RELATIONSHIPS BETWEEN LEARNER CHARACTERISTICS AND COMPUTATIONAL THINKING FOR MIDDLE-SCHOOL STUDENTS IN TWO DIFFERENT CONTEXTS

Tugba Ayer
Georgia State University
ACCEPTANCE
This dissertation, RELATIONSHIPS BETWEEN LEARNER CHARACTERISTICS AND
COMPUTATIONAL THINKING FOR MIDDLE-SCHOOL STUDENTS IN TWO
DIFFERENT CONTEXTS, was prepared under the direction of the candidate’s Dissertation
Advisory Committee. It is accepted by the committee members in partial fulfillment of the
requirements for the degree, Doctor of Philosophy, in the College of Education and Human
Development, Georgia State University.

The Dissertation Advisory Committee and the student’s Department Chairperson, as
representatives of the faculty, certify that this dissertation has met all standards of excellence and
scholarship as determined by the faculty.

Jonathan D. Cohen, Ph.D.
Committee Chair

Brendan Calandra, Ph.D.                      Maggie Renken, Ph.D.
Committee Member                           Committee Member

Teri Holbrook, Ph.D.
Committee Member

Date

Brendan D. Calandra, Ph.D.
Chairperson, Learning Technologies Division

Paul A. Alberto, Ph.D.
Dean, College of Education and
Human Development
AUTHOR’S STATEMENT

By presenting this dissertation as a partial fulfillment of the requirements for the advanced degree from Georgia State University, I agree that the library of Georgia State University shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to quote, to copy from, or to publish this dissertation may be granted by the professor under whose direction it was written, by the College of Education and Human Development’s Director of Graduate Studies, or by me. Such quoting, copying, or publishing must be solely for scholarly purposes and will not involve potential financial gain. It is understood that any copying from or publication of this dissertation which involves potential financial gain will not be allowed without my written permission.

__________________________
Tugba Ayer
NOTICE TO BORROWERS

All dissertations deposited in the Georgia State University library must be used in accordance with the stipulations prescribed by the author in the preceding statement. The author of this dissertation is:

Tugba Ayer
Instructional Technology
College of Education and Human Development
Georgia State University

The director of this dissertation is:

Jonathan D. Cohen
Learning Technologies Division
College of Education and Human Development
Georgia State University
Atlanta, GA 30303
CURRICULUM VITAE

Tugba Ayer

ADDRESS: 470 Waterhaven Ln.
Alpharetta, GA 30004

EDUCATION:

<table>
<thead>
<tr>
<th>Degree</th>
<th>Year</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ph.D.</td>
<td>2018</td>
<td>Georgia State University Learning Technologies</td>
</tr>
<tr>
<td>Masters Degree</td>
<td>2011</td>
<td>Edgewood College Master of Arts in Educational Studies</td>
</tr>
<tr>
<td>Bachelors Degree</td>
<td>2008</td>
<td>Istanbul Bilgi University Computer Science</td>
</tr>
<tr>
<td>Bachelors Degree</td>
<td>2007</td>
<td>Istanbul Bilgi University Business Administration</td>
</tr>
</tbody>
</table>

PROFESSIONAL EXPERIENCE:

<table>
<thead>
<tr>
<th>Year</th>
<th>Position</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-2018</td>
<td>Instructor</td>
<td>Learning Technologies Division, College of Education and Human Development, Georgia State University, Atlanta, Georgia</td>
</tr>
<tr>
<td>2015-2018</td>
<td>Research Assistant</td>
<td>Learning Technologies Division, College of Education and Human Development, Georgia State University, Atlanta, Georgia</td>
</tr>
</tbody>
</table>

PRESENTATIONS AND PUBLICATIONS:


**PROFESSIONAL SOCIETIES AND ORGANIZATIONS**

<table>
<thead>
<tr>
<th>Year</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016</td>
<td>American Educational Research Association</td>
</tr>
<tr>
<td>2017</td>
<td>Society for Technology and Teacher Education</td>
</tr>
</tbody>
</table>
Computational thinking (CT) is a complex problem-solving process that develops over an extended period of time. After Wing (2006) popularized CT as a concept, there has been growth in studies of it, with the majority taking place in computing courses. Although previous research has demonstrated the relationship between learner characteristics and programming success in higher education (Watson et al., 2014), a comprehensive approach to understand the relationships between learner characteristics and CT in computing courses in K-12 education is lacking.

The aim of this dissertation was to address this gap by exploratory analysis to determine how a set of learner characteristics were related to a group of middle school students’ CT, and to determine which of these factors had the strongest association with participants’ CT in two computer science educational contexts.
This research took place in two study sites in different districts in the U.S. In total, 314 students participated in this research. Students completed a CT quiz and a learner profile survey and developed digital artifacts in an app-building computing course. Artifact analysis was conducted to examine the CT practices in the artifacts. Correlational analysis followed by regression analysis was used to examine the relationships between student variables, including self-efficacy, interest, prior experience with creative computing, and goal orientation, and CT measures such as quiz score and CT practice, after controlling for gender, and grade level.

The results of this study demonstrated that self-efficacy had a significant relationship with CT on both of the study sites. The regression analysis showed that none of the other learner characteristics explained significant amount of variation of CT. However, among the control variables, only gender had a significant correlation with CT practices profile; there were significantly more male than female students who demonstrated CT practices in their digital artifacts. Taken together, the findings of this study have provided evidence on which learner characteristics are related to CT for middle school-aged students. Instructional designers, educators, and researchers should consider these learner characteristics in their design in CT-infused, middle school computing courses.

INDEX WORDS: Computational Thinking, Learner Characteristics, Middle School
RELATIONSHIPS BETWEEN
LEARNER CHARACTERISTICS AND COMPUTATIONAL THINKING
FOR MIDDLE-SCHOOL STUDENTS IN TWO DIFFERENT CONTEXTS:
FORMAL AND INFORMAL LEARNING ENVIRONMENTS

By
TUGBA AYER

A DISSERTATION
Presented in Partial Fulfillment of Requirements for the
Degree of
Doctor of Philosophy
in
Instructional Technology
in
the College of Education and Human Development
Georgia State University
Atlanta, GA
2018
ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Dr. Jonathan Cohen for striking an excellent balance letting me pursue my research interest while giving me very useful guidance. I would also like to express my sincerest gratitude to my department chair and committee member, Dr. Brendan Calandra, for his constant support during my Ph.D. years. Without his support, guidance, and encouragement, this research would not have been possible. I would also like to thank my other committee members—Dr. Teri Holbrook, for her patience, willingness to help, and timely feedback on my writing, and Dr. Maggie Renken, for providing me critical feedback.

My special thanks go to my high school mathematics teacher, Sevim Ozel for her constant support and hard work, and to my undergraduate computer science department chair, Chris Stephenson, for believing in me and encouraging me to get a Ph.D.

Finally, I thank my husband, Turgay Ayer, and my mom, Melek Ketenci, for encouraging me to achieve my dreams, and for supporting me in any ways they could to make this happen.
# Table of Contents

LIST OF TABLES ........................................................................................................................................ vii

LIST OF FIGURES ...................................................................................................................................... viii

CHAPTER 1 INTRODUCTION ................................................................................................................... 1

Purpose ................................................................................................................................................ 1

Computational Thinking Framework ................................................................................................. 3

The Study .............................................................................................................................................. 6

CHAPTER 2 REVIEW OF THE LITERATURE ......................................................................................... 8

Definition of Computational Thinking ............................................................................................. 8

Computational Thinking in K-12 Education ..................................................................................... 9

Development of Computational Thinking in the K-12 Programming Context ................................ 12

Computational Thinking Assessments .............................................................................................. 13

Previous Studies Regarding Learner Characteristics in Computer Science Education .................... 15

Control Variables in this Study ............................................................................................................ 18

CHAPTER 3 METHODOLOGY ............................................................................................................... 20

Overview of the Research .................................................................................................................. 20

Design ................................................................................................................................................ 22

Study 1 ................................................................................................................................................ 23

*Site and Participants.* ...................................................................................................................... 23

*Lessons, Procedure, and Data Collection.* .................................................................................... 23

Study 2 ................................................................................................................................................ 27
Primary Research Question: What are the Relationships between Learner Characteristics and CT Quiz and Practices?

Correlation between CT-App Quiz and Learner Characteristics. ........................................ 57

Correlation between CT Practices and Learner Characteristics. ................................. 57

Secondary Research Question: How much of the variation in CT Quiz scores do selected variables explain in Study 1 and Study 2? ......................................................... 60

Secondary Research Question: How much of the variation in CT practices do selected variables explain in Study 1 and Study 2? ......................................................... 61

CHAPTER 5 DISCUSSION................................................................................................. 64

Review of the Research Questions ................................................................................... 65
LIST OF TABLES

Table 1 .......................................................................................................................................... 4
Table 2 ........................................................................................................................................ 38
Table 3 ........................................................................................................................................ 56
Table 4 ........................................................................................................................................ 57
Table 5 ........................................................................................................................................ 58
Table 6 ........................................................................................................................................ 59
Table 7 ........................................................................................................................................ 61
Table 8 ........................................................................................................................................ 62
Table 9 ........................................................................................................................................ 63
Table 10 ...................................................................................................................................... 65
Table 11 ...................................................................................................................................... 71
LIST OF FIGURES

Figure 1. Favorite Artist App ..................................................................................................... 24

Figure 2. Calculator worked example/worked example ........................................................ 25

Figure 3. Sample app I developed for illustration purposes ................................................ 26

Figure 4. Sample Question 1: Operations from CT concepts, Troubleshooting/Debugging from CT Practices .................................................................................................................. 30

Figure 5. Sample Question 2: Sequence from CT concepts, Being Iterative from CT Practices. ................................................................................................................................................. 31

Figure 6. Sample Question 3: Event from CT concepts, Reuse/Remix from CT Practices ...... 32

Figure 7. Endangered Animals and Body Mass Index Calculator App ................................. 40

Figure 8. A sample of Code from Calculator Worked example ........................................ 41

Figure 9. Student Sample Code ............................................................................................ 41

Figure 10. Sample Project for Using New Code Blocks ...................................................... 42

Figure 11. Code block same with design change ................................................................. 43

Figure 12. Adding the similar code block for additional design elements ......................... 43

Figure 13. Apps without any code block ............................................................................. 43

Figure 14. Scatterplot for Standardized Residuals vs. Predicted Value ............................... 49

Figure 15. Normal P-P Plot of Regression Standardized Residuals .................................... 50

Figure 16. Scatterplot for Standardized Residuals vs. Predicted Value ............................... 51

Figure 17. Q-Q Plot of CT Quiz ........................................................................................... 52

Figure 18. Participant flow through Study 1 and Study 2 data collection at four time points: Spring 2017, pretest and posttest; Fall 2017, pretest and posttest ........................................... 54

Figure 19. Mediation model ................................................................................................. 67
CHAPTER 1
INTRODUCTION

A decade ago, Jeanette Wing (2006) coined the term *computational thinking*, stating that computational thinking (CT) compiles high-level skills in such a way that one can define and provide solutions for a problem (Cuny, Snyder, & Wing, 2010). Wing and other computer scientists (e.g., Yasar, 2018) maintained that CT is an important skillset not only for those working in computing, but it also can be successfully applied across different areas of knowledge (Buitrago et al., 2017; Henderson, Cortina, Hazzan, & Wing, 2007; Grover & Pea, 2013). Since that time, CT has received a considerable amount of attention, and most of the computational thinking researchers agree that CT is in line with many aspects of 4Cs of education—Communication, Critical Thinking, Collaboration, and Creativity framed by National Education Association (NAE, 2002)—of 21st century skills (e.g., Lye & Koh, 2014). Now the dilemma for educators is to incorporate CT into K-12 education. In order to address this issue, a few states (e.g., Iowa, Maryland, and Virginia) are currently working with non-profit organizations such as CodeVA (https://www.codevirginia.org/) to develop standards, curricula, and professional development programs to integrate CT into the curriculum of such content areas as English, Mathematics, and Science (CSTA, 2016; ISTE, 2016).

Purpose

While there is a surge of initiatives to teach CT in computing courses and numerous education tools for reinforcing CT, only a handful of studies consider effective instructional design for teaching (Dawson et al., 2018) and assessing CT (Gouws, Bradshaw, & Wentworth, 2013; Grover, 2017). More specifically, while abundant evidence indicates that learner characteristics--including prior knowledge, self-efficacy, and interest--are strong predictors of students’ academic outcomes in programming courses (Beyer, 2014; Biggs, Kember, & Leung, 2001; Grover, Pea, & Cooper, 2016; Haungs, Clark, Clements, & Janzen, 2012; Lishinski, Yadav, & Enbody, 2016; Wiedenbeck,
2005), very few studies have examined the importance of learner characteristics such as self-efficacy and interest in middle-school computer science (CS) classrooms (e.g., Grover, Pea, & Cooper, 2016). Thus, it remains unclear how individual differences may impact the development of CT skills in K-12 education, and how the instructional design should be organized to reflect those differences in the instruction. In addition, of the existing studies, only a few have considered the role of age (e.g., Román-González, Pérez-González, Moreno-León, & Robles, 2016), in spite of the fact that earlier studies indicated that grade and age-appropriate curriculum design for CT might maximize its influence on K-12 education (Brennan & Resnick, 2012; Dweck, 2008; Wing 2006).

This lack of understanding of the roles of individual characteristics in effective acquisition of CT skills has resulted in “one size fits all” curricula to teach CT concepts and practices in today’s K-12 classrooms, possibly leading to CT being inaccessible to many students (Black & Deci, 2000; Dede 2005; Beyer, 2015; Dawson et al., 2017). More specifically, the literature shows that certain learner characteristics such as self-efficacy, interest, goal orientation, and prior creative computing experience were well studied in college-level CSEd but no prior study has considered the relationship between these learner characteristics and students’ computational thinking concepts and practices in a computational problem-solving (cPBL) environment in middle-schools while controlling for students’ gender, and grade level.

While many explanations are possible for this research gap on learner characteristics with K-12 students, one reason may be a scarcity of valid and reliable instruments for measuring CT (Grover & Pea, 2013; Roman-Gonzalez et al., 2015). As Grover and Pea (2014) noted, without an appropriate assessment, CT has little chance of becoming part of K-12 education because the effectiveness of any instructional design and the assessment of students’ CT depend on the reliable instruments. To address this concern and to add to the literature on computer science education, I developed a multiple-choice CT assessment instrument based on MIT’s App Inventor tool (http://ai2.appinventor.mit.edu), a web-based, drag-and-drop block-based Android programming
environment. Through the use of this assessment instrument after the implementation of an instructional design that aimed to help a group of middle-school students to learn programming using App Inventor, I aimed to examine the relationships between learner characteristics and CT.

The purpose of this research, therefore, is to shed light on learner characteristics related to middle-school students’ CT concepts and practices. In addition to the custom-built CT assessment instrument, I used a learner characteristics survey and artifact analysis to answer the following main research question: What is the nature of the relationships (if any) between students’ learner characteristics and their demonstrated CT concepts and practices for middle-school-aged students in a formal and an informal context?

**Computational Thinking Framework**

To answer the research question, this research took place in a programming context. Using programming as a tool to reinforce CT in the formal and informal classrooms required me to determine what exactly students learn when they are designing apps and how their learning relates to CT (Kafai, Burke, & Resnick, 2014). In order to do that, I decided to employ Yadav, Gretter, Good, and McLean’s (2017) CT definition which operationalized CT as the thought process that involves breaking down complex problems into manageable smaller chunks of the problem, using algorithms (a sequence of steps) to solve problems, transferring the solution to similar problems, and determining if an intelligent agent can effectively carry out the solution. I chose this operational definition of CT because it is widely employed in K-12 education computing classes. This might be due to the fact that this definition encompasses how CT is taking place in computing class in general terms. Other definitions include processes that might not be happening in the computing courses. For example, data collection is listed as one of the tenets of CT in Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) definition; however, data collection is not something that happens frequently in computing curriculum.
With this in mind, I used Brennan and Resnick’s (2012) CT framework because it narrows down Yadav, Gretter, Good, and McLean’s general definition of CT. Besides this, CT practices in this framework matched the CT definition (see Table 1 for further information). Last, Brennan and Resnick’s CT framework is more specific to the programming context when I compared it with other CT frameworks (e.g., Weintrop et al., 2016).

