **Blank 2**: Yes |

**Question 5**

A computer takes in a number (X = 8), as shown in the chart below. What number will be displayed at the end?

(**Note**: follow the arrows carefully)

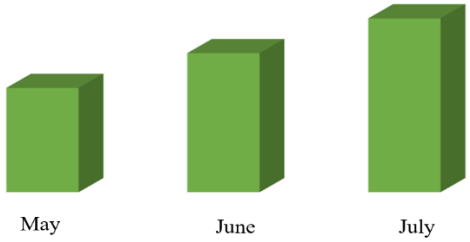



| A. | B. | C. | D. |
|---|---|---|---|
| 1 | 0 | 5 | 8 |

**Question 6**

This table shows how many boxes of cookies a store sold last May, June, and July. The graph on the right gives the same information.

| May | June | July |
|---|---|---|
| 12 boxes | 16 boxes | 20 boxes |




Based on this example, showing how many cups of coffee were sold in a store, fill out the blanks (??) for the table below.

| Friday | Saturday | Sunday |
|---|---|---|
| ?? | ?? | ?? |


A. 20 cups 60 cups 65 cups

B. 20 cups 60 cups 40 cups

C. 20 cups 60 cups 10 cups

D. 20 cups 40 cups 80 cups

**Question 7**

A list has the following numbers: {6 1 2 3 4 5}.

The goal is to sort them in **increasing** order: {1 2 3 4 5 6}. Can you fill in the blank?

| | |
|---|---|
| Step 1 | 6 1 2 3 4 5 |
| Step 2 | 1 6 2 3 4 5 |
| Step 3 | 1 2 6 3 4 5 |
| Step 4 | **??????** |
| Step 5 | 1 2 3 4 6 5 |
| Step 6 | 1 2 3 4 5 6 |

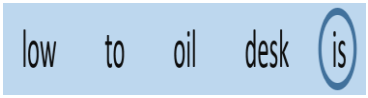
| A. | B. | C. | D. |
|---|---|---|---|
| 1 2 3 6 4 5 | 1 2 3 6 5 4 | 1 3 6 2 4 5 | 1 3 6 4 2 5 |

**Question 8**

Among the five words below, two rules lead you to ONLY the circled word.
One rule is given; what is the second rule?

Rule 1: The length of the word is less than or equal to three letters.

Rule 2: **???**

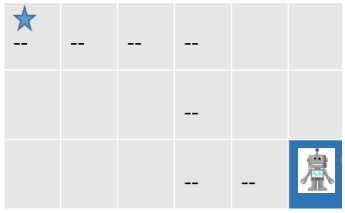



A. The length of the word is two letters.
B. The length of the word is less than 6.
C. The word contains the letter I.
D. The word cannot contain the letter O.

**Question 9**

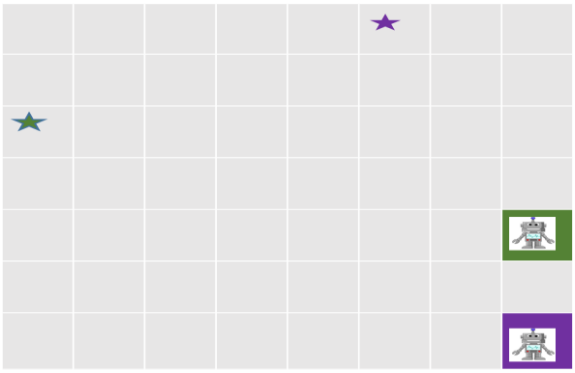
A robot takes **one second** to travel **one cell.** It departs from the blue cell with the command **L2 U2 L3**.

That means the robot will travel 2 cells left, 2 cells up, and 3 cells left (see the map on right). At the **7th** second, the robot arrives at the blue star cell.



One robot departs from the green cell with the command **L3 U2 L4.** It will arrive at the green star cell. (Please mark on chart).

Another robot departs from the purple cell with the command **L2 U6**. It will arrive at the purple star cell. (Please mark on chart).

