St. Catherine University
SOPHIA

Masters of Arts in Education Action Research Papers
Education

5-2018

Confidence and Common Challenges: The Effects of Teaching Computational Thinking to Students Ages 10-16

Chris Belanger
St. Catherine University

Hannah Christenson
St. Catherine University

Kathleen Lopac
St. Catherine University

Confidence and Common Challenges: The Effects of Teaching Computational Thinking to Students Ages 10-16

Submitted on May 15th, 2018

In fulfillment of final requirements for the MAED degree

Chris Belanger, Hannah Christenson, and Kathleen Lopac

St. Catherine University

St. Paul, Minnesota

Advisor: Heather Benedict Date 05-15-18

Abstract

This Action Research Project provides data from three different instructors teaching Computational Thinking (CT) to better understand the effects of CT instruction. The researchers focused on identifying problem-solving strategies used by students, what affect teaching CT has on student confidence and ability to problem solve, and what common challenges can be found at different age levels. The study used student pre and post-reflection to measure understanding and comfort with problem-solving. Researchers taught three common lessons of CT including the following concepts: algorithms, loops, conditional statements, and debugging. For data collection, each student was asked to work on a computer game called Human Resource Machine (HRM) while using video and audio to record themselves. Analysis showed a slight decrease in two categories related to working to find a solution to a difficult problem, and the ability to fix small problems that are part of a larger problem. There was a confidence increase in categories related to the ability to do math, the ability to give directions and the ability to someday build a computer. Two of the research sites were able to further break down the data to analyze the differences in the male vs. the female reflections. While CT is often seen as a separate subject, the analysis also showed that reading comprehension has a strong influence on students' ability to solve CT problems and should be taught in conjunction with CT to ensure students receive the maximum benefit.

Keywords: computational thinking, think-aloud problem solving, computer science

Data. Some may see that combination of four letters and immediately think of a quirky human-like android from a sci-fi TV show, others may think of the definition of the word data, others still may have flashbacks to cramming for a college statistics exam. There are many different ways those four letters can be interpreted and experienced, but what computer scientists already know, and the public-at-large is starting to realize, is that "data" is at the core of everything when it comes to technology and modern innovation (Wing, 2014). Data is collected through our phones by Google to help streamline our commute or to locate a misplaced device. Data is even analyzed by toilets to allow users to have a better idea of their health (Saenz, 2009). Internet service providers and social media websites collect the daily online footprints of their users to target advertisements to the most appropriate audience (Anthony, 2017). Stores collect data on the movement patterns of their shoppers to better optimize the layout of their merchandise (Siegel, 2014). Even governments and politicians have started learning to play the political game a little differently using data (Todd and Dann, 2017). Data is being generated and used in new and creative ways, on a scale never before seen in human history, and it's changing how the world works (Wing, 2014).

All of this data is requiring bigger and faster technology to make sense of it. New, complicated algorithms are being created by companies every day to analyze and act on data without any requirement of human intervention (Wing, 2014). Computer Science (CS) has traditionally been the field responsible for creating software and writing the code that controls much of the world's technology. However, a drop-off in enrollment of CS baccalaureate programs in the early 2000's coupled with the explosion of technology during that same time has created a situation where there are not enough skilled workers to meet the demands of companies wanting to make use of all that data.

The despair of the CS community in the early 2000's was brought on by declining enrollment and cessation of hiring in CS departments. Seeing this, Jeannette Wing (2006) penned her now well-known and often-cited editorial article, "Computational Thinking." It highlights the importance of teaching computational thinking (CT) skills, which are essentially the skills that underlie the field of CS, to all students as an attempt to reinvigorate the discipline and demonstrate the value of CS. It also emphasized that CT is not just for computing, but for all aspects of life. Wing argued that CT, "represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use." (Wing, 2006, p. 33) This article set off a cascading reaction causing many educators and government agencies to recognize the importance of fostering the problem-solving skills related to our modern world of digital technologies. These skills include, among others, creating algorithms (logical steps), decomposition (breaking down) of problems, loops (repetition), conditional statements (if/then/else decisions), and debugging (fixing) of problems (Yadav, Hong, & Stephenson, 2016; Grover & Pea, 2013).

As the demand for workers increases and the rate of data collection continues to march forward, many see teaching CT as a viable solution for preparing students to become part of the modern workforce, and for improving the current disparities of participation in STEM careers by women and minorities as compared to men (Grover & Pea, 2013). Others caution that unclear definitions, and a lack of understanding by the teachers who are supposed to be integrating it into their classrooms, should make us pause and perform a more in-depth analysis before jumping in with both feet (Denning, 2017). The sudden increase in the desire to teach CT has left little time for research before new curricula and standards are written and implemented. This hast has caused some confusion amongst educators about what exactly CT is, what the best integration strategies for already existing standards or subject areas are (Denning, 2017), and what practices and curriculum will facilitate the highest levels of success at varying developmental levels. With all of this information in mind, the purpose of this action research study is to explore the effects of CT instruction in students age 10-16. The study will include an analysis of common CT challenges, preferred problem-solving strategies, and the impact CT instruction has on student problem-solving confidence.

Review of the Literature

> "Computational thinking is a fundamental skill for everyone, not just computer scientists. To reading, writing, and arithmetics, we should add computational thinking to every child's analytical ability" (Wing, 2006, p. 33).

The discipline of computing as a mechanized manner of problem-solving has been around since the 1940's, and the idea of computational thinking since the 1980's (Denning, 2017). However, since Jeannette Wing (2006) published "Computational Thinking," it has become a new popular term for teaching the skills necessary to succeed in our modern technology-driven world, including computer coding. This increase in popularity is evidenced by the frequent mention of CT in peer-reviewed research and writing from the last ten years. The definition of CT is one that encompasses the skills necessary to become a computer programmer, but is also broad enough to apply to almost any activity in a person's day (Wing, 2006). This literature review will cover the history of the introduction of CT into the K-12 lexicon as well as some working definitions for CT. Next, it will examine CT's current place in curricula, including arguments for and against its further incorporation. Finally, it will outline some of the gaps that exist in the research surrounding CT, along with a discussion of some of the ways CT artifacts have been gathered, focusing on "think-aloud" protocols.

History and Definition of Computational Thinking

The term "computational thinking" was used as early as the 1980's by Seymour Papert in his writings about children and computers (Tendre & Denning, 2016). The more recent popularity of the term arose from the publication of the viewpoint "Computational Thinking" (Wing, 2006). Wing (2006) argues for the inclusion of CT in the education of all children and loosely defines it as problem-solving, system designing, and a range of mental tools related to program design. Other individuals and research groups have also offered definitions of CT. Yadav et al. (2016) share their definition:

> "The essence of computational thinking involves breaking down complex problems into more familiar/manageable sub-problems (problem decomposition), using a sequence of steps (algorithms) to solve problems, reviewing how the solution transfers to similar problems (abstraction), and finally determining if a computer can help us more efficiently solve those problems (automation)" (p. 565-566).

In Google's "Exploring Computational Thinking" (n.d.) education program for teachers, CT is defined as problem-solving that includes ordering and analyzing data, using ordered steps (algorithms), and dealing with open-ended problems.