Table 1

CT Definition and CT Framework Practices

<table>
<thead>
<tr>
<th>Yadav, Gretter, Good, and McLean’s Operational CT Definition</th>
<th>CT Practices from Brennan &amp; Resnick’s Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breaking down complex problems into manageable smaller chunks of the problem,</td>
<td>Abstracting and modularizing</td>
</tr>
<tr>
<td>Using algorithms (a sequence of steps) to solve problems,</td>
<td>Being incremental and iterative</td>
</tr>
<tr>
<td>Transferring the solution to similar problems</td>
<td>Reusing and remixing other’s project</td>
</tr>
<tr>
<td>Determining if an intelligent agent can effectively carry out the solution</td>
<td>Debugging and testing the existing project</td>
</tr>
</tbody>
</table>

Brennan and Resnick’s (2012) computational thinking framework has three dimensions: computational concepts, practices, and perspectives. Computational concepts include the use of such concepts as sequences, loops, parallelism, events, conditionals, operators, and data while programming. Sequence refers to systematic design when creating a program, referring to the order of steps taken to reach the solution for a given problem. Looping is an efficient way of running the computer code; when an event appears again and again, students can use a loop to reiterate the solution for the same event, eliminating the need for repeated thinking for the same situation. Parallelism enables students to launch two or more stacks of events at the same time. For example, students can program a button on an app that, when clicked, can change the shape or size of two objects simultaneously. Event refers to an action that triggers other things to happen. For instance, “when key is pressed” and “when button is clicked” are examples of the event handlers for Scratch
and App Inventor, respectively. Conditionals give more flexibility to students while coding and allow them to change the outcome of their program based on given predefined conditions. Operators are useful in terms of “enabling students to perform numeric and string manipulations” (p. 5). Data helps to play with the values. Lists and variables are two main forms of holding values and updating them.

Computational practices include the development of programming strategies while coding, such as abstracting and modularizing, reusing and remixing others’ projects or debugging and testing the existing project, and being incremental and iterative. Abstracting and modularizing refer to dividing a large task into subtasks for more clear and precise solutions. Programmers are expected to solve the small chunks of the problem individually and then combine them together to reach to the specific purpose. Reusing/remixing the existing project requires students to connect their previous experiences with the new ones in order to address the new problem. Debugging and testing the existing project respectively refer to fixing any errors in the code if necessary and executing the code to observe the outcome. Being incremental and iterative requires students to build up the programs in a sequential order and to use loops if any iterations are needed.

The final dimension, computational perspectives, refers to students’ vision of themselves and the world around them as they engage with computing (Kafai & Peppler, 2011). Brennan and Resnick (2012) outlined computational perspectives in their studies as expressing themselves through artifacts, connecting with others, and questioning the limitations of their design.

With this CT framework, researchers can examine students’ learning of concepts, practices, and perception of computing. For this purpose, I developed an instrument, CT-App, in accordance with this framework, and supplemented it with a computational problem-solving activity to evaluate students’ CT.
The Study

The previous research in CSEd on learning has confirmed the importance of the learner characteristics in understanding the student’s performance in CS but the majority of the previous research took place in higher education. However, college students, especially those in computing courses, are not representative of a large portion of society. Many children do not have opportunities to engage in computing (Margolis, Estrella, Goode, Holme, & Nao, 2010) and are, therefore, not represented later on in university-level computing courses. Thus, there is a need for studies that examine if similar relations persist in K-12 education or not. Besides this, since CT is broader than programming, it is important to understand the relationship between learner characteristics and CT starting at an early age in a computing context. This current research aimed to close that gap by exploring the relationship between a wide range of variables and middle-school students’ CT skills.

In addition, this study was also methodologically unique when compared to previous studies that focused on CT because it was implemented in two different contexts (Román-González, Pérez-González, Moreno-León, & Robles, 2016), and two assessment types were used to address the research questions (Werner, Denner, Campe, 2012).

Almost all of the previous studies that brought together the variables that may have a relation with CS learning and CT took place in one type of environment (e.g., Durak & Saritepe, 2018), either in informal learning environments or formal classrooms. The variables that were found significantly correlated with CT in an informal context may not hold in the formal education context. For this reason, this research was conducted in both learning contexts: formal with mandatory participation and afterschool settings with voluntary participation.

Furthermore, this study was more complex than previous studies because two different assessment types were used to understand how the learner characteristics were related to CT as measured by written and performance-based tests. Although the literature suggests that CT assessment should be in the variety of forms (Brennan & Resnick, 2012; Grover 2017), previous
studies used only one assessment format. For example, Román-González, Pérez-González, Moreno-León, & Robles (2016) used only a multiple-choice exam. On the other hand, Wenner, Denner, and Shampe (2012) used only a performance-based assessment known as Fairy Assessment.

This research is likely to make a contribution to the empirical evidence in the field of computational thinking in a computing education context by examining the relationships between learner characteristics and middle-school-aged students’ CT in formal and informal learning environments. In the next section, the review of the related literature is presented. It starts with a summary of the different definitions of CT in the literature, then a broad view of the programs that focused on CT in K-12 education is presented. Next, more specifically, the studies that took place in a computing context and their way of assessing the CT are synthesized. After that, the previous studies regarding learner characteristics that were found significantly related to computing education are summarized. It ends with situating this research in the literature.
CHAPTER 2

REVIEW OF THE LITERATURE

The goal of this study is to examine the nature of the relation between students’ CT and learner characteristics. The study was informed by a review of the literature, reported below, of two key domains: CT in K-12 domain and the relationship between the learner characteristics and programming success in higher level computer science education. Specifically, I begin this review with a broad description of CT in K-12 education, CT in K-12 programming domain, and CT assessments found in the literature. Then, the related literature based on selected variables is explored, including: self-efficacy, interest, goal orientation, and prior experience with creative computing tools. Finally, the literature regarding the variables that were controlled for in this research is also synthesized.

Definition of Computational Thinking

More than a decade ago, Jeannette Wing (2006) coined the term computational thinking (CT) in her prominent article in the Communications of the Association of Computing Machinery where she argued that CT provides a particular lens to understand the problem and its solution (Kafai et al., 2014). Wing (2006) defined that CT provides certain types of thinking patterns conducive to “learning and problem-solving in computing-related subjects or even everyday reasoning, and that while it is necessary for computer scientists, it is also a universal skill set that everyone needs, thanks in part to widespread application of computers and computing” (Chen et al., 2017, p. 163). CT has since been defined in many different ways. The CSTA and ISTE defined CT as a problem-solving process that involves problem formulation; data collection, analysis, and representation; decomposition of the problem; abstraction; algorithms; automation; implementation of most efficient solution in terms of resources and steps; and transfer of this process to a wide variety of problems (Barr & Stephenson, 2011). Yadav, Gretter, Good, and McLean’s (2017) interpretation of CT, which I have chosen to adopt for this research, operationalizes CT as the thought process that
involves breaking down complex problems into manageable smaller chunks, using algorithms (a sequence of steps) to solve problems, transferring the solution to similar problems, and determining if an intelligent agent can effectively carry out the solution.

**Computational Thinking in K-12 Education**

In 2011, Wing extended the definition of CT. In an extended CT definition, she emphasized that using computers is not the only way of developing CT or teaching “computational thinking without computers” (Kafai et al., 2014, p. 8). Similar to Kafai et al.’s ideas, Wing argued that CT helps students apply computer science principles to other fields such as science, mathematics, and English and it is not always necessary to have computers involved in order to develop CT. Thus, Wing’s claims regarding the benefits of CT are two-fold: application of computing ideas to enhance work in other fields and application of those ideas in daily life (Guzdial, 2015). One example of Wing’s former claim is the contribution of using computers in biology which led to the emergence of the bioinformatics field, which proponents claim helped humanity to understand problems better and formulate the solutions efficiently. An example of Wing’s latter claim is the system of filling the dishwasher developed for efficient clearing out process by stacking similar plates together so that they can be carried all together with a one-time effort. In addition to Wing’s claims, Kafai et al. (2014) also stated that CT is:

> a catch-all term for what understanding computer science can contribute to the increasingly digital world… computational thinking has become the rallying cry for those who study what youth need to know about computer science and what it means to think systematically about solving all types of problems, big and small. (p. 7)

After Wing’s redefinition of CT, many researchers have studied on how to better operationalize CT. One of the well-known works in this area is the dissertation entitled “Foundations for Advancing Computational Thinking” by Grover (2014). In her dissertation, Grover (2014) summarized the elements of CT that can be used as a basis for a CT-infused curricula and
assessment: “systematic processing information, symbols systems and representations, algorithmic notions of flow of control, structured problem decomposition, iterative, recursive, and parallel thinking, conditional logic, abstractions and pattern generalizations, efficiency and performance constraints, and debugging and systematic error detection” (p. 2).

Israel et al. (2015) suggested that teaching CT to all students can address the longstanding issue of the poor representation of women and minorities in the computational fields by reaching the widest audience (Ericson & Guzdial, 2014; Grover & Pea 2013; Grover, 2014). National data have shown that only 44% of high school seniors attending high schools took computer science courses (Change the Equation, 2016), and integrating CT into the K-12 curriculum can help to address the effort to serve underrepresented youth (Margolis et al., 2010). Barr and Stephenson (2011) also underlined the importance of integrating CT in K-12 education mainly because CT helps to develop problem-solving and critical thinking skills. In addition, Yadav et al. (2017) suggested that CT offers students an opportunity to “create, design, and develop technologies, tools, or systems that will be instrumental in advancing any field in the future” (p. 207). It should also be highlighted that CT is not the same as programming, as Fletcher and Lu (2009) suggested that CT allows people to think systematically and efficiently while processing information and tasks with or without computers.

Recently, there has been an increasing number of curriculum initiatives to embed CT in K-12 classrooms. Most of the initiatives have taken place in computer science classrooms (Armoni, 2016) since this is one of the efficient ways of incorporating CT into education (Kafai & Resnick, 1996). For example, College Board (2014) developed a whole curriculum and implemented under an Advanced Placement Computer Science Principles course based on CT practices. Google provided after-school club type of curriculum which teaches programming and CT. In addition to these, there were also initiatives that targeted underrepresented groups in computer science such as Girls Who Code (Saujani, 2017).
However, not all schools offer computing courses, and this leads to unequal opportunities for developing CT among students. In order to make CT accessible for all, some states (e.g., Arkansas, California, Georgia, Idaho, Indiana, Iowa, Maryland, Massachusetts, Nebraska, Nevada, New Jersey, North Carolina, Utah, and Washington) cooperated with non-profit organizations such as Computer Science Teachers Association to develop standards and to integrate CT into the curriculum of such content areas as English, mathematics, and science (CSTA, 2016). Similar work was conducted by International Society for Technology in Education (ISTE, 2016) which incorporated CT into the current ISTE standards. Also, the Next Generation Science Standards (NGSS) added CT to the list of key engineering and scientific practices for K-12 students.

The research community also supports the idea of integrating CT into the core subjects because even if schools mandate computing courses, there are not enough CS graduates to teach programming in schools. As Guzdial (2017) underlined in his blog post, “…in 2016, only 75 teachers graduated from universities equipped to teach computer science.” As such, there is a growing need to research how to expand CT into other disciplines. For example, in their influential article, Weintrop et al. (2016) proposed a framework that educators can use to teach CT effectively in the core subject classes and argued that students would have a chance to work with real-world problems and applications in that kind of instruction. Recently, the efforts of the research and education community in this area have resulted in broad changes. For example, the state of Virginia went through a revision in its Standards of Learning for English, Mathematics, Science, and History and Social Science and mandated the reinforcement of the computational thinking in these subjects.

In addition to incorporating CT into core subjects, other strategies have evolved to teach and learn CT, such as the use of CS Unplugged projects (Rodrigue, Kennicutt, Rader, & Camp, 2017). CS Unplugged projects are defined as the collections of activities developed to improve CT without using computers, including games, puzzles, and movements. The aim is to expose young
learners to challenges and questions that computer scientists explore in everyday life (Bell Alexander, Freeman, & Grimley, 2008; Rodriguez et al., 2017). This strategy alleviates the concerns of equating CT with coding (Kafai et al., 2014).

**Development of Computational Thinking in the K-12 Programming Context**

The history of programming for K-12 education goes back to the 1970s. Seymour Papert (1980) pioneered the field, arguing that programming gives students the opportunity to learn about their own thinking. After Papert’s influential Logo programming study, low-floor (i.e., easy to program), high-ceiling (i.e., powerful to create complex programs) constructionist tools have started to emerge (Grover & Pea, 2013). Through such tools, learners were able to see the effects of their own programming. With technological improvements, low-floor, high-ceiling tools such as visual, block-based programs are widely available today. Previous research on the use of these tools has shown that programming is an effective context to reinforce computational thinking in K-12 students (Kafai et al., 2014; Margolis, Goode, & Bernier, 2011; Resnick et al., 2009).

Of particular relevance to this research is whether learner characteristics have significant relation with middle-school students’ CT in programming context. In a recent study, Kafai et al., (2014) examined the use of electronic textiles (e-textiles) for introducing key computational thinking concepts and practices while broadening perceptions about computing in high school computer science classroom through the LilyPad Arduino. Kafai et al. analyzed the students’ artifacts, program code, and their design approaches in their e-textile artifacts and computing perspectives. The authors found through students’ design that they “expanded their thinking about the relevance of computing to their personal lives, their self-concepts as computer scientists, and their understanding of computing as a field” (p. 17). In another study, Denner, Werner, and Ortiz (2012) analyzed 108 games created by middle-school girls using Stagecast Creator in an after-school class. Their results have shown that game programming is a promising strategy to learn computational thinking concepts. Similarly, Grover (2014) designed and implemented Foundations
for Advancing Computational Thinking (FACT), a six-week online, introductory computer science (CS) for middle-school students using Scratch, visual block-based programming tool. As a result of the study, students gained an understanding of CT concepts such as sequence and loops. Besides this, Grover, Pea, and Cooper (2016) conducted a multilevel analysis with the same data to understand the relationship between learner characteristics (e.g., interest and prior experience) and student performance on CT assessments. They found that prior computing experiences and mathematics and English abilities were predictors of learning gains. In addition, Grover and Pea (2013) conducted another study to teach two groups of middle-school students programming using App Inventor. App Inventor is a visual, block-based programming tool which uses a graphical programming environment. It enables students to create applications (apps) for Android devices. Part of the rationale for choosing this learning tool was that App Inventor was found to be a “gender neutral and truly democratic” (Grover & Pea, 2013, p. 726) tool for teaching computing.

In the current study, I implemented an App Inventor curriculum to extend the effort of previous research with a broader range of variables to better understand which learning characteristics have a significant relationship with students’ CT in a middle-school programming environment. In order to do that, and more broadly to make CT a part of K-12 education, there needs to be appropriate assessments (Grover & Pea, 2013), because the accuracy of the assessment of students’ CT depend on how reliable the instruments are. In the next section, I synthesize the CT assessments found in the literature.

**Computational Thinking Assessments**

My review of the literature shows that there has been limited research on the evaluation of CT for K-12 students (Basawapatna et al. 2011; Grover & Pea, 2013; Maiorana, Giordano, & Morelli, 2015; Roman-Gonzalez, 2015; Werner, Denner, & Campe, 2012). This might be due to the methodological challenges associated with the study of CT instrument development and validation,
such as the lack of consensus on the operational definition of CT and the variety of CS domain-specific needs.