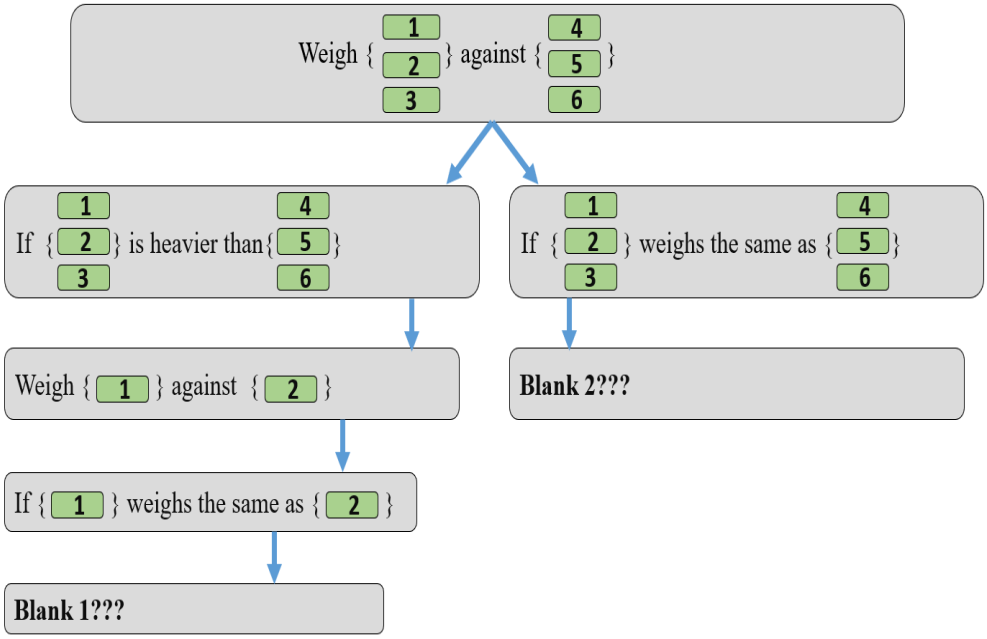


**They depart at same time.** It takes both robots one second to travel one cell. At which second will they meet at the **same cell**?

| A. | B. | C. | D. |
|---|---|---|---|
| 2th | 4th | 3th | Never |

**Question 10**


The **EIGHT** blocks above look the same. But **ONLY ONE** of the blocks is heavier. By using the outline below, Juan has figured out which of the blocks is heaviest. Please fill in the blanks.




A.

**Blank 1:** He will decide { 3 } is heavier

**Blank 2:** He will weigh { 7 } against { 8 }

B.

**Blank 1:** He will weigh { 1 } against { 8 }

**Blank 2:** He will weigh { 7 } against { 8 }

C.

**Blank 1:** He will decide { 3 } is heavier

**Blank 2:** He will weigh { 6 } against { 7 }

D.

**Blank 1:** He will weigh { 1 } against { 8 }

**Blank 2:** He will decide { 7 } is heavier.

**Question 11**

If  =  ,

then  = **???**

A. 

B. 

C. 

D. 

**Question 12**

Maria draws one card at a time in a game. She will receive two bonus points if **the first three cards** she draws meet the following requirements:

*The first card must be a yellow card.*
*The second card must be a blue card.*
*The third card must NOT be a blue card.*

She played two rounds. Can you fill out the blanks in the table below?

| | Is the first card yellow? | Is the second card blue? | Is the third card blue? | Result |
|---|---|---|---|---|
| Round 1 | Yes | Yes | **Blank 1???** | She received bonus points |
| Round 2 | Yes | No | No | **Blank 2???** |

A.

**Blank 1** = No

**Blank 2** = She received bonus points

C.

**Blank 1** = Yes

**Blank 2** = She received bonus points

B.

**Blank 1** = No

**Blank 2** = She did NOT receive bonus points

D.

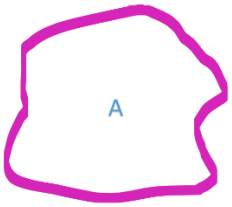
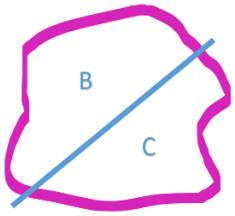
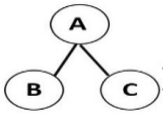
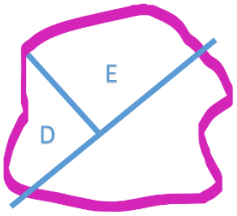
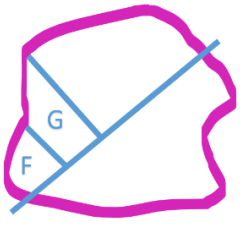
**Blank 1** = Yes

**Blank 2** = She did NOT receive bonus points

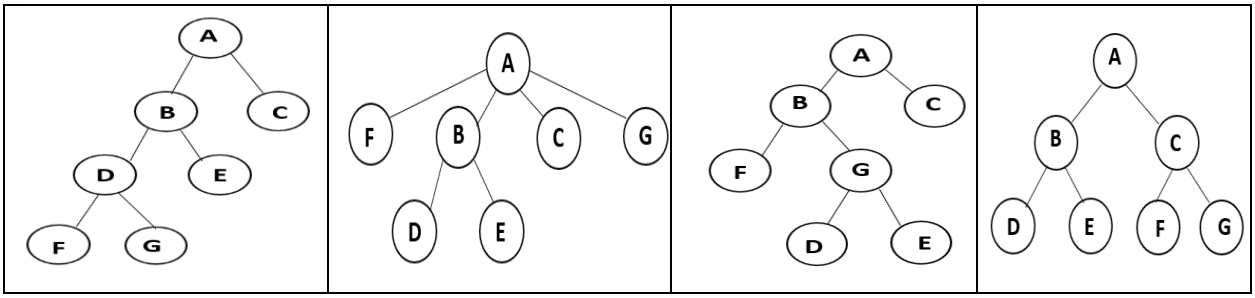
**Question 13**

Please observe below that in every step, the picture on the left represents the graph at right.