Since the 1950's, researchers have advocated for the skill of thinking analytically to solve problems (Forsythe, 1959, as cited in Tendre & Denning, 2016). During the 1970's, clarifying the particular thinking skills required to make a computer function became one of the primary tools used to get CS recognized as a discipline unto itself and not just an extension of mathematics (Tendre & Denning, 2016). The rapid increase in the desire to teach CT extends the ongoing recognition of the importance of teaching the foundational concepts of decomposition, abstraction, algorithmic thinking, and the other thought processes linked with computer science (Tendre & Denning, 2016).

Publication of "Computational Thinking" (Wing, 2006) brought CT to a broader audience. "She catalyzed a discussion around computational thinking and mobilized resources to bring it into K–12 schools" (Denning, 2017, p. 35). Since 2006, many changes have occurred in the K-12 education realm related to the recognition of the importance of CT and computing education (Wing, 2016). In 2014, the UK made computing education part of their K-12 standards, and the BBC partnered with Microsoft to design and distribute small programmable devices, called micro:bit to every 11 and 12-year-old in the country and their teachers. In 2013, Code.org, a non-profit founded by Hadi Partovi and responsible for the creation of the "Hour of Code" was founded, and President Barack Obama's Computer Science for All Initiative allocated $4 billion in funding for CS education in U.S. schools (Wing, 2016). Additionally, block-based visual programming environments like Scratch and metaphorical environments such as Toontalk have fostered a renewed interest in introducing programming to K-12 students by making programming less abstract and adding a visual component (Lye & Koh, 2014).

Computational Thinking in Curriculum

There are many arguments for the inclusion of CT and computing education as skills for all students. One of the biggest is the current lack of a workforce capable of supporting a 21st-century economy (Grover & Pea, 2013). Others argue that every person needs to be able to be an intelligent *creator* of technology and not just a competent *user*. Our goal should be "moving students from merely being technology-literate to using computational tools to solve problems and represent knowledge" (Yadav et al., 2016, p. 568). Wing (2016) also views CT as cross-disciplinary, so even students who do not plan to enter a technology career can benefit from learning it.

Although there is a demand to create this 21st-century workforce and to develop students who create technology, not just consume it, there are still significant hurdles related to including CT in the curriculum. The inclusion of CT or CS as part of the state or district educational standards has yet to occur in most places (Zinth, 2016). In the years immediately following the publication of "Computational Thinking" (Wing, 2006) the number of schools offering computer science courses fell, and as of 2010, only nine states allowed CS to count as a math or science credit for graduation (Wilson, Sudol, Stephenson, & Stehlik, 2010). The number of states allowing a CS course to count towards graduation credits is increasing but still sits at only around half of the states depending on exactly how their policies are written (Zinth, 2016).

There is much work being done to get CT and CS effectively integrated into curriculum and standards, but there are some who warn against jumping on board with a concept that has yet to prove its many promises. Denning (2017) argues that CT lacks a solid definition and empirical evidence to support its many claims, and raises concern that although CT may prove to be as good as advertised, it also may not. In addition to a general lack of empirical evidence related to CT, there is evidence that teachers require more support than they are currently getting to be able to effectively implement new standards for teaching CT (Yadav et al., 2016). All of these points indicate that until there is more evidence, educators and policy-makers should proceed with caution when supporting CT initiatives.

Computational Thinking Research and Data Gathering

According to Yadav, Stephenson and Hong (2017), "Research on embedded computational thinking is starting to emerge" (p. 58). This is a critical point to understand because there is comparatively little research surrounding embedded CT as compared to many other educational practices. Yadav et al. (2017) go on to say, "[The research] has suggested that students exposed to computational thinking show significant improvement in their problem solving and critical thinking skills." Research on embedded CT is showing positive results, but many researchers still see much work to be done in this area (Grover & Pea, 2013). Additionally, Lye and Koh (2014) indicate that much of the research conducted on CT has taken place in afterschool or other voluntary activities which do not represent the populations found in mainstream K-12 classrooms. They suggest future studies be carried out in "naturalistic" classroom settings to ensure data can be generalizable. A better understanding of how students are thinking during the process of solving CT problems is also needed (Lye & Koh, 2014). This lack of understanding of the thinking process leads to the idea of using a think-aloud method to measure, or gain insight into, the thought processes of students as they are completing CT tasks.

While there is much excitement around the idea of teaching CT, currently, there seem to be more questions than answers related to best practice. Fessakis, Gouli, and Mavroudi (2013) articulated specific questions that could undergo a systematic investigation to greatly further the understanding of the impacts of CT in the classroom include:

- Which programming environments/tools are the most effective to use at each grade level?
- How can CT be effectively integrated into other curriculum areas besides computer coding?
- What should teachers be required to know before they are qualified to teach CT?

Over the last ten years, another area has gained popularity for teaching CT skills, and that is “unplugged” activities– activities that involve learning about CS concepts without the use of an electronic computer (Cortina, 2015). Advocates say these activities can lessen the digital divide because they do not require technology, allow for kinesthetic engagement, and stimulate CT thinking processes in a collaborative manner. Best practice research supports the importance of using collaborative instruction methods for girls in STEM, which further supports the use of unplugged activities in CT as part of a school curriculum (Twin Cities Public Television, 2013). Although one study conducted by Taub, Armoni, and Ben-Ari (2012) showed that using CS Unplugged did not make more students interested in following a technology career, Cortina (2015) asserts that the primary purpose of these activities is to teach thinking, not create a pipeline to tech careers.

Think-Aloud Problem Solving

Davidson, Deuser, and Sternberg (1994) described problem-solving as “the active process of trying to transform the initial state of a problem into the desired one” (p. 207–208). Additionally, Özcan, Imamoglu, and Katmer (2017) note that the problem-solving process is where one synthesizes existing knowledge then transfers it to a new problematic situation. The concept of metacognition directly ties into problem-solving. “The term metacognition literally means cognition about cognition, or more informally, thinking about thinking.” (Metacognition, 2009, n.p.) Pate and Miller (2011) assert that “Metacognition aids problem solving by helping an individual focus on identifying the problem, defining the problem space, generating a mental representation of the problem, planning how to proceed, and evaluating what is known about their own performance” (p. 108)

As part of CT instruction, studies have found that thinking aloud has helped students problem solve. Pate and Miller (2011) noted numerous studies that identify “self-explanation” as a means to improve student problem-solving performance when learning CT. “The think-aloud method is considered as one of the most effective ways to assess higher-level thinking processes” (Özcan et al., 2017, p. 131). Students who give reasons for their actions while problem-solving perform at a higher level than students who are silent or asked problem–focused questions (Pate & Miller, 2011). Quiet students could be working in a trial-and-error mode[1], instead of critically thinking about what they are doing. It is necessary for them to think aloud when solving problems such as programming and working on computational activities, to avoid trial-and-error strategies (Lye & Koh, 2014). “Thinking aloud during problem-solving means that the subject keeps talking, speaking out loud whatever thoughts come to mind while performing the task at hand” (Özcan et al., 2017, p. 131). Many techniques gather verbal data while giving interruptive suggestive prompts, breaking the subject’s concentration. However, thinking aloud allows one to give a concurrent account of their thoughts, avoiding interpretation and explanation (Özcan et al., 2017). With the help of technology such as recording audio or video, it is easier to collect

[1] Trial-and-error mode is a method of problem solving where the learner repeats varied attempts, and continues until they succeed or give up. An example of trial and error would be trying new medical drugs to see what is successful for a patient.

and analyze data, and easier for the subject to capture their immediate thoughts (Özcan et al., 2017).