Most CT studies have used artifact analysis as an evaluation strategy. One of the prominent studies addressing the need for CT assessment is known as the Fairy Assessment through Alice, which is a visual block-based programming tool to create animations and games while students apply CT concepts and practices. In their study, Werner, Denner, and Campe (2012) measured CT performance among middle-school students. Their results showed that the Fairy Assessment is a promising strategy for assessing CT among middle-school-aged students in “picking up a range of CT across students, and a variety of types of CT across the three tasks” (p. 4) including algorithmic thinking, abstraction, and modeling. In another study, Basawapatna et al. (2011) designed a CT patterns quiz to test participants’ ability in recognizing and understanding patterns in a given problem. They found students were successful in finding similar patterns between the problem and the previous programming assignment. In addition to these, there are a few online quiz platforms for tracking progress, such as Quizly and Quizmaker, where grading is done automatically through App Inventor (e.g., Maiorana et al., 2015). However, these assessments are limited in terms of providing information on students’ development of CT skills over time or as the result of an intervention because they can be done only as a posttest assessment.

After an extensive literature review, I found two studies that detailed instrument creation and validation steps—that is, to establish content and to construct criterion, convergent, and divergent validities (Thorndike & Hagen 1961), for assessing K-12 students’ CT—that can address the above-mentioned limitation. Roman-Gonzalez et al. (2015) created and validated a CT multiple-choice quiz based on Scratch visual block-based programming. They administered their instrument to about 1,250 students and provided criterion validity of their instrument. However, Roman-Gonzalez et al. (2015) did not provide evidence for their construct validity. Chen et al. (2017) designed an instrument to assess fifth-grade students’ CT and established content and construct
validities. Chen et al.’s validation process was based on the definition of CT adapted from CSTA’s standard; they went through Rasch modeling to analyze the items in their test and found that their instrument had promising results when it came to revealing growth in participants’ CT ability. However, Chen et al. did not test the criterion validity, and acknowledged that robotics programming was emphasized in the instrument; this could limit their findings’ generalizability.

As Chen et al. (2017) stated, each programming context needs its own assessment for CT to ensure content validity. For example, if the instrument is designed based on visual block-based programming instruction such as robotics, its validity is questionable if it is used as an assessment for Java or Python programming languages. Since this research involved app-building activities, there was a need to conduct a CT assessment in App Inventor context. To address this need, I developed and validated an instrument for middle-school students based on the CT constructs in accordance with Brennan and Resnick’s (2012) framework.

**Previous Studies Regarding Learner Characteristics in Computer Science Education**

A literature review showed that previous studies in computer science education (CSEd) examined the learner characteristics that are related to CS learning in college-level courses. Next, I explain more fully these learner characteristics that I used in this study for investigating their relationships with middle-school students’ CT, and then I conclude this section by explaining the variables that were controlled for in this research.

**Self-efficacy** refers to one’s beliefs about the ability to succeed in a situation or a task (Bandura, 1997, p. 36) and it takes its origins from Bandura’s social-cognitive theory. Bandura states that students’ self-efficacy beliefs are the strongest determinant of their behavior change because they influence their initial decision on how to behave in a situation and persistence in overcoming difficulties; students with high self-efficacy do their best to achieve their goals. Bandura also argued that past experiences have an impact on one’s self-efficacy beliefs which in turn may affect one’s academic performance. Self-efficacy has since been accepted as the most
significant motivational factor that shows the relationship between prior experiences and future performance (e.g., Britner & Pajares, 2006; Pajares & Miller, 1994).

In agreement with Bandura’s theory, the literature showed a connection between students’ self-efficacy and STEM-related choices (Fouad, 2007; Moakler & Kim, 2014). This relationship starts in early education and continues throughout the years. Evidence from previous research showed that there is a strong positive correlation between students’ self-efficacy and their performance in college-level computer science courses (Lishinski, et al., 2016; Wiedenbeck, 2005). In a recent study with college students, Lishinski et al. (2016) investigated the relationships between students’ gender, goal, self-efficacy, and performance. They found through structural equation modeling (SEM) that the best indicator for performance was self-efficacy, including metacognitive strategies and goal orientation. Wiedenbeck (2005) also employed SEM via path analysis to study the effect of self-efficacy on non-major students’ CS performance and found that it was a significant predictor of student success.

Another learner characteristic included in this study is students’ initial interest in the subject. Interest is “a psychological state of having an affective reaction to and focused attention for particular content and/or the relatively enduring predisposition to re-engage particular classes of objects, events, or ideas” (Renniger & Hidi, 2002, p. 174). In the literature, interest was categorized into two groups: situational and individual interest (Krapp & Fink, 1992; Krapp, 1999; Schiefele, 1991). Individual interest refers to a person's interest with specific content over time (Renniger & Hidi, 2002). Situational interest arises in the moment and triggered by whatever happening in that moment. Situational interest is important to attract students’ attention to the activity, but individual interest is crucial to holding that attention (Renniger & Hidi, 2002). In situational interest research studies, researchers found that meaningful instruction, student involvement, (Mitchell, 1993) and cognitive demand (Chen & Darst, 2001) were important factors that generate that kind of interest. Besides this, individual interest studies showed positive correlations between students’ individual
interest and their learning. For example, Beyer (2014) investigated the relationship between students’ grades in a computer science class and predictors such as prior experiences, interests, and self-efficacy, and discovered that students with interest in computer science were more likely to stay in the course and have higher grades. Furthermore, Haungs et al. (2012) developed a college-level curriculum that provided some flexibility for students. They created different instructional tracks (e.g., robotics, gaming, music, mobile app) so students were able to choose their own learning path according to their personal interest. Haungs et al. argued that their intervention increased students’ academic performance and retention in a CS context.

**Goal orientation**, another learning characteristic that may influence quality of CSEd refers to students’ expectations while performing a task. Goal orientation affects student behavior according to the types of outcomes that students desire, which in turn affect their performance. The basic idea of this theory is summarized in two categories: performance and mastery orientation. The former refers to individuals who want to demonstrate their competence to others and the latter refers to individuals that develop their competence for the sake of learning (Elliot & Dweck, 1988). Elliot and McGregor (2001) further developed the two orientations by subdividing each one into approach and avoidance orientations, leading to four types of goal orientations: *performance approach*, *performance avoidance*, *mastery approach*, and *mastery avoidance*. Previous research has found a positive relationship between computer science learning and mastery goal orientation (Bergin, Mooney, Ghent, & Quille, 2015; Zingaro & Porter, 2016), implying that students’ goal orientation may have an impact on their development of CT.

Another important factor in CSEd performance may be students’ *prior experience* in the content area. With the technological improvements students come to the classrooms with broad range of prior knowledge, skills, and experiences (Svinicki, 2004). Svinicki argued that learning is related to what students know about the content. The content in programming course entails the experience with the selected programming language or tool. However, the relevant research is far
from conclusive in the CS literature. In a recent study, Grover, Pea, and Cooper (2016) analyzed the factors affecting computer science learning individually and found that prior experience predicts higher grades in an online programming course for K-12 students. In addition, Beyer (2014) found that students who had positive prior experiences were more likely to remain in CS. To the contrary, however, Watson, Li, and Godwin (2014) argued in their study that prior experience had little impact on CS performance.

**Control Variables in this Study**

**Gender.**

Gender differences in CSEd is well-documented in the literature as a social construct. Although many international school systems are now requiring CS as part of their core curriculum (Balanskat & Engelhardt, 2015), studies indicate that a gender gap still exists in computational fields (Margolis, Estrella, Goode, Holme, & Nao, 2008; National Science Foundation, 2017). For example, girls are found in general to be less confident in their use of computers, and boys “have significantly more positive attitudes toward computers than girls, finding computers more ‘enjoyable’, and are more likely to get involved with coding and computer programming” (Mark & Henson, 1992, p.1; Robertson, 2012). Similarly, in 2015, only 21.9% of the all Advanced Placement (AP) in Computer Science exam takers were female. However, compared with AP exam-taking male students, AP-taking female students are reported to be ten times more likely to major in computer science (Change the Equation, 2016). With such differences reported in the literature, it is worth looking at the differences among female and male students in terms of their CT in a computing course.

**Age.**

Since CT in K-12 settings is still in its infancy (Yadav et al., 2011), few studies examine the relationship between CT and age or grade level. For example, Atmatzidou and Demetriadis (2016) found that CT skill level does not differ according to age. However, they also stated that girls needed more training time to reach the same CT level as boys. In contrast to this, in a study that
took place in Turkey with high school students, Durak and Saritepeci (2018) found that student grade level had a significant, positive effect on CT, and students’ CT increased as students became older in their cross-sectional study. In addition to these studies, age/grade level was found to be positively correlated with skills that are highly correlated with CT such as reasoning (Mills et al., 1993; Román-González, Pérez-González, Moreno-León, & Robles, 2016) and problem solving (Blanchard-Fields, Jahnke, & Camp, 1995). Despite these studies, research is inconclusive about which CT concepts and practices should be introduced to what age groups of students (Lee et al., 2011). In addition, it is reported in the literature that there is a scarcity of research studies with regard to CT assessment for young students (Barr & Stephenson, 2011). Given that the literature has contradictory results, therefore, it is important to include age as a variable in the analysis.
CHAPTER 3

METHODOLOGY

Overview of the Research

This research is the part of a larger project called Acquainting Metro Atlanta Youth with STEM (AMAYS). It is a CSEd intervention that is designed to increase access to computing experiences for underrepresented youth by bringing CS learning to where the students already are, in a subsidized after-school program embedded at their public schools. AMAYS is a design-based research and development project, which began in the summer of 2015 and is scheduled to end in summer 2018. In this current research, two studies took place. Study 1 was conducted at a STEM-based, private middle school located in northern Atlanta in Spring 2017 and Study 2 took place in an informal after-school learning context in the metro Atlanta area in Fall 2017. The curriculum, pedagogy, and procedures were kept similar in the two studies. Both groups participated in the similar app-building activities designed to teach programming while improving their CT skills. My purpose with this research was to understand which learner characteristics were related to CT for middle-school-aged students and to examine the generalizability of my findings. Data was collected in the form of a learner profile survey, a CT quiz, and digital artifacts. As mentioned in the introduction section above, the overarching research question in the current research is: What is the nature of the relationships (if any) between students’ learner characteristics and their demonstrated CT concepts and practices?

Specific research hypotheses related to this research question are:

H1. There will be a correlation between students’ self-efficacy level and CT quiz score in the formal computer class.

H2. There will be a correlation between students’ self-efficacy level and CT practice in the formal computer class.
H3. There will be a correlation between students’ self-efficacy level and CT quiz score in the after-school environment.

H4. There will be a correlation between students’ self-efficacy level and CT practice in the after-school environment.

H5. There will be a correlation between students’ interest level and CT quiz score in the formal computer class.

H6. There will be a correlation between students’ interest level and CT practice in the formal computer class.

H7. There will be a correlation between students’ interest level and CT quiz score in the after-school environment.

H8. There will be a correlation between students’ interest level and CT practice in the after-school environment.

H9. There will be a correlation between students’ goal orientation and CT quiz score in the formal computer class.

H10. There will be a correlation between students’ goal orientation and CT practice in the formal computer class.

H11. There will be a correlation between students’ goal orientation and CT quiz score in the after-school environment.

H12. There will be a correlation between students’ goal orientation and CT practice in the after-school environment.

H13. There will be a correlation between students’ gender and CT quiz score in the formal computer class.

H14. There will be a correlation between students’ gender and CT practice in the formal computer class.
H15. There will be a correlation between students’ gender and CT quiz score in the after-school environment.

H16. There will be a correlation between students’ gender and CT practice in the after-school environment.

H17. There will be a correlation between students’ grade level and CT quiz score in the formal computer class.

H18. There will be a correlation between students’ grade level and CT practice in the formal computer class.

H19. There will be a correlation between students’ grade level and CT quiz score in the after-school environment.

H20. There will be a correlation between students’ grade level and CT practice in the after-school environment.

H21. There will be a correlation between students’ prior experience with creative computing technologies and CT quiz score in the formal computer class.

H22. There will be a correlation between students’ prior experience with creative computing technologies and CT practice in the formal computer class.

H23. There will be a correlation between students’ prior experience with creative computing technologies and CT quiz score in the after-school environment.

H24. There will be a correlation between students’ prior experience with creative computing technologies and CT practice in the after-school environment.

Design

In this current research, I used a nonexperimental design. The primary reason for choosing this design is that the research questions of this research are broad and exploratory. More specifically, this research was intended to be a preliminary investigation of the relationships between the learner characteristics and CT. In addition, I did not manipulate the variables included in this research, and
it was not feasible to do random assignments due to the naturalistic nature of the study sites. This is because Study 1 site was a formal classroom which did not allow for random assignment and activities in Study 2 site took place in an informal after-school setting where one classroom was designated for students who signed up for and self-selected themselves into the after school program called AMAYS.

The two studies were kept as separate studies because the environment and student profile were substantially different from each other: as mentioned earlier, Study 1 took place in a formal computer class in a STEM-based, private middle school whereas Study 2 took place in an informal after-school learning setting within a low-income neighborhood in the metro Atlanta area. Besides this, students in Study 1 had diverse ethnic backgrounds, but Study 2 participants consisted of only African-American students. I did studies in those different sites to understand which learner characteristics were related to CT for middle-school-aged students and to examine the generalizability of my findings. The detailed description of the study sites and participants are described in following section.

**Study 1**

**Site and Participants.**

In total, 166 middle-school students from a private, STEM-focused school located in a large Southeastern city participated in the research. Female students accounted for 56.8% of participants and males for 29.7%; and 13.5% of the students did not provide their gender information in the survey. Class size ranged from 18 to 21 students, and seven classes participated in this study under the supervision of 2 teachers. Each student in the school was required to attend to a computer class within his or her grade.

**Lessons, Procedure, and Data Collection.**

The intervention included activities in a total of eight hours of class time over a single semester. The intervention took place one day a week for 50 minutes over a period of seven
weeks in a computing class. During the first week of the course, students were given a course orientation during which they took the adapted version of Barron, Walter, Martin, & Schatz’s (2010) survey. From weeks 2 to 5 of the intervention, students completed self-paced activities designed to help them learn the features of App Inventor. During the sixth and seventh week, students completed a performance-based assessment in which they were asked to customize an app that gives health recommendations to a mobile app user according to the user’s selection. The last week was designated to be a posttest and a pizza party. At the end of the intervention, the teacher shared students’ digital artifacts with the researchers for performance-based evaluation.

Whenever possible, each activity was designed to connect the app building with relevant, and socially responsible themes. For example, I created the favorite artist app with what is relevant to today’s youth pop culture in mind. In this activity, students learned some of the basics of app programming by creating a soundboard app that plays audio (sound effects, music, etc.) when users press buttons. Figure 1 shows a screenshot from the Favorite Artist app.

Figure 1. Favorite Artist App
More specifically, activities—with gradually increased difficulty, and gradually decreased amounts of scaffolding (Kirschner et al., 2006)—were distributed to the students. The first set of activities involved students developing pre-designed apps using App Inventor by following step-by-step instructions (with worked examples). These activities intentionally exposed participants to computational concepts and practices such as conditionals, testing, and debugging (Brennan & Resnick, 2012). For example, the calculator worked example/worked example aimed to teach conditionals. Figure 2 demonstrates the calculator worked example.

![Figure 2. Calculator worked example/worked example.](image)

**App:** Calculator  
**Level:** Intermediate  

**What students learned:**

- How to add a horizontal arrangement.  
- How to use the listpicker function to create a menu of options.  
- How to use “if” statements. How to display the results of a calculation.

Upon the completion of these initial activities, students were given a problem-based activity with fewer direct instructions (Guzdial, 2009). These problem-based activities, which I called cPBL in short, were similar in design to the performance-based assessment that students would ultimately be required to complete, in that they were presented a problem and asked to design a solution using App Inventor. Students were also given a chance to pick their own app design and/or development problem to solve. For example, health advice app was one of the cPBL activities, which was the extension of the calculator app. Using what students had learned
from the calculator app, they tried to create an app that can give health advice to people in need.