Fill out the blank for **Step 4**. (Note: **NOT** Step 3)

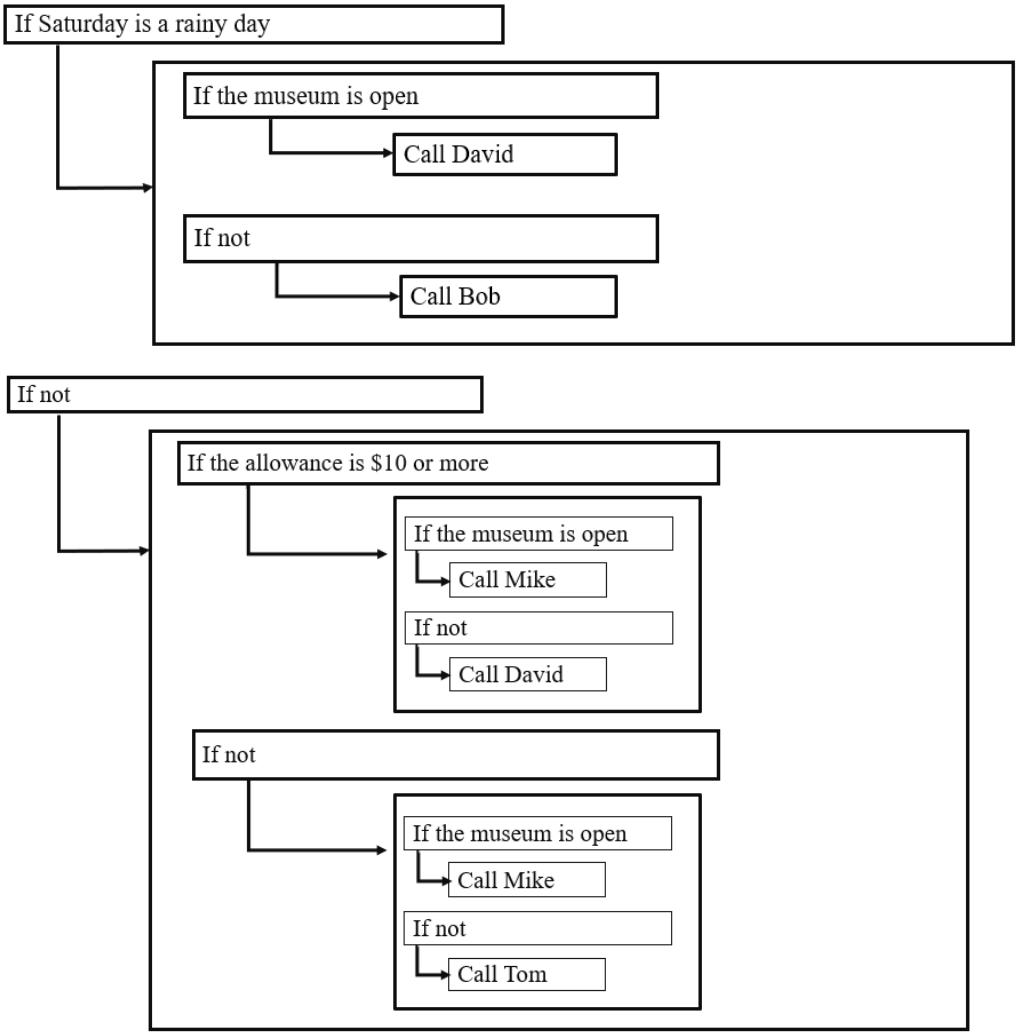
| | **Picture** | **Graph** |
|---|---|---|
| **Step 1** |  |  |
| **Step 2** |  |  |
| **Step 3** |  | |
| **Step 4** |  | **???** |

A. B. C. D.




**Question 14**

Last Saturday was **NOT** a rainy day, Kyle received **$6** allowance, and the museum was **NOT** open. Based on the sequence of events below, what did Kyle end up doing? (**Note**: read each statement carefully)



A. Call Bob
B. Call both Bob and Tom
C. Call both Bob and David
D. Call Tom

**Question 15**

Blue Rabbit follows the instruction:

**left, down, down, down, left, left** to arrive the blue cell.


Yellow Rabbit follows the instruction:

**left, up, up, left, up, left** to arrive the yellow cell.

**Note:**

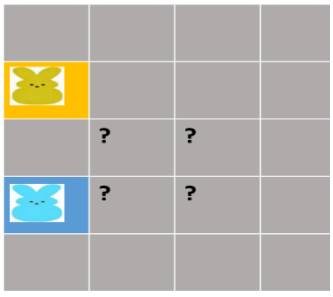
Every time a rabbit passes through a cell, the light switch changes.

(As you go, mark each move on the chart below).

**Final Destinations:**

For these four cells,

can you tell whether the light switch is **on** or **off** on this map?




A. B. C. D.