Think-Aloud Partner Problem Solving (TAPPS) is one strategy found to be effective when problem-solving. TAPPS encourages students to verbalize during problem-solving to develop the student's ability to monitor their thoughts (Courgey, 1998, as cited in Pate & Miller, 2011). The procedure involves "a student solving a problem while a listener asks questions to prompt the student to verbalize their thoughts and clarify their thinking" (Lochhead, 2000, as cited in Pate & Miller, 2011, p. 109). One challenge is that if students feel they are poor problem solvers, they will not try to verbalize their confusion or monitor their thoughts, which may reduce the number of problems solved (Hacker, 1998, as cited in Pate & Miller, 2011).

There is a lack of research in the field of thinking-aloud problem-solving while completing CT activities. There are a few think-aloud studies in other areas, particularly in math. Basaraba, Zannou, Woods, and Ketterlin-Geller (2013) asked second through fourth-grade students to complete a concurrent think aloud while solving each math problem. Then, they asked a series of targeted questions following the assessment. When reviewing the existing literature, Özcan, et al. (2017) found few studies have examined thinking processes in solving mathematical problems by the think-aloud method, so they developed their study to investigate problem-solving methods of math students in the first years of middle school.

There is much positive feedback for think-aloud requirements, but some studies have also found thinking aloud while problem-solving is not beneficial. In 1975, Flaherty completed a mathematical problem-solving study in which 100 students were asked to complete problems using a think-aloud method. The groups that were not required to think aloud performed significantly better (Flaherty, 1975).

Literature Review Findings

Based on the literature review, a study was conducted to investigate the problem-solving processes students use while completing CT problems. To obtain more data about the actual process of problem-solving, not exclusively before and after data, students used a think-aloud method to explain their thinking (Lye & Koh, 2014; Pate & Miller, 2011; Özcan, et al., 2017). This research intended to answer these questions: What can teachers learn to improve instructional practices in CT by identifying common challenges for students aged 10-16 as they attempt to solve CT problems using a think-aloud process? In what ways does teaching CT affect students' confidence in their ability to solve problems? What strategies do students typically employ to solve CT problems?

Methodology

This study focused on how students responded to the teaching of CT by examining their personal feelings toward their skills and the subject, and teacher observations of student progress during and after instruction. Data included a student self-assessment of his or her problem-solving confidence, student and teacher observations during three common instruction periods on specific CT skills, and think-aloud video recordings created while students played the game Human Resource Machine (HRM).

Human Resource Machine is a leveled computer game that requires the player to use CT skills. The player uses blocks that perform a different function to complete the challenge presented by the “manager.” The basic idea is to make the character pick up the items from the “inbox” and put the correct items in the “outbox.” As the levels increase, new blocks are added to allow completion of more complex tasks. Players start by writing simple algorithms, then blocks are added that allow addition, subtraction, multiplication, loops, and conditional statements. In the higher levels, it is possible to create a solution that works for the data set immediately in front of the player, but not for other sets. In this case, the game does not allow the player to proceed until the solution works for all datasets. Screenshots from the game are in Appendix A.

The collected data was used to determine if, or how, teaching CT impacted student confidence while solving problems, if there were common challenges across the age groups, and which problem-solving strategies students most commonly used for addressing situations that required CT.

There were three separate settings for this action research study: Setting A) 5th grade STEM classes at a suburban elementary school in the midwest (N=107), Setting B) 7th and 8th grade industrial technology classes at a rural midwestern secondary school (N=15), and Setting C) 9th-11th grade honors algebra 2 classes at a private midwestern high school (N=54).

Due to large numbers at some locations, not all student data was used in the final analysis. Each setting had a different sample size and selection criteria. The final sample used in Setting A was comprised of 21 students, nine of whom were female and 12 of whom were male. This sample was selected by final video length, amount of time spent speaking out loud, the frequency of attendance, and gender distribution. Setting B was comprised of 15 students 7 of whom were female and 8 of whom were male. Due to the small number of students at this setting, all videos successfully submitted were part of the sample. The final sample for Setting C was comprised of 25 students, all of whom are female. This sample was selected using the length of the video and amount of time the student spent speaking out loud. The make-up of students in each class was also affected by whether the class was required or optional. The STEM and industrial technology courses are required for all students at the school. The algebra 2 course is a required math class at the high school. However, the honors section is optional for students who qualify.

The three teachers taught CT in their classes based on the amount of time they had available to meet with students. Students in Setting A received approximately 5 hours of instruction over a ten week period. Students in Setting B received approximately 8 hours of instruction over a nine-week period. Students in Setting C received approximately 6 hours of instruction over a nine-week period.

Students completed a pre and post-instruction reflection on their attitudes towards problem-solving, CT, and technology related topics. The full reflection can be viewed in Appendix B. In these reflections, students were asked to self-evaluate their problem-solving abilities, the use and manipulation of technology such as a tablet or computer, their knowledge in math and science classes, persistence in solving problems, ability to follow and give directions, and preferred problem-solving strategies. For the think-aloud data collection, students used the built-in cameras on Chromebooks, iPads, or Macbook computers, depending on the technology available at each setting, to capture their screen and record their voice. These recordings allowed researchers to monitor the student's thinking process while playing HRM. For this study, teacher help was limited during the recording of the think-aloud videos to clarifying questions about how the game functioned. During the three common CT instructional sessions, the three teachers taught the following concepts: algorithms, loops, conditional statements, and debugging. Additional lessons were taught, but due to the difference in available time and teaching methods used, there was no data collection for those lessons.

After all CT lessons were completed, the students were introduced to the game HRM. They were given brief instructions on how the game was operated and then were allowed to play each level. Students recorded their thought process as they played until either they were unable to pass a level or an hour of time had elapsed. Students submitted only their final video and then completed the post self-reflection on their attitudes towards problem-solving, CT, and technology related topics as well.

Data Analysis

The raw data for this study is in two forms. The first data collection method is a student reflection that was completed both before and after the CT instruction took place. The reflection includes ten Likert scale statements with a 1-5 rating. An answer of 1 indicates low agreement with a statement, and an answer of 5 indicates high agreement with a statement. The researchers also asked students to identify problem-solving strategies they commonly use by choosing from a list of pre-identified strategies. The list included an option for “other” where the student could record an answer not on the list.

The other main data source is the set of videos created by the students as they played HRM. The students were instructed to think out loud while they played so their thought processes and strategies could be coded and analyzed. At settings A and C, the sample sizes were reduced by choosing videos based on length, amount of talking, and attendance for all data collection sessions. The researchers then evaluated and documented their findings in an inventory of each video using the following four categories as shown in Appendix C: 1) Concepts/skills used or not understood: Online notes documented the CT or other skills each student appeared to understand. The documented skills included the following: reading comprehension, algorithms, logical reasoning, mathematical reasoning, loops, debugging, and conditional statements. 2) CT vocabulary used: Researchers recorded if any specific CT vocabulary was used. 3) Problem-solving strategies used: Researchers created established definitions for each of the pre-identified problem-solving strategies to increase alignment between video analyses. Researchers recorded if any of the following problem-solving strategies were used: decomposition, think aloud, drawing, writing, observation, teacher help, heuristic thinking, re-reading, prototyping, peer help, reapproach, trial and error, using hints, and other. 4) Attitude/demeanor while solving the problems: Researchers evaluated students attitudes during the problem-solving process according to the following descriptions extremely negative, negative, neutral, optimistic, and extremely optimistic. Students were rated on their attitudes throughout the submitted video.