Figure 3 illustrates the app I developed that was delivered to students for illustration purposes to help with giving them an initial idea about the app they were expected to develop in their own projects. More information about the health app can be found in the Appendix B.

![Sample app I developed](image)

**Figure 3.** Sample app I developed for illustration purposes.

Unlike the App Inventor worked example activities, the problem-based activities gave students the flexibility of choosing to incorporate their home culture and daily life into their computing learning experience during the class time, if they so choose (Kafai et al., 2014; Lachney, 2018). Most of the students worked in self-organized teams of two or three (Kafai et al., 2014) to design their mobile apps, and some of the students worked individually.
Study 2

**Site and Participants.**

Participants were largely African American middle-school students on free and reduced-price lunch who were participating in a free after-school program, which operates at multiple middle-school sites in metropolitan Atlanta, Georgia. In an effort to foster CT skills through the use of visual, block-based programming for middle-school students, AMAYS was integrated into already existing after school program. In the existing after-school program, students were able to select programs to participate in from a list of available programs. The program information was distributed to the families on a flyer among other channels. Students were free to choose activities according to their interest for the semester.

**Intervention Lessons, Procedure and Data Collection.**

AMAYS met for up to 90 minutes twice a week with up to 10 participants per site at nine school sites. In this study, students worked on the revised versions of the Study 1’s instructional materials, while keeping the underlying CT concepts and practices consistent with Study 1. The main difference between Study 1 and Study 2 in terms of the curriculum was the order of the activities. Specifically, students in Study 1 completed five worked examples in order. After completing these activities, they then worked on cPBL activities, whereas Study 2 curriculum required students to complete a worked example, followed by a guided practice activity. For each site in Study 2, after five days, students were given five different cPBL activities to choose from, like in Study 1, and were expected to create their own app to solve the problem.

Pretest data collection took place during the first week of AMAYS when 87 students completed a learner profile survey. Female students accounted for 30% of participants; males, 48%; and 6% of the students declined to provide their gender information in the survey, and 16% of the participant gender information was missing. Class size ranged from 4 to 10 students at nine schools under the supervision of nine teachers and mentors, and five graduate research assistants. After two-and-a-
half months of AMAYS intervention, students took the slightly revised version of the CT-App quiz. That version of the CT-App had one more additional question, 12 questions in total, and two questions for each concept. Furthermore, students’ digital artifacts were collected through the online system throughout the intervention.

**Test Development**

Since there is still not an agreement on the CT definition and there are not any CT tests available for App Inventor, I developed a CT instrument, an App Inventor-specific quiz, and went through its validation process for this study. In the following section, I summarize the instrument’s development and how I assured its content, construct, and criterion validity.

**CT Instrument: CT-App.**

The CT Instrument I developed, which I call CT-App, initially consisted of eighteen questions for measuring the students’ CT through the CT concepts and practices and had 12 questions in the latest version. In this section, I explain the details regarding CT-App by organizing them around the content, construct, and criterion validity.

**Content Validity.**

To assess the content validity of an instrument—or to construct a test to measure a particular set of processes and content areas—requires a researcher to specify the processes and content areas to be measured explicitly. The content validity of a test is therefore established by the correspondence between the statement of what a test is intended to measure and the definition of the trait to be measured (Thorndike & Hagen, 1961).

To establish the content validity, several steps were taken. First, I adopted Yadav et al.’s (2017) definition of CT as stated in Chapter 2. Then, I determined the practices and content areas of CT by using Brennan and Resnick’s (2012) CT framework. I matched the CT definition with the framework to ensure that the CT instrument included the appropriate content referring to the definition and required the application of the appropriate cognitive practices of CT.
Once I had decided on the definition and the framework, I created eighteen multiple-choice items by using App Inventor. Our questions assessed students’ computational concepts and practices in a blended format. I developed the questions by using computational practices, and in order to answer the question correctly, students needed to refer to computational concepts. For example, I developed a troubleshooting question (see Figure 4) based on the CT practices outlined in the Brennan and Resnick framework. Students needed to demonstrate their understanding of CT concepts such as “if-else” conditionals to answer this question correctly. Another sample question (see Figure 5) was designed around CT practices called iteration. In sample question 2, students should have known that the last color would delete the previous color but to answer that correctly they also had to know what CT concepts “when/do block” refers to in App Inventor. In addition, I included questions designed for assessing most of the CT concepts stated in the Brennan and Resnick CT framework. Figures 4, 5, and 6 present sample questions from CT-App instrument. The full version of the test can be found in the Appendix A.
Figure 4. Sample Question 1: Operations from CT concepts, Troubleshooting/Debugging from CT Practices.
Figure 5. Sample Question 2: Sequence from CT concepts, Being Iterative from CT Practices.
For developing the face validity procedure of the test, I collaborated with researchers from computer science, educational technology, and educational psychology. Additionally, two graduate assistants worked on the syntax and semantics of the questions. After that, I conducted a focus group interview and a think-aloud procedure with the sample from our target group. The measure
items were refined by following feedback from a middle-school focus group to eliminate difficult or App Inventor context-specific language (Roman-Gonzalez, 2015).

**Construct Validity.**

Construct validity refers to how well a test or an experiment measures its claims and considers whether the operational definition of a variable reflects the true theoretical meaning of a concept. After solidifying the content validity of our items, I administered an online version of the test to 45 students from grades 6 through 8. I conducted a confirmatory factor analysis to establish the construct validity among the chosen questions (Bogazzi, Yi, & Philips, 1991). I developed one construct model to show that all of the questions were assessing one construct, in this case, CT.

**Criterion Validity.**

Criterion-related validity refers to the case where the results from an instrument accurately relate to/predict some external variable. There are two broad classes of criterion-validity: depending on when the test information or the criterion information is collected. If the test information is to be used to forecast future criterion performance, then it is called predictive validity. On the other hand, sometimes researchers would like to know if scores on one test correlate highly with the scores obtained concurrently on another assessment that assesses the same construct; this is called concurrent validity of the instrument. To examine the criterion validity, I analyzed the concurrent validity of the instrument. The participants in this study developed a digital artifact to address each given problem. Those artifacts were graded through a validated rubric (Sherman & Martin, 2015) as performance-based assessment, which is detailed in the next section. I conducted correlation analysis between our criterion test—defined as the scores obtained through this performance-based assessment—and CT-app score. In this way, I was able to confirm whether students scored similarly on a different test measuring the CT construct.

The results of the reliability and validity of the test in the Study 1 context can be found in the next section.
The Measures

Learner Profile Survey.

I administered an adapted version of Barron et al.’s (2010) survey to find out each student’s learner characteristics. I adopted three of the constructs from that survey: prior experience with creative computing technologies and motivational aspects of learning about technology, including interest and self-efficacy. In addition, I added items that ask about the student’s gender, grade level, and initial purpose in the program.

Interest. Students’ interest in learning about computing technology was measured based on the scheme presented in Barron et al.’s (2010) survey. The Cronbach’s alpha for the items in their study was 0.83, establishing and indicating sufficient reliability of the items. In the current research, I asked students the three interest-specific five-point Likert-scale questions listed in Barron et al. (2010) and one additional question regarding their interest in App Inventor. The additional question measured student interest in the program, with choices including "I cannot wait to get started!" and "I am not very interested," and “I do not want to do it." The Cronbach’s alpha was found to be .853 in Study 1, meaning that survey items that measured the students’ interest had very good internal consistency (Hair, Anderson, Tatham, & Black, 1995).

Prior Experience with Creative Computing Technologies. Students’ prior experience with thirteen creative production activities were derived from student responses to a set of questions from Barron et al.’s (2010) survey. Barron and her research team picked activities that involved some aspects of design, personal expression and/or required more advanced concepts related to computing. The Cronbach’s alpha was .862 in Study 1.

Students answered a series of questions about the extent of their prior experience with creative production. Questions about experience (e.g., “How often have you coded a website using HTML?”) were presented in five-point Likert scale, for example, an item with five options: “I have
never done this”, “I’ve done this once in my life”, “I’ve done this a few times”, “I’ve done this a lot”, and “I don’t know what this is”.

As described in Barron et al. (2010), students’ breadth and depth of experience were calculated to create the four types of student profiles of experience: beginner, specialist, explorer, and generalist. Thus, this is a categorical variable with “beginner” coded as 1, “specialist” as 2, “explorer” as 3, and “generalist” as 4.

**Depth of Prior Experience with Creative Computing Technologies.** Students’ depth of prior experience predictor was calculated according to Barron et al.’s (2010) description. For each of the thirteen creative activities, students indicated how many times they participated in each activity. For each student, the activities that were scored five or more were coded as 1 and the others were coded as 0, similar to the approach taken in Barron et al.’s study. Then, a total score was calculated for each student by summing over the 1/0 outcomes student achieved on each activity, leading to a score between 0 and 13. This overall score served as a measure to indicate the depth of students experience with creative computing technologies. The median split method was employed to determine the students with a high and low depth of prior experience. Thus, this is a binary variable, “low depth” was coded as 1 and “high depth” was coded as 2.

**Breadth of Prior Experience with Creative Computing Technologies.** The breadth of the prior experience is calculated similarly to the way Barron et al. (2010) calculated in their study. I used SPSS 22 (IBM Corp., 2013) to compute this variable by summing up the number of activities each student had participated in at least once. The range could be from 1 to 13. The median split method was used to determine the high breadth vs. low breadth groups. Thus, this is a binary variable, “low breadth” was coded as 1 and “high breadth” was coded as 2.

**Self-efficacy.** Students’ self-efficacy was measured by way of their response to one self-efficacy-specific question. The item for self-efficacy was as follows: “I feel confident in my ability to
learn how to build phone apps.” The response options were presented in a five-point Likert scale from 1 to 5, indicating the range from “strongly disagree” to “strongly agree”.

**Gender.** Student gender information was collected through a survey question. Female students were coded as 1, and male students were coded as 0 in the dataset.

**Purpose.** Participants were asked the following question to assess their goal orientation in the intervention: “Select each of the following that describes your goal in learning App Inventor program” and were given the following response options: “I want to be better than everyone else in the group” (called GoalBetter in short for this research), “I do not want to fail” (GoalFail), “I want to understand how to do this stuff” (GoalUnderstand) and “I want to have fun” (GoalFun). These questions were adapted, except the last one, from Elliot and McGregor’s (2001) achievement goal theory.

In Study 1, this construct was presented in a multiple-choice item so that students were able to pick only one goal orientation. Each subgoal variable was dummy coded. However, in Study 2, this measure was presented in a multiple-selection format, and students were able to pick more than one goal orientation. This time binary variables were coded as 1 if that goal orientation was included in student’ selection and 0 if otherwise.

**Grade Level.** Grade level was a categorical variable and coded as 6th, 7th, or 8th grade.

To evaluate the participants’ computational thinking concepts and practices, I used two data sources: a) Computational Concepts and Practices Quiz score, and b) CT Practices.

**CT-App Quiz.**

The first version of the CT-App instrument which was used in Study 1 showed promising psychometric properties based on the posttest results with regard to its reliability and validity. The reliability analysis was conducted by SPSS (IBM Corp., 2013) because SPSS has a feature that shows the reliability of the questions left when one of the items is deleted. After the item-level reliability
analysis, seven items were omitted, and therefore only eleven items remained in the instrument. The Cronbach’s alpha for the eleven items in the quiz was .80, indicating good reliability (Cronbach, 1951; George & Mallery, 2003). In addition, I conducted confirmatory factor analysis with the selected items and then evaluated the model fit. The chi-square test of model fit for the theorized model was not significant ($\chi^2_{44} = 49.45, p = .26$), indicating that the model fit was good. I also considered two additional model fit indices based on Hu and Bentler’s (1999) comparative fit index (CFI) and Steiger and Lind’s (1980) root-mean-square error of approximation (RMSEA). The proposed model had a CFI value of 0.91, a standardised root mean square residual (SRMR) value of 0.91, and an RMSEA value of 0.06, further strengthening good fitness of the model. A one-model factor was selected because the questions were designed in a blended format that students had to know CT concepts while applying them in their practices.

In addition to the construct validity, I conducted criterion validity analysis between the CT-App and cPBL performance score in Study 1. The scores obtained through the performance-based assessment provided criterion validity for the CT instrument. According to the correlation analysis results, I was able to confirm whether the students performed similarly on a different test that measures the same construct. I found out that the scores on the performance test, which measured their CT through artifact analysis, were significantly related to their scores on the multiple-choice CT-App ($r = .331, p = .036$). Normality of the model residuals was verified.

In addition, the content validity of the quiz was established with the expert judgment analysis. Experts and I agreed that all of the questions refer to one of the computational concepts and practices stated in the Brennan and Resnick’s (2012) framework. In addition, we had a consensus on having twelve questions or less to have the length of the test suitable for my research contexts.

Computational Concepts and Practices Quiz Score was calculated with 11 questions that showed good reliability for each participant in Study 1 who took the first version of the CT-App.
test at the posttest. A slightly revised version (second version) of the CT-App instrument was used to assess CT for Study 2 participants. For assessment purposes, first the values were recoded. Specifically, the correct answer to each question was recoded as 1, and incorrect choices were recoded as 0. The total number of the correct answers constituted each of the students’ final score. The results showed that two of the questions were missed by more than 85% of the students, therefore, the two questions were omitted from the calculation, leaving us with 10 questions in total in the CT-App test for Study 2 participants. Table 2 shows a frequency table including each question, which highlights the excluded questions in bold.

Table 2

<table>
<thead>
<tr>
<th>Question</th>
<th>% of Students Answered Incorrectly</th>
<th>% of Students Answered Correctly</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>68.7</td>
<td>31.3</td>
</tr>
<tr>
<td>2</td>
<td>80.6</td>
<td>19.4</td>
</tr>
<tr>
<td>3</td>
<td>77.6</td>
<td>22.4</td>
</tr>
<tr>
<td>4</td>
<td>73.1</td>
<td>26.9</td>
</tr>
<tr>
<td>5</td>
<td>77.6</td>
<td>22.4</td>
</tr>
<tr>
<td>6</td>
<td>82.1</td>
<td>17.9</td>
</tr>
<tr>
<td>7</td>
<td>86.6</td>
<td>13.4</td>
</tr>
<tr>
<td>8</td>
<td>67.2</td>
<td>32.8</td>
</tr>
<tr>
<td>9</td>
<td>56.7</td>
<td>43.3</td>
</tr>
<tr>
<td>10</td>
<td>73.1</td>
<td>26.9</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>74.6</td>
<td>25.4</td>
</tr>
</tbody>
</table>

**CT Practices Analysis.**

Students’ digital artifacts were categorized according to the presence of any CT practices based on the Brennan and Resnick’s (2012) framework. I analyzed 95 apps created in Study 1 and Study 2, and created a binary variable called CT Practices. Students who demonstrated CT practices in their app were coded as 1, and others were coded as 0. In addition, students who attended Study 1 or Study 2 but did not submit an artifact were also coded as 0. Next, I explain in more detail what constituted CT practices that were found in students’ apps and coded as 1 for the CT practices variable.
One of the general CT practices found in students’ artifacts was modifying or remixing the codes from previous projects created in the class. Furthermore, in some of the projects, computational thinkers demonstrated that they not only applied computational concepts and practices but also utilized an interdisciplinary approach by incorporating science or mathematics concepts into their apps. Figure 7 illustrated sample projects developed by students through remixing several worked examples code developed in the previous sessions of the course. For example, Endangered Animals app which was shown on the left of Figure 7 was developed through combining the Favorite Artist app code block (Figure 1) with Calculator worked example (Figure 2). Favorite Artist worked example was an activity to show how to add a picture and a sound file to the app while linking them together with the button-clicked code block. On the other hand, the Calculator worked example was used to teach how to add a list to an app and do operations according to the user selections. In Endangered Animals app, which was shown at Figure 7, participants successfully used their previous knowledge, remixed the code and created an app that enables the user to pick an animal and get information regarding that animal. Another example is body mass index calculator shown on the right in Figure 7. Students built on the calculator app, add more complexity, and created their body mass index calculator app.
Another set of general CT concepts and practices that appeared in the student artifacts was the reusing of the existing code and clearing away of the unnecessary code block existing in the worked examples. Figures 8 and 9 illustrate how participants shortened the existing code block and demonstrated CT by not copying the existing code exactly but remixing it in a more efficient way (see Figure 9). For example, code blocks “A” and “B” in Figure 8 were combined, the “Listpicker.Selection” code block was dropped out, and code block “C” in Figure 9 was created.
The last general CT practices found in the artifacts were building apps through a new code block that was not covered in the curriculum, and these code blocks were more advanced ones, such as multiple screens and clock timer. The participant’s sample project in Figure 10 used the
multiple screen feature of App Inventor. In Screen 1, the user was asked whether or not they liked Pokémon and Batman. According to the button pressed on the screen, “Yes” or “No”, the app opened another screen with the addition of a thumbs-up or -down image to Screen 1.