Using the Likert scale numbers from the pre- and post-reflection data, the mean and median were found separately for each age level and each reflection statement to identify patterns in the data. There was a particular focus on statements that were rated highly overall (in mean, median or mode), and statements that had a large change from pre- to post-reflection. The pre- and post- reflections were also used to examine preferred problem-solving strategies and whether students used those preferred strategies to complete the problems during their think-aloud recordings. The strategies that students rated as their favorites were compared to what was observed in the video data to determine if students were making use of their preferred methods of problem-solving.

The findings from the student videos were also used to see if there were any trends in the concepts students understood and did not understand. Trends were determined by looking at each age level individually and all groups as a whole. Researchers noted which concepts students understood, along with which concepts they did not understand. Based on the raw numbers that were compared for each age group, it was inferred which topics needed more instruction or intervention, and which topics students showed mastery or understanding.

Reporting Findings

The purpose of this action research study was to explore the effects of CT instruction. The study included an analysis of common CT challenges, preferred problem-solving strategies, and the impact CT instruction has on student problem-solving confidence. The research design was qualitative with students self-reporting their feelings toward CT-related topics and researchers categorizing spoken responses from think-aloud recordings.

The selected subjects for this study were from three different schools: four 5th-grade STEM sections at a suburban Midwestern elementary school, one 7th and one 8th-grade industrial technology class at a rural Midwestern secondary school and three mixed grade, 9th through 11th-grade honors algebra 2 classes at a suburban high school. Total participation in the study was 175 students across the three settings. Selected from those 175 participants were 63 student data sets to be used for in-depth analysis. Table 1 shows the breakdown of the number of male and female students for each setting. During the pre-instruction reflection, three students were absent. During the post-instruction reflection, six students were absent.

Table 1
Demographics of Students

School/Grade Level	*Class*	*Males*	*Females*
Suburban 5th Grade	*STEM*	*12*	*9*
Rural 7th and 8th Grade	*Industrial Technology*	*8*	*7*
Suburban 9th-11th Grade	*Honors Algebra 2*	*0*	*25*

Confidence in Problem-Solving

The first research question addressed how CT instruction affected students' confidence in their ability to solve problems. To answer this question, students were asked to rate themselves using a Likert scale on various statements related to problem-solving, CT, and technology before the first CT lesson and following the last CT lesson. In addition to these self-evaluation statements, students chose up to three problem-solving strategies they use most often, including a blank space to add an option, these items are analyzed below in Table 4. The pre- and post-reflection were completed using an electronic form.

Table 2, shows the mean and median of how students rated themselves on each of the statements in both the pre and post reflections. Additionally, the table includes the p-value for the change in medians for each statement and age level. A majority of the data points stayed the same or had very little change from the pre to the post-reflection. A few places to note where the numbers did change from pre to post-reflection were the increase in the mean of the elementary and high school "I like to solve problems" statement compared to the decrease in the middle school mean. There was also an increase in the elementary school mean of "I am good at solving small problems...part of a bigger problem" statement compared with a decrease in both the middle school and high school. Confidence levels also seemingly decreased across all measures in elementary and middle school for the "When something is wrong I like working to find a solution" statement. Sample sizes were too small to have any statistical significance when using a t-test to determine p-value. However, it should be noted that the two statements with the lowest p-values overall are "I could be a computer programmer" and "I could design and build computers."

Table 2
Student Confidence in Problem Solving

Confidence Statements	*Age Level*	*Mean*			*Median*	
		Pre	*Post*	*P Value*	*Pre*	*Post*
I like to solve problems.	*Elementary (n=21)*	*3.8*	*3.9*	*.38*	*4*	*4*
	Middle (n=15)	*4.1*	*3.9*	*.38*	*4*	*4*
	High (n=25)	*4.0*	*4.1*	*.39*	*4*	*4*
I am good at using a computer or tablet/iPad.	*Elementary (n=21)*	*4.5*	*4.4*	*.40*	*5*	*4.5*
	Middle (n=15)	*4.4*	*4.1*	*.37*	*5*	*4*
	High (n=25)	*4.0*	*4.0*	*.39*	*4*	*4*
I am good at math.	*Elementary (n=21)*	*4.2*	*4.3*	*.40*	*4*	*4.5*
	Middle (n=15)	*4.0*	*4.1*	*.39*	*4*	*4*
	High (n=25)	*4.1*	*4.2*	*.42*	*4*	*4*
I am good at science.	*Elementary (n=21)*	*3.3*	*3.3*	*.37*	*3*	*3*
	Middle (n=15)	*3.4*	*3.6*	*.37*	*4*	*4*
	High (n=25)	*4.0*	*4.0*	*.41*	*4*	*4*
When something is wrong, I like working to find a solution.	*Elementary (n=21)*	*3.5*	*3.4*	*.37*	*4*	*3*
	Middle (n=15)	*3.7*	*3.4*	*.37*	*4*	*3*
	High (n=25)	*3.6*	*3.6*	*.40*	*4*	*4*
I am good at giving directions.	*Elementary (n=21)*	*3.6*	*3.7*	*.36*	*4*	*4*
	Middle (n=15)	*2.9*	*3.1*	*.37*	*3*	*3*
	High (n=25)	*3.4*	*3.6*	*.37*	*3*	*4*

I am good at following directions.	*Elementary (n=21)*	*4.2*	*4.2*	*.38*	*4*	*4*
	Middle (n=15)	*4.1*	*4*	*.41*	*4*	*4*
	High (n=25)	*4.0*	*4.2*	*.41*	*4*	*4*
I am good at finding small problems to fix that are part of solving a bigger problem.	*Elementary (n=21)*	*3.4*	*3.9*	*.37*	*3*	*4*
	Middle (n=15)	*3.6*	*3.4*	*.36*	*4*	*3*
	High (n=25)	*3.9*	*3.7*	*.41*	*4*	*4*
I could be a computer programmer.	*Elementary (n=21)*	*2.6*	*2.8*	*.35*	*3*	*3*
	Middle (n=15)	*2.7*	*2.6*	*.34*	*2.5*	*2*
	High (n=25)	*2.8*	*3.0*	*.36*	*3*	*3*
I could design and build computers.	*Elementary (n=21)*	*2.2*	*2.7*	*.34*	*2*	*3*
	Middle (n=15)	*2.1*	*2.4*	*.33*	*2*	*2*
	High (n=25)	*2.2*	*2.6*	*.35*	*2*	*3*

Table 3 shows the mean and median of how students rated themselves in the pre and post reflection in addition to the p-value for each pre and post-median set. The table also separates the data of males and females within the elementary STEM class and the middle school industrial technology class. One change between pre and post is with the mode value in the statement "When something is wrong … even if it takes a long time." Here, each value decreased for all genders and both grade levels. Another change would be in the statement "I am good at finding small ... solving a bigger problem." The elementary mode increased while the middle school mode decreased. In the "I could be a computer programmer" statement, there was a decrease in the middle school females data by a value of two. Lastly, the elementary confidence for both males and females increased in the "I could design and build computers" statement. While the sample sizes were too small to hold statistical significance, it should be noted that the p values are lowest overall for this statement as well as the statement “I could be a computer programmer.”

Table 3

Student Confidence in Problem Solving- Male vs Female

Confidence Statements	*Age and Gender*	*Mean*			*Median*	
		Pre	*Post*	*P Value*	*Pre*	*Post*
I like to solve problems.	*Elementary Female (n=9)*	*3.3*	*3.6*	*.31*	*3*	*3.5*
	Elementary Male (n=12)	*4.1*	*4.2*	*.36*	*4*	*4*
	Middle Female (n=7)	*3.7*	*3.7*	*.30*	*4*	*4*
	Middle Male (n=8)	*4.4*	*4.1*	*.38*	*4*	*4*
I am good at using a computer or tablet/iPad.	*Elementary Female (n=9)*	*4.2*	*4.5*	*.37*	*4*	*4.5*
	Elementary Male (n=12)	*4.7*	*4.3*	*.36*	*5*	*4.5*
	Middle Female (n=7)	*4.9*	*4.3*	*.38*	*5*	*4*
	Middle Male (n=8)	*3.9*	*4.0*	*.31*	*4*	*4*
I am good at math.	*Elementary Female (n=9)*	*4.2*	*4.0*	*.32*	*4*	*4*
	Elementary Male (n=12)	*4.3*	*4.6*	*.39*	*4.5*	*5*
	Middle Female (n=7)	*3.9*	*4.0*	*.30*	*4*	*4*
	Middle Male (n=8)	*4.1*	*4.3*	*.39*	*4*	*4*
I am good at science.	*Elementary Female (n=9)*	*3.4*	*3.4*	*.33*	*3*	*3.5*
	Elementary Male (n=12)	*3.3*	*3.3*	*.32*	*3.5*	*3*
	Middle Female (n=7)	*3.1*	*3.0*	*.33*	*3*	*3*
	Middle Male (n=8)	*3.7*	*4.3*	*.33*	*4*	*4*
When something is wrong, I like working to find a solution, even if it takes a long time.	*Elementary Female (n=9)*	*3.6*	*3.3*	*.33*	*4*	*3*
	Elementary Male (n=12)	*3.5*	*3.7*	*.31*	*3.5*	*4*
	Middle Female (n=7)	*3.4*	*3.0*	*.31*	*3*	*3*
	Middle Male (n=8)	*4.0*	*3.9*	*.37*	*4*	*4*

I am good at giving directions.	*Elementary Female (n=9)*	*3.8*	*3.8*	*.33*	*5*	*4*
	Elementary Male (n=12)	*3.5*	*3.6*	*.30*	*3.5*	*4*
	Middle Female (n=7)	*2.9*	*2.9*	*.31*	*3*	*3*
	Middle Male (n=8)	*3.0*	*3.4*	*.34*	*3*	*4*
I am good at following directions.	*Elementary Female (n=9)*	*4.8*	*4.5*	*.41*	*5*	*4.5*
	Elementary Male (n=12)	*3.8*	*4.0*	*.37*	*4*	*4*
	Middle Female (n=7)	*4.0*	*3.9*	*.35*	*4*	*4*
	Middle Male (n=8)	*4.3*	*4.1*	*.42*	*4*	*4*
I am good at finding small problems to fix that are part of solving a bigger problem.	*Elementary Female (n=9)*	*3.3*	*3.5*	*.34*	*3*	*3.5*
	Elementary Male (n=12)	*3.5*	*4.2*	*.37*	*4*	*4.5*
	Middle Female (n=7)	*3.1*	*3.1*	*.28*	*4*	*3*
	Middle Male (n=8)	*4.1*	*3.7*	*.37*	*4*	*4*
I could be a computer programmer.	*Elementary Female (n=9)*	*2.4*	*2.6*	*.35*	*2*	*3*
	Elementary Male (n=12)	*2.7*	*3.0*	*.27*	*3*	*3*
	Middle Female (n=7)	*2.6*	*2.0*	*.29*	*3*	*2*
	Middle Male (n=8)	*2.9*	*3.1*	*.28*	*2*	*3*
I could design and build computers.	*Elementary Female (n=9)*	*2.1*	*2.5*	*.30*	*2*	*3*
	Elementary Male (n=12)	*2.3*	*2.9*	*.25*	*2*	*2.5*
	Middle Female (n=7)	*1.7*	*1.9*	*.29*	*1*	*1*
	Middle Male (n=8)	*2.4*	*2.9*	*.29*	*2*	*2*

Common Challenges

The next research question focused on what common challenges if any, students ages 10-16 encountered when solving CT problems. To answer this question, researchers had students play HRM. The students were asked to use the think-aloud problem-solving strategy as they played each level to capture their thought process along with what they were doing on the screen. Students continued to play until either they were unable to pass the level, or until an hour of time had elapsed. The students submitted their final recorded video, and researchers analyzed each video in the following areas: concepts/skills understood in the problem-solving process, concept/skills student appeared not to understand, CT vocabulary used, problem-solving strategies used, and student attitude or demeanor. Also, a field to record any additional notes was included.

Not every level of the HRM game required students to use all of these skills listed in Table 3. For example, some levels were more a test of logic skills and not mathematical skills. Also, there were situations where the student appeared to understand the definition of a concept, such as a "loop" but was not able to effectively create a solution using that concept. In that case, the student was considered to "not understand" that concept.

Table 3
Concepts Students Appeared to Not Understand

Concept/Skill	*Elementary n=21*	*Middle n=15*	*High n=25*
Reading Comprehension	*10*	*4*	*0*
Algorithm	*11*	*6*	*0*
Logical Reasoning	*6*	*5*	*10*
Mathematical	*2*	*1*	*4*
Loops	*10*	*2*	*24*
Debugging	*2*	*3*	*0*
Conditional Statements	*NA*	*3*	*23*

Problem-Solving Strategies

The third research question focused on which strategies students used to solve a CT problem in HRM. Several common problem-solving strategies were pre-identified, including decomposition, think-aloud, drawing, writing, observation, teacher help, heuristic thinking, re-reading, prototyping, peer help, reapproach, trial and error, and using hints. The researchers used a check-box method to record the use of any of the pre-identified strategies. There was also an "other" option to identify a strategy not previously on the list.

The data included in Table 4 is from the student pre and post self-reflections. The students were allowed to choose up to three problem-solving strategies they use most often. Students could also include a strategy they prefer that was not on the list by recording it in the "other" option. Although the students were explicitly asked to use the think-aloud strategy for this study, it was not a preferred strategy in the pre self-reflection, and gaining only a few extra votes in the post-reflection. The most preferred strategies were drawing or writing notes, thinking silently, or working with a friend. Many students indicated they preferred to ask a teacher for help, but during the recording of the think-aloud videos, teachers were limited to answering clarifying questions about how the game functioned. Students were encouraged to refrain from asking for assistance from the teacher about how to solve the problem while they were playing HRM for the think-aloud recording. No other problem-solving strategies were restricted.