![Sample Project for Using New Code Blocks](image)

Figures 10. Sample Project for Using New Code Blocks

On the other hand, general trends among students who did not demonstrate CT in their artifacts—and who were referred to as 0 in the CT practices variable—were building their apps through changing the design of the worked example but not the code, adding the same code block used in the worked example for different design elements, building the app through design elements without any coding, and submitting one of the worked examples without changing anything. In a nutshell, their project did not reflect any changes happened in the original worked examples during Study 1 and Study 2 pertinent to CT concepts and practices. Figure 11 illustrates a project that included Favorite Artist worked example’s code with the change of design elements, Figure 12 demonstrates a project sample that student changed slightly by adding similar code block for additional design elements, and Figure 13 shows a student project without any coding but with design elements.
Fidelity of Implementation

Study 1.

I took several steps to ensure the fidelity of the implementation. First of all, I conducted two hours of teacher training, including introduction of the curriculum, intervention, and data collection procedures. Since teachers already had experience with App Inventor, there was no need to explain the concepts in the curriculum.

Secondly, I attended the first session of the intervention to make sure all of the students understood what the implementation was about and their rights to be involved. After that, I held weekly, online meetings with teachers to check if the intervention was going as it was planned and if there was any modification needed for the next sessions. Lastly, I was available online while classes were taking place in case teachers had any questions for me. Teachers asked for quick meeting for three times while I was waiting online.
Study 2.

Similar steps were taken to ensure the fidelity of implementation during AMAYS intervention. Similar teacher training took place but there was more focus on App Inventor than on the curriculum because the teachers were not familiar with App Inventor. The research group and I made sure that teachers, graduate students, and mentors were all on the same page in terms of the classroom pedagogy. Graduate students and mentors attended one or two sessions weekly in each school to ensure the intervention went as planned. Several concerns were raised during the intervention such as unstable attendance, teachers’ motivation throughout the intervention, unexpected class cancellations which were expected due to the informal, afterschool school site. Mentors and graduate research assistants presented in each class when the classes were available to ensure the fidelity of the implementation.

Access, Role, and Ethics

I gained access to the classrooms used in this research through the following steps. I sent an e-mail to the principal of the private school and subsequent face-to-face meetings with him resulted in initial approval to conduct the research in that school for Study 1. For Study 2, I accessed data through the AMAYS intervention and all the communications were done by the principal investigators of the project.

I communicated the scope and aims of the project directly with the teachers and the principal at study site 1. Teachers all agreed to participate and a general time frame for the intervention—Fall 2016-Spring 2017—was agreed upon. Additionally, the teachers agreed to provide the de-identified data collected in the study. Similar steps were taken to communicate the aim of the intervention with After-School All-Star administration and the participating teachers. However data collection was done by AMAYS team instead of by teachers.
The university’s Institutional Review Board approved the research. The approval process included the drafting of a parental notification letter that was distributed to the student participants. Students had the opportunity to opt out of the research.

I was an active participant in the research only to the extent that I introduced the study and collected the assent forms from the participating students for Study 1.

Care was taken to ensure the confidentiality of all participants involved in the study. Students were identified by their identification number provided through the school. The key to the code was kept by the teachers at Study 1 school site and by researchers at Study 2 sites. Teachers were not given access to identifiable results.
CHAPTER 4

RESULTS

Data Analysis

The quantitative data were compiled and compared to generate insights according to variable- and person center models (Roeser, Strobel, & Quihuis, 2002). On one hand, variable center model refers to the traditional variable-centered correlation techniques to understand the relationships between the independent variables and the outcome variable with the full sample. On the other hand, person center analysis requires researchers first to find out the subgroups among the sample. After that, the researchers will be able to compare the groups in terms of the outcome variable.

In order to do the initial analysis, first survey data was analyzed. All of the categorical predictors were dummy coded, meaning that correct answer was coded as 1 and the rest of the answer choices were coded as 0. I conducted a missing value analysis using SPSS 22 (IBM Corp., 2013) for Study 1 and Study 2 separately. Further information is reported in the following sections.

Since I included a large number of predictors, I conducted Pearson’s bivariate correlation (Pearson, 1920) among the independent and dependent variables to determine which variables to include to answer the research questions. After determining the variables that had significant relationships with the outcome variables through correlation analysis, I investigated the relationship between CT and those variables using simple and multiple regression analysis.

For the inclusion/exclusion criteria, I checked the time spent on the CT quiz for Study 1 and Study 2. Initially, I thought of excluding students who spent a fairly short time on the quiz, had they had significantly different scores than those who spent more time. For this purpose, I used a cutoff time length of five minutes, which on average corresponded to less than half a minute per item, which I deemed quite short. My analysis showed that students who completed the quiz in less than five minutes in Study 2 did not significantly differ with respect to their CT scores, compared
with students who spent more time on the quiz, $t(66) = -1.67, p = .10$. A similar result was found for the Study 1 participants, $t(54) = .08, p = .93$. As a result, no participant was excluded from the analysis based on the criterion of time spent on the quizzes.

In addition to the time spent on the quiz, I also checked if the attendance ratio in Study 2 played any role in students’ performance in CT-App quiz. For this purpose, I created a categorical variable which had 4 levels: students who attended from 0% to 25% of the available classes coded as 1, from 26% to 50% coded as 2, from 51% to 75% coded as 3, and 76% to 100% coded as 4. None of the comparisons showed a significant mean difference between the groups, $F(3, 59) = .81, p = .49$. After that, I created a binary variable that had 2 levels: students who attended from 0% to 50% coded as 1 and 51% and up coded as 2. The $t$-test results did not show any significant difference between the two groups, $t(63) = .16, p = .87$. The same procedure was used to compare students who attended 25% or less of the class available and those who attended more than 25%. No significant mean difference was found between two groups, $t(63) = .836, p = .836$. Given that no significant mean differences were found, I included all the participants who took the test in the data analysis for both of the studies.

A similar procedure was used to determine if the attendance had any role in CT practices variable in Study 2. Students with 25% or lower attendance rate demonstrated significantly lower CT practices than students with more than 25% attendance rate, $F(3, 127) = 9.036, p < .001$. The same analysis was conducted with the other three groups (students with attendance rate 25%-50%, 50%-75%, 75%-100%, respectively). The result showed that there was no significant difference between three groups, $F(2, 80) = 2.872, p = .062$ and their CT practices in Study 2. Therefore, the analysis regarding the relationship between learner characteristics and CT in Study 2 context only included those students with more than 25% attendance rate.
Data Cleaning

Outliers and Influential Points.

After all the data were combined for each study, the outlier and influential data point analyses were conducted through SPSS 22 (IBM Corp., 2013) as described by Stevens (1999). Outlier analysis was done automatically using SPSS syntax. The syntax for this analysis can be found in the Appendix C. The output of this syntax shows standardized residuals and Cook’s distance (Cook, 1977) for top 10 cases. A Cook’s distance cut-off point was calculated through the formula (4/n) so that the influential data points could be found and eliminated from the dataset. Further, any data points with standardized residuals outside the range of -2 and 2 were identified as outliers (Pituch, Stevens, & Whittaker, 2013).

Study 1.

Two of the cases’ standardized residuals (-2.585 and -2.478) were found outside the (-2, 2) range and thus they were removed from the dataset. None of the cases was identified as an influential data point based on Cook’s distance analysis in Study 1.

Study 2.

By employing the same procedure, three cases were identified as outliers in Study 2 and they were omitted from the dataset. None of data points was identified as an influential data point.

Assumption Checking

The key assumptions of regression analysis are linearity, independence, homoscedasticity, and normality (Pituch, Stevens, & Whittaker, 2013). I also checked multicollinearity.

Regression Assumptions for Study 1.

Linearity.

The independent and dependent variables need to have a linear relationship to meet the assumption of linear regression. To test this assumption, the scatterplot of residuals—the difference between the observed values of the dependent variable and the predicted values—against predicted
values was used in this study. As can be seen from Figure 14, residuals are mostly symmetric around the zero line, indicating validity of the linearity assumption.

**Figure 14.** Scatterplot for Standardized Residuals vs. Predicted Value

**Independence:**

For testing independence, I again used the residual analysis plot shown in Figure 14. Given that it is not possible to directly check independence, I checked for uncorrelated errors. As no clustering of residuals exists, I concluded that uncorrelated error assumption is valid in this case.

**Homoscedasticity: Testing homogeneity of error variance.**

The scatter plot which was drawn for linearity (see Figure 14) is also used to test homogeneity of error variance. In order to meet the homoscedasticity assumption, the residual should be equally
or homogenously distributed around the zero line. Figure 14 showed that this assumption was tenable.

**Tests on Normality of Residuals.**

The general way to test the normality assumption for regression analysis is to obtain Q-Q plots between the residuals of the dependent and independent variables. The resulting Q-Q plot for Study 1 is shown below in Figure 15. The normality assumption is satisfied, because as seen in Figure 15, the points are clustered around the diagonal line.

![Normal P-P Plot of Regression Standardized Residuals](image)

*Figure 15. Normal P-P Plot of Regression Standardized Residuals*

**Tests on Multicollinearity.**

Self-efficacy is significantly correlated with interest, therefore multicollinearity was checked by the variance inflation factor (VIF) metric (Pituch, Stevens, & Whittaker, 2013). The VIF predicts how much the regression coefficient variance is inflated due to multicollinearity. VIF metric is readily available on SPSS output. As a rule of thumb, VIF bigger than 5 is considered as cause for
concern. VIF was 1.036 for this regression model so multicollinearity was not deemed to be a concern in the analysis.

**Regression assumptions for Study 2.**

**Linearity.**

The independent and dependent variables need to have a linear relationship to meet the assumption of linear regression. To test this assumption, the scatterplot of residuals against predicted values was used. As can be seen from Figure 16, residuals are mostly symmetric around the zero line, indicating validity of the linearity assumption.

![Figure 16. Scatterplot for Standardized Residuals vs. Predicted Value](image)

**Independence:**

For testing independence, I again used the residual analysis plot shown in Figure 16 and checked for uncorrelated errors. As no clustering of residuals exists, I concluded that uncorrelated error assumption is also valid in this case.
Homoscedasticity: Testing homogeneity of error variance.

Similar procedure stated in the above section with regard to this assumption was employed to test the homogeneity of error variance. Since Figure 16 shows that residuals are homogenously distributed around the zero line, this assumption was tenable.

Tests on Normality of Residuals.

The resulting Q-Q plot for Study 2, which was used for checking the normality assumption, is shown below in Figure 17. The distribution seems normal, because the points clustered around the diagonal line.

![Normal Q-Q Plot of ct_quiz](image)

Figure 17. Q-Q Plot of CT Quiz

Tests on Multicollinearity.

Study 2 included one predictor to predict a dependent variable through regression analysis because that variable was found significantly correlated with the dependent variable. Thus, there was no need for multicollinearity testing.
**Logistic Regression Assumptions for Study 1 and Study 2.**

Logistic regression assumptions differ from linear regression assumptions. Logistic regression requires linearity between the predictor and outcome variable, residuals do not have to be normally distributed, and homoscedasticity is not an issue for logistic regression models. However, the following assumptions need to be checked: the observations need to be independent of each other; there should not be multicollinearity among the independent variables; and it requires a large sample size. In order to make observations independent, the data should not be in a repeated measure format or matched. Also to satisfy the multicollinearity assumption, the independent variables should not be highly correlated with each other. Sample size required for logistic regression can be found through power analysis (Lipsey, 1990; Faul, Erdfelder, Buchner, & Lang, 2013).

Our analysis showed that the Study 1 dataset satisfied the logistic regression assumptions and hence was appropriate to be used in logistic regression analysis to understand the relationships between learner characteristics and CT practices. Study 2 dataset also satisfied all the requirements, but sample size is not large enough for drawing reliable statistical inference. Specifically, in order to obtain an adequate sample size using an alpha level of .05, a power of .80, and a larger effect size (odd ratio = 2.856), there should be 71 participants. However, Study 2 had only 29 participants that took both of the tests.

**Participant flow**

The participant flow through each stage of this current research can be found in Figure 18.
Figure 18. Participant flow through Study 1 and Study 2 data collection at four time points: Spring 2017, pretest and posttest; Fall 2017, pretest and posttest.

Missing data

Study 1.

Study 1 included 166 participating students, 71 of whom had CT quiz scores. Among all of the participants, 110 of them took the learner profile survey. In order to understand the causes of the data that were missing such as Missing Completely at Random (MCAR), the Little’s test (1998) was conducted through SPSS. The test results showed that the missing data was completely at random for Study 1, $\chi^2 (1) = 1.431, p = .232$. This means that there was not any significant relationship between the missing and existing data points, and the missing data was a random subset of the data. Thus, no further analysis was required.

Study 2.

The total of 146 students participated in either one of the pretests or posttests, or in both. Among those participants, 87 of the students participated in the learner profile survey and 67 of the students participated in the CT quiz. The same statistical procedure took place to examine the missing data. The test results showed that the missing data was completely at random for Study 2, $\chi^2 (1) = .352, p = .553$. Thus, no further analysis was conducted.
Descriptive statistics

Study 1.

In Study 1, 31% of the participants were sixth graders, 32% seventh graders, and 36% eighth graders. For the Study 1 participants who took the learner survey, 33% rated their self-efficacy on learning how to build phone apps as “strongly disagree”, 3% “disagree”, 20% “neutral”, 30% “agree”, and 14% “strongly agree”. Furthermore, 36% them rated their interest on learning about computing technology as “strongly agree”, 31% “agree”, 17% “neutral”, 16% “disagree”, and 1% “strongly disagree”. Students’ prior experience with creative computing technologies were also categorized: 40% of the participants expressed their prior experience as “beginner”, 28% “specialist”, 10% “explorer”, and 23% “generalist”. Finally, students’ initial purpose were also obtained through the survey: 8% of them wanted to be better than their peers, 13% did not want to fail, 37% wanted to understand how the system works, and 42% wanted to have fun.