Table 4
Student Preferred Problem Solving Strategies

Strategy	*Elementary n=21*		*Middle n=15*		*High n=25*	
	Pre	*Post*	*Pre*	*Post*	*Pre*	*Post*
Draw or write notes	*14*	*10*	*4*	*2*	*12*	*19*
Think silently about the problem	*13*	*6*	*3*	*7*	*11*	*15*
Think out loud about the problem	*1*	*3*	*0*	*0*	*2*	*5*
Work with a friend	*15*	*10*	*7*	*4*	*6*	*12*
Ask a teacher	*5*	*4*	*1*	*1*	*5*	*13*
Wild guess	*1*	*1*	*0*	*0*	*0*	*3*
Give up	*1*	*1*	*0*	*0*	*0*	*2*
Other	*1*	*3*	*0*	*1*	*0*	*1*

Table 5 displays the problem-solving strategies used in the student videos. As previously noted, the students were explicitly directed to use the think-aloud process so that the researchers could get a better idea of how each student was thinking about the problem. Even with this direction, some students talked very little or not at all, and sometimes the recorded talking gave no insight into the student's thought-process and was therefore considered not to be using the think-aloud strategy. Outside of the requested think-aloud process, trial and error was the most commonly used strategy across all levels.

Table 5
Problem Solving Strategies Used

Strategy	*Elementary n=21*	*Middle n=15*	*High n=25*
Decomposition	*0*	*3*	*3*
Think Aloud	*13*	*9*	*25*
Drawing	*0*	*0*	*0*
Writing	*0*	*0*	*0*
Teacher Help	*2*	*2*	*0*
Heuristic Thinking	*9*	*6*	*12*
Re-reading	*6*	*7*	*24*
Prototyping	*5*	*2*	*25*
Peer Help	*4*	*1*	*1*
Reapproach	*7*	*4*	*18*
Trial and Error	*11*	*14*	*25*
Using Hints	*5*	*4*	*16*
Other	*0*	*0*	*0*

Conclusions

This research aimed to learn more about teaching CT, the effects of CT instruction on students' problem-solving confidence, and to observe any common challenges students face when using CT to solve problems. Over the course of the study, data was collected through pre-instruction self-reflection, think-aloud student video analysis, and post-instruction self-reflection.

Analysis of the pre- and post-instruction reflections indicated an increase in student confidence related to math, direction giving, and the possibility of designing or building computers. Since there are many commonalities between mathematical thinking and CT, and HRM has some levels that incorporate math skills, this was an anticipated increase. However, between the pre- and post-instruction reflections, there was an overall decrease in student confidence in two categories: solution finding even if it takes a long time and fixing small problems that are part of a bigger problem. This could be because the challenge of HRM caused frustration for some, so they felt they were not good at problem-solving.

Following the analysis of the think-aloud videos, there are indications of overlaps in some of the challenges to the different grade levels, but also some differences in the grade levels as they increased. The elementary school data showed a need for focused instruction in using reading comprehension skills while solving CT problems. Another identified need was for more instruction on constructing algorithms and using loop functions. This data would seem to indicate that students should have a solid foundation in understanding what the problems are asking them to do and how to construct a solution (algorithm) before moving on to the more complex pieces of CT such as loops or conditional statements.

The middle school data also showed they could benefit from focused reading comprehension instruction related to CT problems and more guidance on algorithm construction. Additionally, the middle school data showed that using logical reasoning to find a solution was difficult. The high school data showed no problems with reading comprehension, but similar to the middle school data, the students might benefit from more focus on logical reasoning in CT instruction and with using loop functions and conditional statements.

As was previously mentioned, not all of the levels of HRM required the use of all of the skills listed in Table 3. Since some of the younger students did not reach levels with loops or conditional statements included, and many of the levels did not have a direct mathematical thinking requirement, mathematical thinking did not show up as a challenge as often as it might have if all of the levels required mathematical thinking. Therefore this data cannot be used to conclude that teaching CT results in immediate student improvement in regards to mathematical reasoning.

Another example of the difference in skills that was required is that the data in Table 3 seems to indicate the high school students displayed an overwhelming misunderstanding of loops and conditional statements while the elementary students did not. This makes it seem as though the elementary students understood the concept and the high school students did not. However, this can be explained by the fact that the levels reached by the elementary students did not require the use of these skills as often as the high school students.

A point of clarification for the data collected about the high school student videos should also be noted. As the students reached higher levels, they tended to focus on the set of numbers that was immediately in front of them instead of creating broad solutions. Multiple videos showed students were able to create a solution that worked with one specific set of numbers, but that solution would not work in all cases. Due to this being an unforeseen situation, and lacking a better way for it to be coded, the videos were recorded as showing the ineffective creation of loops or conditional statements, and as showing the use of heuristic thinking.

Recommendations

Based on the findings of this study it is recommended that CT in upper elementary school be taught in conjunction with reading comprehension strategies to ensure students understand what the problem is asking them to do. The teaching of CT should also have more emphasis on how to create an ordered set of instructions (an algorithm) to solve a problem and not just as a list of steps. Students in middle school could also benefit from attention to reading strategies and algorithms, but they should be given more support in how to logically work through a problem. There is a need for more instruction for high school students in the application of loops and conditional statements, so they can create solutions, not just restate definitions. It would be interesting to reanalyze the data to develop a more precise breakdown of what caused students to struggle with creating proper loops and conditional statements when during regular classroom instruction they seemed to understand them. From this study, researchers were not able to determine why students struggled to create broad solutions (solutions that work for all data sets, not just the set in front of them). More study should be done to determine how teaching practice can be changed to help students with critical thinking or analytical problem-solving skills that encourages their creation of broad solutions.

As was addressed previously, CT is a newer curriculum field that has very quickly begun to be adopted into classrooms. This quick adoption has caused implementation of curricula and standards with limited background research into “what works” at different developmental levels for CT education and integration into the K-12 curriculum. For that reason, this study identifies issues that need further exploration for CT integration in K-12 settings. As novices in computational thinking, the researchers were unable to anticipate all the issues that would arise before implementation. Since completion of this, it has become obvious to all parties involved that much more time was needed to create more than just a passing familiarity with the subjects taught. The amount of time that would be required to create deep understanding and mastery is unknown other than it needs to be more than what this study was able to provide.

Two additional areas of consideration were discovered by the research team. First, the majority of students across all three settings indicated they liked writing things down or drawing pictures as a means of solving problems. However, in the practice of creating the think-aloud videos only one participant made any indication of a desire to write down her thoughts, and none of the participants made use of the writing or drawing strategy. Writing or drawing was never specifically prohibited by the participating teachers so this outcome raises the following questions: a) What was the primary inhibitor to the students using a problem-solving technique they indicated they preferred? b) Would using the writing or drawing method along with the think-aloud have any impact on the student outcomes if they did use it? And c) Would students use the writing or drawing strategy if they were completing an “unplugged” CT activity as opposed to one that is computer-based? If one of the primary goals of teaching CT is to get students to be better problem-solvers, then the instruction should focus on how the students best solve problems and not just on the technology that is attached to the problems.

Second, in some measures of student confidence, the mean, median, and mode markers decreased from the pre-instruction survey to the post-instruction survey. If another primary goal of teaching CT is to increase student confidence in problem-solving, why did the confidence decrease from the beginning to the end of the teaching sessions? If there is a possibility that inadequately teaching CT lowers student confidence then it would be important to ask three questions: a) Does teaching CT even have a positive impact on student problem-solving confidence? And b) If teaching CT does have a positive impact on students' confidence in problem-solving, what amount of CT instruction is required to achieve that positive impact? c) Knowing that the experience of "failure" in the iterative process of problem-solving that happens across STEM has been identified as an obstacle to young women's pursuit of STEM in post-secondary contexts, how could the process of "failing fast" be celebrated or differently taught to change the way females internalize this experience? (Dweck & Simmons, 2014).

This action research study was completed to encourage educators to pursue CT instruction and continue researching best practices. The results of this study will be used to enhance the CT instruction of the researchers and encourage CT program development at the participating schools. In the researchers' classrooms, CT instruction will be modified to better fit the needs of the specific age group of students by putting more focus on the topics in which the most challenges were encountered and observing the impact of encouraging them to use more of their preferred problem-solving strategies. Finally, it is hoped that the findings from this research will also be used to frame future, more robust studies into the efficacy of teaching CT skills.

References

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal Of Educational Technology & Society*, *19*(3), 47-57. Retrieved from http://www.jstor.org/stable/jeductechsoci.19.3.47

Anthony, A. (2017, May 21) *The great digital age swindle...and the man fighting back.* Retrieved from https://www.theguardian.com/technology/2017/may/21/jonathan-taplin-interview-move-fast-break-things-facebook-google-amazon-dylan-scorsese

Armour, P. G. (2013). When faster is slower. *Communications of the ACM, 56*(10), 30. doi:10.1145/2505338

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

Basaraba, D., Zannou, Y., Woods, D., Ketterlin-Geller, L., & Society for Research on Educational Effectiveness. (2013, September). *Exploring the utility of student-think alouds for providing insights into students' metacognitive and problem-solving processes during assessment development*. Presented at the meeting of the Society for Research on Educational Effectiveness, Washington, D.C.

Cortina, T. J. (2015). Reaching a broader population of students through “unplugged” activities. *Communications Of The ACM*, *58*(3), 25-27. doi:10.1145/2723671

Davidson, E., Deuser, R., & Sternberg, J. (1994). *The role of metacognition in problem solving.* In J. Metcalfe & A. P. Shimamura (Eds), Metacognition: Knowing about knowing (pp.207-226). Cambridge, MA: MIT Press.

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM, 60*(6), 33-39. doi:10.1145/2998438

Depryck, K. (2016). From computational thinking to coding and back. *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM 16*. doi:10.1145/3012430.3012492

Dweck, C. & Simmons, R. (2014) *Why do women fail?* Retrieved from http://www.cnn.com/2014/07/29/opinion/dweck-simmons-girls-confidence-failure/index.html

Ericsson, K. A. (n.d.). Protocol analysis and expert thought: Concurrent verbalizations of thinking during experts performance on representative tasks. *The Cambridge Handbook of Expertise and Expert Performance,* 223-242. doi:10.1017/cbo9780511816796.013

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97. doi:10.1016/j.compedu.2012.11.016

Flaherty, E. (1975). The thinking aloud technique and problem solving ability. *The Journal of Educational Research, 68*(6), 223-225. doi:10.1080/00220671.1975.10884753

Google. (n.d.). *Exploring Computational Thinking*. Retrieved June 23, 2017, from https://edu.google.com/resources/programs/exploring-computational-thinking/

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43. doi:10.3102/0013189x12463051

Kelley, T. R., Capobianco, B. M., & Kaluf, K. J. (2014). Concurrent think-aloud protocols to assess elementary design students. *International Journal of Technology and Design Education, 25*(4), 521-540. doi:10.1007/s10798-014-9291-y

Lye, S. Y., & Koh, J. H. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51-61. doi:10.1016/j.chb.2014.09.012

Metacognition. (2009). In E. M. Anderman & L. H. Anderman (Eds.), *Psychology of Classroom Learning*: *An Encyclopedia* (Vol. 2, pp. 604-608). Detroit: Macmillan Reference USA. Retrieved from http://go.galegroup.com.pearl.stkate.edu/ps/i.do?p=GVRL&sw=w&u=clic_stkate&v=2.1&it=r&id=GALE%7CCX3027800175&sid=exlibris&asid=c0361ed7282f57d41adcb4031cf705cc

Pate, M., & Miller, G. (2011). A descriptive interpretive analysis of students oral verbalization during the use of think-aloud pair problem solving while troubleshooting. *Journal of Agricultural Education*, 52(1), 107-119. doi:10.5032/jae.2011.01107

Özcan, Z. Ç., Imamoglu, Y., & Katmer Bayrakli, V. (2017). Analysis of sixth grade students' think-aloud processes while solving a non-routine mathematical problem. *Educational Sciences: Theory And Practice*, *17*(1), 129-144.

Payne, J. W. (1994). Thinking aloud: Insights into information processing. *Psychological Science,5*(5), 241-248. doi:10.1111/j.1467-9280.1994.tb00620.x.

Saenz, A. (2009, May 12) *Smart toilets: doctors in your bathroom.* Retrieved from https://singularityhub.com/2009/05/12/smart-toilets-doctors-in-your-bathroom/#sm.001kobl5b153nf03q0u1b0gpepl7f

Siegel,R. (Producer).(2014). *All Things Considered*[Radio Series]. Washington, DC: NPR. Retrieved from https://www.npr.org/2014/06/16/322597862/how-retailers-use-smartphones-to-track-shoppers-in-the-store

Solomon, G. (2015). Hours of code: Past & present. *Tech & Learning, 36*(3), 28-31.

Taub, R., Armoni, M., & Ben-Ari, M. (2012). CS Unplugged and Middle-School Students' Views, Attitudes, and Intentions Regarding CS. *ACM Transactions on Computing Education, 12*(2), 1-29. doi:10.1145/2160547.2160551

Tedre, M., & Denning, P. J. (2016). The long quest for computational thinking. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research - Koli Calling 16*. doi:10.1145/2999541.2999542

Todd, C., & Dann, C., (2017, March 14) How Big Data Broke American Politics. Retreived from https://www.nbcnews.com/politics/elections/how-big-data-broke-american-politics-n732901

Twin Cities Public Television. (2013). SciGirls Seven: How to engage girls in STEM. St. Paul, MN: Author. Retrieved from http://www.scigirlsconnect.org/scigirls.

Wilson, C., Sudol, L., Stephenson, C. and Stehlik, M. (2010), Running on empty: The failure to teach K–12 computer science in the digital age, ACM, available at: www.acm.org/runningonempty/fullreport.pdf

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33. doi:10.1145/1118178.1118215

Wing, J. M. (2014). Computational thinking benefits society. Retrieved from http://socialissues.cs.toronto.edu/index.html%3Fp=279.html

Wing, J. M., & Stanzione, D. (2016). Progress in computational thinking, and expanding the HPC community. *Communications of the ACM,59*(7), 10-11. doi:10.1145/2933410

Yadav, A., Gretter, S., Good, J., & Mclean, T. (2017). Computational thinking in teacher education. *Emerging Research, Practice, and Policy on Computational Thinking,* 205-220. doi:10.1007/978-3-319-52691-1_13

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, 60(6), 565-568. doi:10.1007/s11528-016-0087-7

Yadav, A., Stephenson, C. & Hong, H. (2017) Computational thinking for teacher education. Communications of the ACM, 60(4), 55-62.