In addition to learner characteristics, there were two types of the primary variables that measured CT. One of them was the Computational Concepts and Practices Quiz score. Eleven questions from the quiz which established the validity and the reliability in Study 1 context were summed up for students’ final quiz score. The other variable was the students’ demonstration of CT practices in their digital artifacts; 61% of the participants either did not develop the final artifact or did not demonstrate CT practices in their apps. On the other hand, 39% of the students demonstrated CT practices in their artifacts. Descriptive statistics regarding those variables can be found in Table 3. More specifically, Table 3 summarizes the sample size, minimum and maximum values, mean, and standard deviation of all the variables for Study 1.
Table 3

Descriptive Statistics for Study 1

<table>
<thead>
<tr>
<th>Variable</th>
<th>Sample Size</th>
<th>Min</th>
<th>Max</th>
<th>M</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Demographics</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td>110</td>
<td>1</td>
<td>3</td>
<td>2.03</td>
<td>.826</td>
</tr>
<tr>
<td>Gender</td>
<td>110</td>
<td>0</td>
<td>1</td>
<td>.34</td>
<td>.47</td>
</tr>
<tr>
<td><strong>Learner Characteristics</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior Experience</td>
<td>110</td>
<td>0</td>
<td>3</td>
<td>1.09</td>
<td>1.17</td>
</tr>
<tr>
<td>Prior Depth</td>
<td>110</td>
<td>0</td>
<td>1</td>
<td>.33</td>
<td>.47</td>
</tr>
<tr>
<td>Prior Breadth</td>
<td>110</td>
<td>0</td>
<td>1</td>
<td>.48</td>
<td>.47</td>
</tr>
<tr>
<td>Self-Efficacy</td>
<td>110</td>
<td>0</td>
<td>5</td>
<td>2.7</td>
<td>1.69</td>
</tr>
<tr>
<td>Interest</td>
<td>110</td>
<td>1</td>
<td>5</td>
<td>3.83</td>
<td>1.07</td>
</tr>
<tr>
<td>GoalFail</td>
<td>110</td>
<td>0</td>
<td>1</td>
<td>.12</td>
<td>.33</td>
</tr>
<tr>
<td>GoalFun</td>
<td>110</td>
<td>0</td>
<td>1</td>
<td>.43</td>
<td>.49</td>
</tr>
<tr>
<td>GoalUnderstand</td>
<td>110</td>
<td>0</td>
<td>1</td>
<td>.38</td>
<td>.48</td>
</tr>
<tr>
<td>GoalBetter</td>
<td>110</td>
<td>0</td>
<td>1</td>
<td>.07</td>
<td>.26</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quiz Score</td>
<td>69</td>
<td>2</td>
<td>7</td>
<td>4.85</td>
<td>1.37</td>
</tr>
<tr>
<td>CT Practices</td>
<td>146</td>
<td>0</td>
<td>1</td>
<td>.42</td>
<td>.49</td>
</tr>
</tbody>
</table>

**Study 2.**

Of the 146 students, around 36% of them took the learner profile survey, 9% of were sixth graders, 16% seventh graders, and 6% eight graders. Two percent of the participants who participated in profile survey rated their self-efficacy on learning how to build phone apps as “disagree”, 7% as "neutral”, 9% “agree”, and 21% “strongly agree”. Furthermore, 17% them rated their interest on learning about computing technology as “strongly agree”, 20% “agree”, 4% “neutral”, and 2% “disagree”. Students’ prior experience with creative computing technologies were also categorized; 4% of the participants expressed their prior experience as “beginner”, 2% “specialist”, 11% “explorer”, and 20% “generalist”. Finally, students’ initial purpose were also obtained through the survey; 16% of them wanted to be better from their peers, 18% did not want to fail, 4% want to understand how the system works, and 1% wanted to have fun.

In Study 2, a slightly changed version of the CT-App was used to examine students’ CT. Like Study 1, Computational Concepts and Practices Quiz score was calculated but this time with 10 questions. The other outcome variable was the students’ CT practices in their digital artifacts. De-
Descriptive statistics regarding those variables can be found in Table 4. Furthermore Table 4 summarizes the sample size, minimum and maximum values, mean, and standard deviation of all the variables’ for Study 2. More information can be found in Table 4.

Table 4

<table>
<thead>
<tr>
<th>Variable</th>
<th>Sample Size</th>
<th>Min</th>
<th>Max</th>
<th>M</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demographics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td>76</td>
<td>1</td>
<td>3</td>
<td>1.78</td>
<td>.776</td>
</tr>
<tr>
<td>Gender</td>
<td>76</td>
<td>1</td>
<td>2</td>
<td>1.70</td>
<td>.490</td>
</tr>
<tr>
<td>Learner Characteristics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior Experience</td>
<td>73</td>
<td>0</td>
<td>3</td>
<td>2.39</td>
<td>.718</td>
</tr>
<tr>
<td>Prior Depth</td>
<td>73</td>
<td>0</td>
<td>1</td>
<td>.55</td>
<td>.52</td>
</tr>
<tr>
<td>Prior Breadth</td>
<td>73</td>
<td>0</td>
<td>1</td>
<td>.91</td>
<td>.27</td>
</tr>
<tr>
<td>Self-Efficacy</td>
<td>86</td>
<td>1</td>
<td>5</td>
<td>4.20</td>
<td>1.05</td>
</tr>
<tr>
<td>Interest</td>
<td>87</td>
<td>1</td>
<td>5</td>
<td>4.07</td>
<td>1.02</td>
</tr>
<tr>
<td>GoalFail</td>
<td>87</td>
<td>0</td>
<td>1</td>
<td>.47</td>
<td>.502</td>
</tr>
<tr>
<td>GoalFun</td>
<td>87</td>
<td>0</td>
<td>1</td>
<td>.63</td>
<td>.485</td>
</tr>
<tr>
<td>GoalBetter</td>
<td>87</td>
<td>0</td>
<td>1</td>
<td>.238</td>
<td>.436</td>
</tr>
<tr>
<td>GoalUnderstand</td>
<td>87</td>
<td>0</td>
<td>1</td>
<td>.49</td>
<td>.503</td>
</tr>
<tr>
<td>Performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quiz Score</td>
<td>67</td>
<td>0</td>
<td>8</td>
<td>2.78</td>
<td>1.79</td>
</tr>
<tr>
<td>CT Practices</td>
<td>131</td>
<td>0</td>
<td>1</td>
<td>.167</td>
<td>.375</td>
</tr>
</tbody>
</table>

Primary Research Question: What are the Relationships between Learner Characteristics and CT Quiz and Practices?

**Correlation between CT-App Quiz and Learner Characteristics.**

A Pearson product-moment correlation coefficient was computed to assess the relationships between the learner characteristics and CT concepts and practices quiz score. Among the selected independent variables, CT-App quiz score had a significant positive correlation with participating students’ self-efficacy ($r = .385, n = 44, p = .010$) and interest ($r = .333, n = 44, p = .027$) in Study 1. In Study 2, none of the variables were found significantly related to CT-App quiz score.

**Correlation between CT Practices and Learner Characteristics.**

In addition, a Pearson product-moment correlation coefficient was also computed to examine the correlation between CT practices that emerged in their do-it-yourself type, guided practices activities and cPBL projects and learner characteristics. The data for Study 1 showed that students’
self-efficacy had a significant positive relationship with CT practices \( (r = .564, n = 110, p < .001) \). On the other hand, their gender (male = 0, female = 1) had significant negative correlation with the dependent variable \( (r = -.225, n = 110, p = .018) \).

The results for Study 2 showed that the students’ self-efficacy had a significant positive relationship with CT practices \( (r = .384, n = 50, p = .006) \). Furthermore, students who had the goal to understand how to build an app had a similar significant relationship with their CT practices \( (r = .352, n = 52, p = .011) \).

All other correlation analyses among the dependent and independent variables did not show any significant relationships in both the learning environments in Study 1 and Study 2. Please see Table 5 for further information.

Table 5

<table>
<thead>
<tr>
<th></th>
<th>Study 1</th>
<th>Study 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quiz</td>
<td>CT Practices</td>
</tr>
<tr>
<td>Age</td>
<td>.019</td>
<td>.197</td>
</tr>
<tr>
<td>Prior Depth</td>
<td>.153</td>
<td>-.053</td>
</tr>
<tr>
<td>Prior Breadth</td>
<td>.154</td>
<td>-.113</td>
</tr>
<tr>
<td>Prior Experience</td>
<td>-.193</td>
<td>-.157</td>
</tr>
<tr>
<td>Gender</td>
<td>-.123</td>
<td>-.225**</td>
</tr>
<tr>
<td>Self-Efficacy</td>
<td>.385*</td>
<td>.564**</td>
</tr>
<tr>
<td>Interest</td>
<td>.333*</td>
<td>-.025</td>
</tr>
<tr>
<td>GoalFail</td>
<td>-.084</td>
<td>-.134</td>
</tr>
<tr>
<td>GoalFun</td>
<td>.138</td>
<td>.040</td>
</tr>
<tr>
<td>GoalUnderstand</td>
<td>-.238</td>
<td>.221</td>
</tr>
<tr>
<td>GoalBetter</td>
<td>.231</td>
<td>-.064</td>
</tr>
</tbody>
</table>

Note. *p < .05, **p < .01

Table 6 shows a summary of the results for all the hypotheses. I describe the details regarding the accepted hypotheses in the next section.
Table 6
*Hypothesis Acceptance/Rejection table.*

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1  There will be a correlation between students’ self-efficacy level and CT quiz score in the formal computer class.</td>
<td>Accepted</td>
</tr>
<tr>
<td>H2  There will be a correlation between students’ self-efficacy level and CT practice in the formal computer class.</td>
<td>Accepted</td>
</tr>
<tr>
<td>H3  There will be a correlation between students’ self-efficacy level and CT quiz score in the after-school environment.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H4  There will be a correlation between students’ self-efficacy level and CT practice in the after-school environment.</td>
<td>Accepted</td>
</tr>
<tr>
<td>H5  There will be a correlation between students’ interest level and CT quiz score in the formal computer class.</td>
<td>Accepted</td>
</tr>
<tr>
<td>H6  There will be a correlation between students’ interest level and CT practice in the formal computer class.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H7  There will be a correlation between students’ interest level and CT quiz score in the after-school environment.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H8  There will be a correlation between students’ interest level and CT practice in the after-school environment.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H9  There will be a correlation between students’ goal orientation and CT quiz score in the formal computer class.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H10 There will be a correlation between students’ goal orientation and CT practice in the formal computer class.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H11 There will be a correlation between students’ goal orientation and CT quiz score in the after-school environment.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H12 There will be a correlation between students’ goal orientation and CT practice in the after-school environment.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H13 There will be a correlation between students’ gender and CT quiz score in the formal computer class.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H14 There will be a correlation between students’ gender and CT practice in the formal computer class.</td>
<td>Accepted</td>
</tr>
<tr>
<td>H15 There will be a correlation between students’ gender and CT quiz score in the after-school environment.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H16 There will be a correlation between students’ gender and CT practice in the after-school environment.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H17 There will be a correlation between students’ grade level and CT quiz score in the formal computer class.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H18 There will be a correlation between students’ grade level and CT practice in the formal computer class.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H19 There will be a correlation between students’ grade level and CT quiz score in the after-school environment.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H20 There will be a correlation between students’ grade level and CT practice in the after-school environment.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H21 There will be a correlation between students’ prior experience with creative computing and CT quiz score in the formal computer class.</td>
<td>Rejected</td>
</tr>
<tr>
<td>H22 There will be a correlation between students’ prior experience with creative computing and CT practice in the formal computer class.</td>
<td>Rejected</td>
</tr>
</tbody>
</table>
Secondary Research Question: How much of the variation in CT Quiz scores do selected variables explain in Study 1 and Study 2?

Study 1.

A multiple linear regression analysis was conducted to predict CT quiz score based on students’ self-efficacy in the pretest that took place in Study 1. A significant regression equation was found, $F(2, 40) = 5.219, p = .010$, and $R^2 = .207$. Participants’ predicted CT quiz score is equal to:

$$3.392 + .291(self - efficacy) + .206(interest),$$

where self-efficacy and interest were coded in a Likert-scale from 1 through 5 from “strongly disagree” to “strongly agree”. Participant’s CT quiz score increased .291 points significantly for 1-point increase in the self-efficacy (see Table 7 for more information).

Although students’ interest was found significantly related to CT in a regression analysis, it became insignificant when self-efficacy was added into the model. Besides this, the correlation analysis between self-efficacy and interest were also found significant ($r = .243, n = 108, p = .011$). Since this research is an exploratory study to understand the relationship between learner characteristics and CT, I conducted a mediation analysis between interest and CT, as self-efficacy being a mediator variable as a post-hoc analysis. However the relationship between interest and CT was not significantly reduced when self-efficacy was included, Sobel’s test $= 1.3661, p = .1718$. In addition to mediation analysis, I conducted post-hoc power analyses. A post-hoc power analysis revealed that the power of this multiple regression with medium effect size (Cohen’s $d = .026$) was .81 which is in acceptable range (Stevens, 1991).
Table 7

Study 1 Summary of Regression Analysis for Learner Characteristics Predicting CT-App Quiz Score (n=44)

<table>
<thead>
<tr>
<th>Predictor</th>
<th>B</th>
<th>SE B</th>
<th>B</th>
<th>t</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-efficacy</td>
<td>.281</td>
<td>.113</td>
<td>.346</td>
<td>2.484</td>
<td>.017*</td>
</tr>
<tr>
<td>Interest</td>
<td>.313</td>
<td>.173</td>
<td>.252</td>
<td>1.810</td>
<td>.077</td>
</tr>
</tbody>
</table>

Note. *p < .05

Study 2.

Since none of the variables were found related to CT-App quiz score, power analysis was conducted. The power analysis using an alpha level of .05 with a sample size of 23 and an effect size of .282 revealed that the power was .38 for the correlation analysis. The highest correlation coefficient was chosen as the effect size, because the rest of them would have less power than the variable that had the strongest correlation with the dependent variable.

Secondary Research Question: How much of the variation in CT practices do selected variables explain in Study 1 and Study 2?

Study 1.

A logistic regression analysis was conducted to predict the participants’ CT practices using gender and self-efficacy as independent predictors. A test of the full model against a constant-only model was statistically significant, indicating that the variables reliably distinguished between students who employed CT practices in their digital artifacts and those who did not ($\chi^2 = 44.204, p < .001$ with $df = 2$).

Nagelkerke’s R$^2$ of .446 indicated a moderately strong relationship between prediction and grouping. Prediction success overall was 77% (68% for students without CT practices and 88% for students with CT practices). The Wald criterion demonstrated that gender ($p = .05$) and self-efficacy ($p < .001$) made a significant contribution to prediction. Also, Cox and Snell R Square
showed that 33% of the variance in the outcome variable could be explained by the model. The logistic regression model was as follows:

\[
\ln(ODDS) = -2.456 - 1.025 \text{Gender} + .890 \text{Self-efficacy}
\]

The Exp(B) value associated with self-efficacy was 2.434. Hence, when self-efficacy level was raised by one unit the odds ratio is 2.434 times as large, and therefore students with high self-efficacy level were 2.434 more times likely to demonstrate CT in their digital artifacts. Besides this, the Exp(B) value associated with gender was .359 which means that male students were more likely to demonstrate CT practices than female students in Study 1 (see Table 8).

Table 8

Study 1 Summary of Logistic Regression Analysis for Learner Characteristics Predicting CT practices

<table>
<thead>
<tr>
<th>Predictor</th>
<th>B</th>
<th>SE</th>
<th>Wald</th>
<th>Sig.</th>
<th>(e^B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-efficacy</td>
<td>.890</td>
<td>.180</td>
<td>24.443</td>
<td>.000*</td>
<td>2.434</td>
</tr>
<tr>
<td>Gender</td>
<td>-1.025</td>
<td>.523</td>
<td>3.850</td>
<td>.050</td>
<td>.359</td>
</tr>
<tr>
<td>Constant</td>
<td>-2.456</td>
<td>.619</td>
<td>15.754</td>
<td>.000</td>
<td>.086</td>
</tr>
</tbody>
</table>

*Note. *\(p < .05\)

Study 2.

The same model from research question one with the same analyses from Study 1 addressing research question 2 were repeated with data from Study 2. The results showed that a test of the full model against a constant-only model was statistically significant, indicating that the variable reliably distinguished between students who employed CT practices in their digital artifacts and those who did not (\(\chi^2 = 10.447, p = .005\) with \(df = 2\)).