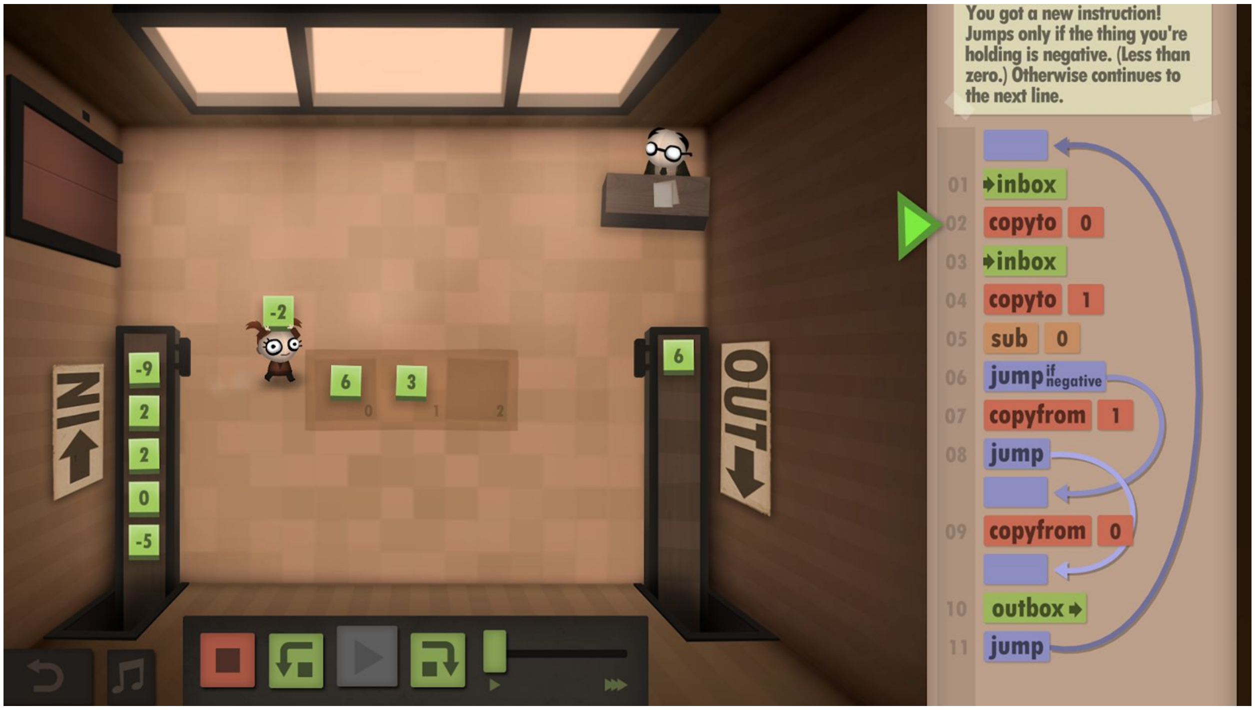
Zhong, B., Wang, Q., Chen, J., & Li, Y. (2015). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research, 53*(4), 562-590. doi:10.1177/0735633115608444

Zinth, J. (2016, September). Computer Science in High School Graduation Requirements. *Education Trends.* Retrieved from the Education Commission of the States website: https://www.ecs.org/

Appendix A

Screenshots from HRM

Used with permission from Tomorrow Corporation.


You got a new instruction! Jumps only if the thing you're holding is negative. (Less than zero.) Otherwise continues to the next line.
IN
OUT
01 inbox
02 copyto 0
03 inbox
04 copyto 1
05 sub 0
06 jump if negative
07 copyfrom 1
08 jump
09 copyfrom 0
10 outbox
11 jump


SUCCESS
Size Challenge
Use 11 or fewer instructions.
✓ Your current solution uses 11 instructions.
11 11
Efficiency Challenge
Complete in 37 or fewer steps.
– Your solution completes in 38 steps, on average.
37 38
*These challenges can be very difficult! And in many cases, not possible to optimize both simultaneously with one solution.
back
continue to elevator

Appendix B

Reflection Statements

Although the picture indicates it is the pre-reflection, the same statements were used for both the pre- and post-reflection.

Computational Thinking Pre-Reflection

* Required

1. **Student study number** *

2. **Gender** *
Mark only one oval.
   - Female
   - Male

3. **Grade** *

4. **I like to solve problems.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

5. **I am good at using a computer or tablet/iPad.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

6. **I am good at math.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

7. **I am good at science.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

8. **When something is wrong, I like working to find a solution, even if it takes a long time.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

9. **I am good at giving directions.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

10. **I am good at following directions** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

11. **I am good at finding small problems to fix that are part of solving a bigger problem.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

12. **I could be a computer programmer.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

13. **I could design and build computers.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

14. **Choose up to three problem solving strategies you use MOST OFTEN. You may add in an answer if one of your favorites is not listed.** *

Check all that apply.

- [ ] Draw or write notes
- [ ] Think silently about the problem
- [ ] Think out loud about the problem
- [ ] Work with a friend
- [ ] Ask a teacher
- [ ] Wild guess
- [ ] Give up
- [ ] Other: ______________________

Appendix C

CT Video Analysis Form

This form was used to record notes while viewing student problem-solving videos.

CT Video Analysis

Use this tool to record findings on videos from the CT problem solving activity.

* Required

1. **Reviewer** *
   Mark only one oval.
   - Christopher Belanger
   - Hannah Christensen
   - Kathy Lopac

2. **Source content being reviewed from:** *
   Mark only one oval.
   - Chris
   - Hannah
   - Kathy

3. **Video Identification** *

   ____

4. **Concepts/Skills Understood in Problem Solving Process**
   Check all that apply.
   - [ ] Reading Comprehension
   - [ ] Algorithm
   - [ ] Logical Reasoning
   - [ ] Mathematical
   - [ ] Loops
   - [ ] Debugging
   - [ ] Conditional Statements

5. **Concept Skill Student Appeared to Not Understand**
   Check all that apply.
   - [ ] Reading Comprehension
   - [ ] Algorithm
   - [ ] Logical Reasoning
   - [ ] Mathematical
   - [ ] Loops
   - [ ] Debugging
   - [ ] Conditional Statements

6. **CT Vocabulary Used**

7. **Problem Solving Strategies Used**
Check all that apply.

- [ ] decomposition
- [ ] think aloud
- [ ] drawing
- [ ] writing
- [ ] observation
- [ ] teacher help
- [ ] heuristic thinking
- [ ] re-reading
- [ ] prototyping
- [ ] peer help
- [ ] reapproach
- [ ] trial and error
- [ ] using hints
- [ ] Other:

8. **Attitude Demeanor** *
Mark only one oval.

- ( ) Extremely Negative
- ( ) Negative
- ( ) Neutral
- ( ) Optimistic
- ( ) Extremely Optimistic

9. **Additional Notes**