Nagelkerke’s \(R^2\) of .405 indicated a moderately strong relationship between prediction and grouping. Prediction success overall was 79.3% (81.3% for students without CT practices in their apps and 76.9% for students with CT practices). Also, Cox and Snell R Square showed that 30% of the variance in the outcome variable could be explained by the specified model.
The logistic regression model was as follows:

\[
\ln(ODDS) = -5.388 + 1.049(\text{Self-efficacy}) + 1.444(\text{goalUnderstand})
\]

Although the logistic regression model was found significant, the Wald criterion demonstrated that neither self-efficacy \( (p = .062) \) nor the students’ goal orientation of understanding how to build an app \( (p = .126) \) made a significant contribution to prediction. Table 9 shows a summary of the analysis.

Since the Exp(B) value associated with self-efficacy and interest were not found statistically significant, a post-hoc power analysis was conducted. As found in the logistic regression assumption checking, the sample size was not large enough to have a power of .80 at an alpha level of .05 with a large effect size (odd ratio = 2.856).

Table 9

*Study 2 Summary of Logistic Regression Analysis for Learner Characteristics Predicting CT practices*

<table>
<thead>
<tr>
<th>Predictor</th>
<th>B</th>
<th>SE B</th>
<th>Wald</th>
<th>Sig.</th>
<th>( e^B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-efficacy</td>
<td>1.049</td>
<td>.562</td>
<td>3.481</td>
<td>.062</td>
<td>2.856</td>
</tr>
<tr>
<td>GoalUnderstand</td>
<td>1.444</td>
<td>.945</td>
<td>2.338</td>
<td>.126</td>
<td>4.240</td>
</tr>
<tr>
<td>Constant</td>
<td>-5.388</td>
<td>2.390</td>
<td>5.081</td>
<td>.024*</td>
<td>.005</td>
</tr>
</tbody>
</table>

*Note. *\( p < .05 \)*
CHAPTER 5

DISCUSSION

Although there has been an increasing number of computing and computational thinking (CT) research studies, computational thinking research for middle-school students is still at its infancy. While prior research in computer science education on the relationships between learner characteristics and students’ programming performance has confirmed the importance of learner characteristics on the outcomes, the focus of our study was computational thinking, a broader concept than programming (computer science content). In addition, most of the related studies in the literature focused on college students, leaving open questions about the relationship between a range of learner characteristics and middle-school students’ CT skills.

In this current research, I aimed to address some of these open questions by utilizing a learner characteristics survey, a computational thinking quiz, and an artifact analysis. Participating students first completed the learner characteristics survey. Next, they learned app building through MIT’s App Inventor tool with modularized worked examples and do-it-yourself hands-on activities. After that, the students completed problem-based programming tasks. The students then took a posttest for measuring their CT. The students’ digital artifacts were then explored with the aim of determining whether or not the computational thinking practices that are highlighted in Brennan and Resnick’s CT framework presented in the digital artifacts, and whether or not this variable correlates with the students’ learner characteristics.

This chapter begins with a discussion on the results of the exploration of the relationships between learner characteristics and middle-school students’ CT as measured by CT quiz and the artifact analysis in two different contexts: formal computer class and informal after-school setting. This chapter also reviews the research questions outlined in Chapter 4. The chapter then describes the limitations of the study, explores the implications of its research findings on both computational thinking and instructional design, and, finally, it suggests some proposed future research.
Review of the Research Questions

This study was framed by one main research question with four subquestions or Research Questions 1, 2, 3, and 4. The intent of these four research questions was to explore the relationships between learner characteristics and middle-school-aged students’ CT in a formal and an informal environment during a programming course. The main research question asked about the nature of the relationships between the learner characteristics and CT. The main research question was broken down into the four research questions so that they could be accordingly responded to by the results of this current research. Table 10 summarizes the overall findings.

Table 10
Summary of the Overall Findings.

<table>
<thead>
<tr>
<th>Variables of Interest</th>
<th>Setting</th>
<th>Formal Classroom</th>
<th>Informal, After-School Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables Correlated with Outcome</td>
<td>CT Quiz</td>
<td>Self-efficacy Interest</td>
<td>Self-efficacy</td>
</tr>
<tr>
<td>Variables Correlated with Outcome</td>
<td>CT Practices</td>
<td>Self-efficacy Gender</td>
<td>Self-efficacy Gender</td>
</tr>
</tbody>
</table>

Research Questions 1 and 2.

What is the nature of the relationships between middle-school students’ learner characteristics and CT as measured by a multiple-choice assessment (CT-App) (1) in a formal learning environment and (2) in an informal learning environment?

These two research questions sought to determine whether relationships existed between middle-school students’ learner characteristics and CT, as evaluated by their performance in custom-built, CT quiz in an app building context, and, if so, to explore the nature of those relationships.
The correlation analysis showed that students’ initial self-efficacy and interest in app building activities had a significant positive relationship with their CT quiz score in the middle-school formal computer course.

More specifically regression analysis showed that self-efficacy explained a significant amount of variance of middle-school-aged students’ CT in formal classroom settings independently. This finding aligns with the previous research that showed the positive effect of self-efficacy on programming performance (Lishinski et al., 2016; Wiedenback, 2015). Lishinski et al. investigated the relationship between self-regulated learning constructs such as, self-efficacy, metacognitive strategies, and goal orientation and programming performance through structural equational modeling. They found that self-efficacy was the most important predictor of students’ outcome. Studies outside of the U.S. also showed similar results. For example, Cigdem and Oncu (2015) conducted a study with 267 military vocational college students in Turkey to examine the relationships between self-regulation constructs—including self-efficacy, anxiety, interactivity, satisfaction with usefulness of the system—and learners’ academic achievement in an online computer programming course. The findings of Cigdem and Oncu’s study demonstrated that only perceived self-efficacy and usefulness were significantly correlated with course grades. Further analysis showed that perceived self-efficacy was the strongest significant positive predictor of the course grade. This might be due to the positive effect of self-efficacy on student learning in general. As Bandura (1977) stated, self-efficacy influences learning in three ways: activity choices, the amount of effort they want to expand, and how much persistence in their effort dealing with challenging situations.

In addition to self-efficacy, students’ interest in the app building activities were found to be significantly correlated. However, when interest and self-efficacy were added to the multiple regression model, the results showed that the variance in CT quiz score explained by interest variable was not significant but such variance explained by self-efficacy was statistically
significant. To better understand the relationships between student interest, self-efficacy, and students’ CT, I further conducted a mediation analysis, which is an appropriate quantitative model to understand those relationships as such analysis helps to clarify the nature of the relationships between the independent variables, mediator variables, and dependent variables (MacKinnon, 2012). Figure 19 shows the suggested mediation model, which was not found statistically significant to explain the differences among students for CT quiz score.

![Mediation model](Figure 19)

In addition to regression and mediation analysis, the interaction effect of self-efficacy and interest was also assessed. An interaction effect would help to clarify if students with high self-efficacy and high interest would outscore students with low self-efficacy and low interest. The findings showed that there was not a significant interaction effect of self-efficacy and interest on CT.

Since the power was found to be .81 which is just .01 more than acceptable range, I believe that the overall lack of statistical significance found for the interest variable is due to small sample size. Thus, further studies should investigate the relationship between interest and CT with a larger sample size because similar studies at the college level CSEd have confirmed the positive effect of interest on the programming performance (e.g., Beyer, 2014; Haungs et al., 2012). However, recent K-12 CSEd studies have not confirmed the relationship between interest and CT, including this current research. In the regression and mediation analyses in this research, statistically significant results regarding this relationship between the interest and CT were not found; and therefore, the findings in this study also did not prove such positive relationship. Furthermore, researchers in
some recent studies argued that interest was not significant in a K-12 computing context. For example, Grover et al. (2017) argued that interest was not significantly correlated with the programming outcome in the high school setting. In addition, Yadav et al., (2017) stated that after an Hour of Code activity in a primary school context, female students did not show interest in doing a similar type of activity in the future whereas male students did despite the fact that female students’ scores were as good as male students’ in the programming activity. In this current research, the focus of the assessment was on CT rather than programming performance and its findings showed significant positive correlations with some of the variables included in this research. Taken together, special attention must be given to the fact that the interventions helped students regardless of their gender but further analysis showed that female students were not interested in the CT practices as much as male students were in the middle-school computing class.

Although correlation analysis results in this study aligned with those in the literature and showed a positive relationship between self-efficacy and CT quiz score in a formal education classroom (Research Question 1), self-efficacy was not significantly correlated with CT quiz score in informal after-school environment (Research Question 2). None of the learner characteristics were found significantly correlated to CT quiz score in an informal context. This might be due to three reasons: the nature of the students who chose to attend the program, the assessment format, and the small sample size. First of all, students in the after-school program came to the program voluntarily which led the group to be more homogenous than the formal classroom where students had to be there anyway. This was found in the data. For example, students in the informal learning context scored high on self-efficacy in the pretest; the mean for self-efficacy was 4.16 out of 5. However, there was not much variation in the self-efficacy scores, therefore, correlation analysis failed to find a possible relationship between self-efficacy and the CT quiz scores, and for this similar reason, relationships between all other variables and CT quiz scores were not found. Therefore, these
results should be interpreted with caution. Secondly, students who attended the after-school pro-
gram might not have taken the quiz seriously because their score did not count for their grade. The
mean of the CT quiz for students in informal learning environment was 2.76 out of 12 questions.
This might also be due to the fact that students might have seen the multiple-choice quiz as a
school-type assignment and chosen not to do their best on it. Last, the small sample might be an-
other reason that none of the variables were found significantly related to students’ CT in the infor-
mal learning context. The sample size for students who took the learner profile survey and com-
pleted CT quiz was only 15. Further studies are needed with a larger sample size.

**Research Questions 3 and 4.**

*What is the nature of the relationships between middle-school students’ learner characteristics
and CT practices as measured by artifact analysis (3) in a formal learning environment and (4) in
an informal learning environment?*

Using the same procedure as for Research Questions 1 and 2, correlation analysis showed that
students’ self-efficacy significantly correlated with students’ CT practices demonstrated in their
digital artifacts regardless of the learning context. In addition to self-efficacy, students’ goal orien-
tation and CT practices were also significantly related to each other in the informal afterschool con-
text.

Those students whose goal orientations included understanding how to build an app showed CT
practices in their apps more than those without that goal orientation. In the literature, students with
the desire of understanding how to develop new skills, new knowledge or new abilities are referred
to having *mastery goal approach*, whereas students who are extrinsically motivated are referred to
having *performance goal approach* (Erhel & Jamet, 2013). In this research, the findings showed
that students for whom one of the goal was mastery goal approach in the study demonstrated CT
practices more than those students without that goal orientation in the informal learning environ-
ment. Previous studies also showed differences in students’ performance in the course depending
on their goal approach. For example, similar to this research, Zingaro (2015) found positive relationship between exam score and students’ mastery goal approach. More specifically students with performance goal scored lower on final exams irrespective of they were taught by a peer or an instructor. The possible explanation for the negative relation with performance goal approach and the positive correlation between CT quiz score and mastery goal approach, is that students with performance goal approach tend to put much effort on handling the task in order to avoid looking incompetent rather than mastering the content to acquire new skill (Zarankin, 2008). Although the findings of this study aligns with those in the literature, more investigation is needed to understand how the differences in students’ goal approach specifically affect their CT practices.

Besides goal orientation, students’ self-efficacy was found significantly correlated to CT practices demonstrated in the digital artifacts in an informal learning environment. However the results of the logistic regression analysis were not significant. The power analysis showed that it might be due to the small sample size, which was a total of 29 participating students.

The findings of this study in the formal learning context demonstrated similar relationship between self-efficacy and the CT practices found in students’ artifacts. The results of the follow-up, logistic regression analysis indicated that self-efficacy was significant to explain the variation in middle-school computing course. In addition to this, gender was also found to be significantly correlated with the students’ demonstration of CT practices in the formal context. Male students used CT practices in their apps more often than did female students. Table 11 showed the probabilities of demonstrating CT practices in the digital artifacts for female and male students in each self-efficacy level. For example, the probability of demonstrating CT practices for male students with the highest level of self-efficacy was .88, whereas female students with the same self-efficacy level had a .72 probability of using CT practices in their apps.
Table 11

Probabilities of demonstrating CT practices in the digital artifacts by gender and self-efficacy level

<table>
<thead>
<tr>
<th>Self-efficacy Point</th>
<th>Female</th>
<th>Male</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.06</td>
<td>.17</td>
</tr>
<tr>
<td>2</td>
<td>.15</td>
<td>.33</td>
</tr>
<tr>
<td>3</td>
<td>.30</td>
<td>.55</td>
</tr>
<tr>
<td>4</td>
<td>.51</td>
<td>.75</td>
</tr>
<tr>
<td>5</td>
<td>.72</td>
<td>.88</td>
</tr>
</tbody>
</table>

**Additional Findings.**

Since there was a significant gender difference in students’ CT practices found in the digital artifacts in the formal learning environment, I investigated additional variables that might have gender differences as well. The results of $t$-test showed that there was a significant mean difference between female and male students only in terms of their interest in computing activities ($M = .873$, $SD = .201$). Male students indicated more interest in the activities than did female students, $t (107) = 4.349$, $p < .001$. This finding might have an indirect relation with students’ performance on practicing CT in their digital artifacts. Further analysis is needed to understand the relationships between students’ interest, gender, and their CT practices in their apps.

**Limitations**

Although some patterns have emerged from the contributions of recent studies to the field of CT in a computing context, I discuss in this section a few limitations of this research that reduce its power and restricts the type of broader conclusions, and generalizable implications which researchers and educators can draw on. The findings of this study are limited by five predominant issues: *research design*, *sample size*, *duration of the research*, *instrument*, and *data collection*.

**Research Design.** Since this study was a pure correlation study, the findings of this research have limitations in drawing causal conclusions. Regression analysis does not provide further causal explanations between variables. For this reason, the results should be interpreted cautiously. Future studies could use these results as a basis for knowing which constructs merit further attention, and then experimental studies could be conducted to enable causal conclusions to be drawn.
Sample Size. Forty-two participating students in Study 1 and 23 in Study 2 had complete data for the regression analysis that was conducted to find the relationships between the selected variable and CT quiz scores. The results of power analysis showed that larger samples were needed for the study in the informal context. Therefore, the findings from that context are necessarily constrained, and must only be interpreted more generally.

Duration. The duration of both studies was not long enough for the teachers to teach all the CT concepts and practices assessed in CT quiz. Study 1 instruction was 7 hours long and Study 2 lasted for approximately 16 hours. However, student attendance in Study 2 fluctuated a lot due to the informal after-school context. This was mitigated to some extent by including only the participants who attended more than 25% of the sessions available to them. However, still, the duration of both studies was not adequate for the participants to master all of the CT concepts and practices outlined in Brennan and Resnick’s (2012) framework (Hatley, 2016).

Instrument. CT-App quiz was validated and found reliable in the formal learning context in Study 1. However, I was not able to validate the quiz due to the small sample size in the informal learning environment in Study 2. Further exploration is needed to test its validity and reliability in middle-school, informal settings.

Data Collection. Qualitative data collection did not take place in the formal context in Study 1 and field notes collected in the informal environment in Study 2 did not focus on the their CT practices or individual learner characteristics. This issue resulted in the lack of understanding of the individual differences that occurred during the studies. However, qualitative data could help to gain insights into the differences between the digital artifacts of female and male students. To ameliorate this limitation, the structured qualitative data should be collected in future studies.

Implications

This study was conducted to understand which learner characteristics are related to middle-school-aged students’ CT in a visual block-based computing context. The identification of these
learner characteristics is important to make informed decisions in terms of instructional design. The findings of the present research suggest a number of implications for both CT instruction and CT assessment.

**Implications for CT Instruction.**

One of the takeaways of this study is that among all other variables, self-efficacy was the most significant variable that was related to CT quiz score and CT practices as measured by a performance-based test in both formal and informal contexts. This current research only showed a significant correlation between self-efficacy and CT. However, it did not provide information regarding causal relationships. Therefore, researchers could conduct an experimental study to show the causal relationships between students’ self-efficacy and CT. The finding in this study that self-efficacy significantly was related to CT practices in both of learning environments has the implication that special attention should be given to the CT instruction. Explicit instruction in problem solving has the potential for effective CT instruction. For example, Govender et al. (2014) found that teaching problem solving explicitly increased students’ self-efficacy in coding. Another effective strategy could be scaffolding (e.g., Repenning, Webb, & Ioannidou, 2015) with incremental challenges (e.g., Liu, Zhi, Hicks, & Barnes, 2017).

Future research could also develop a more fine-grained understanding of how students’ self-efficacy changes in a specific activity or from time to time during the intervention (Lishinski et al., 2016). Additionally, since self-efficacy and gender were found significantly related to students’ interest in this research, strategies that increase students’ interest in the early phase of the instruction could potentially be important for increasing students’ self-efficacy and closing the gender gap in CT practices and CS participation.

One more interesting findings emerged from the data analysis in this study was about gender differences. Female students were as good as male students in terms of CT practices in the informal learning environment, whereas female students in the formal classroom showed lower CT practices
than did male students. This might be due to the nature of the learning environment depending on voluntary versus involuntary participation because female students showed less interest than did male students in the formal context. In order to address this gender gap, I agree with the literature that curriculum should be relevant to students’ lives in order to get them interested in the subject (e.g., Margolis et al., 2011). Specifically, it was argued that female students have an affinity for communal goals and social causes (e.g., Diekman & Steinberg, 2013; Paulin, Ferguson, Schattke, & Jost, 2014). Thus, educators and researchers should take into consideration these studies and use other means to make content more relevant when designing CT-infused, CSEd activities.

One other implication of this study is about the relationship between students’ goal orientation and CT practices. Students whose goal orientation included understanding how to develop an app demonstrated more CT practices in their digital artifacts compared to other students. This suggests the potential for instructional design strategies that reinforce mastery goal orientation at the beginning of the intervention. One strategy could be to give students agency through collective goal setting (Bandura, 1977; 1997) because, as Bandura argued, participation in collective goal setting will not only help with students’ goal setting but will also increase their efficacy and performance.

To summarize, self-efficacy was found in this study to have the strongest positive relationship with middle-school-aged students’ CT. Goal orientation and interest also were found related to CT. Since these three variables stood out among other variables—such as prior experience with creative computing and age—self-regulated learning theory might be used in future work as a lens to understand the causal relationships between these three variables and CT. Instructional design strategies in self-regulated learning research can be used in future studies because self-regulated learning theory posits that students’ self-efficacy and goal setting are important predictors of students learning outcomes (Zimmerman, 2013).
CT Assessment.

One implication of this research is about CT assessment in an informal context. The findings on students’ CT quiz scores in an after-school setting showed that school-type assessments might not reveal participating students’ true knowledge and differentiate them from each other. Although the multiple-choice test is an easy method of assessment, it may not be valuable to the learner in the informal learning context. This is significant in that more typical formative assessments such as multiple choice quiz have the potential to mislead researchers regarding students’ CT skills. Future research should focus more on performance-based assessments than on multiple-choice tests.

Research can follow Brennan and Resnick’s (2012) suggestions with regard to six strategies for CT assessments: supporting further learning, incorporating artifacts, illuminating processes, checking in at multiple waypoints, valuing multiple ways of knowing, and including multiple viewpoints. They recommended that researchers could remix project analysis, artifact-based interviews, and design scenarios to create new forms of CT assessment.

Suggestions for Future Research

The goal of this study was to determine the nature of the relationships between middle-school-aged students’ CT and learner characteristics so that researchers and practitioners can design CT instructional approaches accordingly. Data analyses indicated that students’ self-efficacy had the strongest relationship with CT. However, since the main focus of the current study is correlation, it did not provide enough details to generalize that conclusion. Thus, the first step for future research is to replicate the basic structure of the current study, but with modifications that address its limitations as summarized above.

Since the findings of this research point to self-efficacy, interest, and goal orientation as learner characteristics that warrant further attention, the next step for future research is to conduct experimental studies with an instructional design that incorporates these characteristics at the beginning
of the intervention. The experimental design could include one condition having the current instruction design, while the other condition has the instructional design with strategies or activities that reinforce these three learner characteristics. The posttest scores of the two groups (under the two different experimental conditions) could be compared, using scores on a CT test in a formal classroom and CT practices in both informal and formal contexts.

The second potential avenue of a future study that could proceed from this study involves assessing students’ self-efficacy and interest periodically. This can also be done at the end of each activity. In this way, the activities that are helpful to increase students’ self-efficacy and interest can be identified. Examining the relationships between the activities and students’ self-efficacy and interest may also help the identification of the gender gap on CT practices.

Finally, future studies can employ self-regulated learning theory as a theoretical framework because self-regulated learning also underlines the importance of self-efficacy and goal orientation for students’ success in programming (Lishinski, Yadav, Good, & Enbody, 2016). Employing self-regulated learning theory will also help with finding a better instrument to assess self-efficacy, interest, and goal orientation because there are a variety of instruments for this well-known theory that can be found in the literature.

Summary

The findings of this study suggest that there is a significant relationship between learner characteristics and CT and that self-efficacy has the strongest relationship with CT.

The study’s findings have limitations in five main ways: research design, sample size, duration of the research, instrument, and data collection. I believe that an experimental design would help to understand what causes differences in CT depending on learner characteristics, a larger sample size would allow for greater precision in the model, a longer intervention could show more accurate results, a better self-efficacy measure would facilitate greater precision in assessment, and, finally,
collection of qualitative data would help to understand other factors, in addition to learner characteristics, which are related to CT.

This study has implications for better instructional design of CT and for improving CT assessment. Learner characteristics that were found related to CT need further investigation. Future research can expand on these relationships through experimental designs. The findings of this research also suggest that the multiple-choice assessment may not be appropriate for assessing CT in an informal learning context. Future research can explore the development of new assessment tools to examine students’ CT.
REFERENCES


APPENDICES

Appendix A: CT-App Quiz

Question 1
Look at the code below. What do you think happens once the score gets to 2?

- If the player touches the mole, the score will increase to 3.
- If the player touches the mole, the phone will vibrate.
- Game over procedure will be called.
- Update score procedure will be called.
- When `winner touched` will be called.
Question 2
Now let's look at this code block. What color do you think will appear if you click the `i_see_rainbows` button?

- Green
- Navy Blue
- Yellow
- Orange
- Purple

Question 3
If you want to create a variable named "Score", which combination of two blocks shown below would you use?

1) initialize global (name) to

2)

3)

4)

- 1 & 2
- 2 & 3
- 1 & 4
- 3 & 4
- 1 & 3
Question 4

What is the purpose of the code "get x" in the code block below?

- Getting a vertical point to draw circle on the canvas
- Getting a horizontal point to draw circle on the canvas
- Setting radius when the canvas touched
- Setting fill when the canvas touched
- Setting radius and fill

Question 5

You would like to make an elephant picture appear immediately after the elephant sound is completed. Where would you add your new code block to make that happen?

- a
- b
- c
- d
- None of the options are correct
Question 6

The code block above already has one song to play in it. If you wanted a different song to play before the song that's already there, where would you add that code?

- when HatButton . Click
  do set HatButton . Image to "rabbit.jpg"
  call Sound1 . Play
  call Sound2 . Play

- when HatButton . Click
  do set HatButton . Image to "rabbit.jpg"
  call Sound2 . Play
  call Sound1 . Play

- when HatButton . Click
  do set HatButton . Image to "rabbit.jpg"
  call Sound1 . Play

- when Button1 . Click
  do set Button1 . Image to "hat.jpg"

- when Button2 . Click
  do call Sound1 . Play
Question 7
What happens in the app below when the Button0 is clicked?

- Button0 will change to “5”
- Button0 will change to “True”
- Button0 will change to “False”
- Nothing will happen.
- It will give an error message.
Question 8

You realize that the code for your fitness level app (above) is broken. It should multiply height by 2 and then divide by 10. Which of the code blocks below would fix your app?
Question 9

You’d like someone using your app to click on ArtistButton2 and make the artist button song pause and another song play. Which of the blocks of code below would make that happen?

1. ```
   when ArtistButton2 .Click
   do
   call Freedom .Pause
   call Happy .Start
   ```

2. ```
   when ArtistButton2 .Click
   do
   call Happy .Pause
   call Freedom .Play
   ```

3. ```
   when ArtistButton2 .Click
   do
   call Freedom .Start
   ```

4. ```
   when Plc_button .Click
   do
   call Camera1 .TakePicture
   ```

5. ```
   when HatButton .Click
   do
   set HatButton .Image to “rabbit.jpg”
   call Sound2 .Play
   call Sound1 .Play
   ```
Question 10
The mole has hit the edge four times. What's the score?

initialize global score to 0

when Mole.EdgeReached
do set global score to get global score + 3

☐ 2
☐ 4
☐ 10
☐ 12
☐ 14
You want to build an app that gives health recommendations that are specific to an age range selected. You write the code above, but you find out that it doesn't have recommendations for the age range of 45 to 50. Of the options below, which one looks like it will fix your code?

- Change the second "if-then" block to the code below

- Change the second "if-then" block to the code below

- Add the code below into "when result clicked" code block

- Change the fourth "if-then" block to the code below
According to the code block above, what text is going to be displayed if your BMI is 28?

- You should eat more
- You should not eat that much
- Good Job. You should continue to eat
- You're overweight. Exercise and eat healthier
- Congratulations. You have the world's highest BMI.
Appendix B: Sample Lesson Plan

Tutorial: Calculator App Tutorial

This app help users solve basic math problems.

1. Open AppInventor.

1.1. Log into AppInventor and create a new project called Calculator.

1.2. Now you will be taken to the designer screen.

2. Build the Calculator Interface

   Interface means what the user works with and sees. When we say “user,” we mean a person who uses your app.

   2.1. From the Palette panel on the far left side of the screen, drag and drop a Label on your screen.

   2.2. In Components, rename the Label “TitleLabel.”

   2.3. In Properties, set the font size to 18.0.

   2.4. In Properties, set the Text property to “MyCalc.”
2.5. From the Palette panel on the far left of the screen, open the Layout drawer.

2.6. Drag and drop a **HorizontalArrangement** onto your screen below the “MyCalc” label.

![Image of screen showing HorizontalArrangement]

2.7. Open the User Interface drawer of the Palette, and drag a **textbox** into the HorizontalArrangement.

![Image of dragging a textbox into the HorizontalArrangement]

2.8. Select TextBox1 in the Components panel on the right side of the screen.

2.9. Go to the Properties panel on the far right side of the screen.

2.10. Remove the text from Hint.

2.11. Set its Width to 50 pixels.

2.12. Select the Numbers Only checkbox.

![Image of setting properties in the Properties panel]
2.13. Go back to the Palette panel, and make sure the User Interface drawer is open.


2.15. Place it to the right of the TextBox1.

2.16. Go to the Properties panel.

2.17. Set the Text property to + for the ListPicker. Your screen should look like the picture below.

2.18. Drag another TextBox into the HorizontalArrangement. Place this to the right of the ListPicker.
2.19. Go to the Properties panel on the far right side of the
2.20. Remove the text from Hint.
2.21. Set the Width to 50 pixels.
2.22. Select the NumbersOnly checkbox.

2.23. Drag-and-drop another Label to the right of the second TextBox.

2.24. In the Components panel, select the label and rename it EqualsLabel.
2.25. In the Properties panel, set the Text to =.
2.27. In Components, rename the label as ResultLabel.

2.28. Set the Text property to Result.

2.29. From the Palette panel, drag-and-drop a Button to the screen and place it below the HorizontalArrangement.

2.30. In Components, rename the button CalculateButton.

2.31. In Properties, set the Text property to Calculate.

You now have everything that you need from the Palette on your screen.
2.32. Under the Components panel, select ListPicker1.

2.33. In the Properties panel, look for the field called ElementsFromString. Add the following text: “+, -, x, /” (without the quotation marks).

2.34. Look for the field called Selection and type in the following text: “+”. This will make addition the default equation. If you want subtraction or a different equation to be the default, then you should change the text field to the appropriate mathematical symbol, like this for subtraction “-“.

3. Add Functionality to the Interface

The functionality of your calculator gets added by using the Blocks Editor. Along the top right corner of your App Inventor panel, click the button that says Blocks.

3.1. Go to the Blocks panel on the far left side of the screen and click ListPicker1.

3.2. Drag-and-drop the **when ListPicker1.AfterPicking do** block onto the board.
3.3. Click ListPicker1 in the Blocks panel again, then drag-and-drop the `set ListPicker1.Text` block into the `when ListPicker1.AfterPicking do` block. You’ll have to scroll to find the `set ListPicker1.Text` block.

![Diagram](image)

3.4. Click ListPicker1 in the Blocks panel again. Then, find the `ListPicker1.Selection` piece and connect it to the `set ListPicker1.Text` block.

![Diagram](image)

4. Add an if-then statement

4.1. From the Blocks tab, select CalculateButton and drag out the `when CalculateButton.Click do` block onto the board.

   **Click do** block onto the board.

   ![Diagram](image)

4.2. Open the Control drawer and drag the `if-then` block onto the board.

4.3. Insert the `if-then` block into the `when CalculateButton.Click` block.
4.4. From the Blocks tab, select ResultLabel and drag the `set ResultLabel.Text to` block onto the board.

4.5. Open the Math drawer and drag the `+` block onto the board.

4.6. Add this piece to the `set result label.Text to` piece.

4.7. From the Blocks tab, click TextBox1 and drag over the `TextBox1.Text` piece and then the `TextBox2.Text` piece.

4.8. Add the `TextBox1.Text` piece to the first gap in the `+` piece. Add the `TextBox2.Text` piece to the second gap in the `+` piece.

Connect the combined `Set ResultLabel.Text` and `+` blocks to the `then` part of the `if-then` block. This is the bottom half of the `if-then` block.
Open the Math drawer and drag an $=$ block to the board.

4.9. Open the Math drawer and drag an $=$ block to the board.

4.10. Connect the $=$ block to the test part of the if-then block.

4.11. Open the ListPicker drawer and drag the ListPicker1 .Selection block to the board.

4.12. Add this piece to the first empty space in the $=$ block.

4.13. Open the Text drawer and drag a text (" ") piece of the board.

4.14. Change the text in this block to " + " (without the quotation marks) by clicking the block and typing inside of it.
4.15. Add this block to the second empty space in the equal block.

4.16. Duplicate the if-then block. Right-click the block (Ctrl + Click on a Mac) and click Duplicate in the menu that appears.

4.17. Duplicate the if-then blocks two more times.

4.18. In the duplicated blocks, change one text piece to -, one to x, and one to / for each math function.
4.19. Change the math pieces from + to -, x, and / to match the text you just changed in the text pieces. You’ll have to replace the three + blocks from the duplicated if-then blocks with the appropriate blocks from the Math drawer.

4.20. Connect the three new blocks to the when CalculateButton.Click do block.
4.21. Your completed blocks should look like this:

5. Test your app
   5.1. Click Connect, then AI Companion.
   5.2. Turn on your smartphone and launch the AI Companion app.
   5.3. In the AI Companion app, tap on scan QR code.
   5.4. Scan the QR code on your computer screen, wait for a few seconds, and your app will appear on the phone screen.

6. Upload the App to AMAYS Website
   6.1. Save the app as a .aia file to your computer.
Appendix C: Syntax for Outlier Analysis

REGRESSION

/MISSING LISTWISE

/STATISTICS COEFF OUTS R ANOVA

/CRITERIA=PIN(.05) POUT(.10)

/NOORIGIN

/DEPENDENT CTquiz2nd11q

/METHOD=ENTER Confidence

/RESIDUALS HISTOGRAM(ZRESID) OUTLIERS(ZRESID LEVER COOK) ID(StudentID).