IJCSES

Volume 4, No: 2
October 2020
ISSN 2513-8359

International Journal of Computer Science Education In Schools

Editors

Dr Filiz Kalelioglu

Dr Yasemin Allsop

www.ijcses.org

International Journal of Computer Science Education in Schools

October 2020, Vol 4, No 2

DOI: DOI 10.21585/ijcses.v4i2

Table of Contents

Incorporating Cyber Principles into Middle and High School Curriculum

Jessica Ivy
Robert Kelley
Kristin Cook
Kevin Thomas

jivy@bellarmine.edu
rkelley@bellarmine.edu
kcook@bellarmine.edu
kthomas@bellarmine.edu

DOI: https://doi.org/10.21585/ijcses.v4i2.101

Abstract

Although many practicing teachers have not experienced teacher preparation programs that teach cyber security (Pusely & Sadera, 2011) or are familiar with cyber principles (Authors), embedding these ideas into instruction in a variety of content areas is essential for promoting cyber literacy and citizenship. This study explores a professional development program that provided middle and high school teachers across disciplines with opportunities to explore, first as learners and then as educators, cyber citizenship and programming concepts with explicit connections to the cybersecurity principles and concepts. Participating teachers experienced inquiry-based learning, focused classroom discourse, and collaborative learning that centered on GenCyber Cybersecurity First Principles and GenCyber Cybersecurity Concepts (GenCyber, 2019). Results indicated the professional development enabled teachers to iteratively reflect on best practices in cyber education while learning and applying the content of GenCyber Principles within the context of their own field of study.

Keywords: cyber security, professional development, cyber citizens, cyber literacy, high school

1. Introduction

The National Science Foundation describes cybersecurity as one of the defining issues of our time (NSF, n. d.). Education is not immune to this threat. According to a report by the K-12 Cybersecurity Resource Center (Levin, 2020), 2019 saw a dramatic rise in cyberattacks on K-12 schools, including denial of service, phishing, ransomware, and unauthorized disclosure/breach. Students and staff were responsible for 9% of the data breaches, which can result in unauthorized access and use of students' personal information (e.g., social security numbers, school records, etc.) (REMS, 2017). In fact, student data was included in more than 60 percent of K–12 data breaches in 2018 (Levin, 2020). In addition to data breaches, cyber threats to students also include cyber-bullying, -predators, and -stalking (Bamford, 2005; Kessel Schneider, O'Donnell, & Smith, 2015). Research indicates that most students do not understand the "vulnerability, threats, risks, and mitigations associated with cyber threats (Thompson, Herman, Scheponik, Oliva, & Shrman, 2018, p. 1). In response to these issues, teachers are increasingly navigating issues of cybersecurity and citizenship in the high school classroom. For example, teachers can integrate cyber practices such as creating secure passwords, understanding their cyber vulnerabilities, and identifying different types of phishing emails. Teachers can also integrate more skill-based content related to a programming language, design thinking, and problem solving; however, most teachers are not familiar with the GenCyber Principles (Author). Furthermore, most teacher preparation programs do not include a focus on cyber principles nor do they incorporate course requirements that help teachers connect their area of study to cyber principles (Pusely & Sadera, 2011). Because of the importance of cyberliteracy amongst students and the lack of teacher training in this area, programs have begun supporting teachers to learn about cyber principles and how to integrate them into classroom instruction across a variety of fields. One such program is jointly funded by the National Security Agency and the National Science Foundation, GenCyber, and its aim is to equip teachers with the necessary knowledge of sixteen GenCyber Principles and Cybersecurity Concepts (GenCyber, 2019) and the pedagogical content knowledge to support student cyberliteracy. This study explores a professional development program, funded by GenCyber, in which thirty-seven middle/high school teachers experienced inquiry-based learning, focused classroom discourse, and collaborative learning that centered on GenCyber Principles and Cybersecurity Concepts. The research questions that guided the study are as follows:

1. In what ways did participating in the GenCyber Knights experience effect teachers' knowledge of GenCyber Concepts?
2. In what ways did participating in the GenCyber Knights experience foster teachers' connection of the GenCyber Concepts to their classroom?
3. How did the unifying concept of dilemma support participants' engagement in cyber security and cyber citizenship?
4. How did the GenCyber Knights experience foster growth in participants' programming and technical skills which are transferrable to their classrooms?

This work builds on similar experiences for middle and high school students, which have shown promising results for increasing the availability of cybersecurity workforce pathways for students (Author.) In this work, students explore cybersecurity through independent and team modules, explore digital forensics with hands-on case study projects, and learn basic computer programming skills with robots through team-based problem-solving design and implementation. This GenCyber Knights Teacher Camp is distinct in the focus on training teachers in a myriad of disciplines to integrate these ideas in their own classrooms and to model good cybersecurity and cyber citizenship practices for their own students. As the GenCyber Knights program expands, opportunities and access for cyber pathways expand for students.

2. Background

GenCyber teacher camps are designed to work with the participating teachers to build their technology literacy in the area of GenCyber Principles and cyber security. Two frameworks are most relevant to implementing effective instruction in this context: the Davies (2011) framework for developing technology skills and expertise and the TPACK framework developed by Koehler and Mishra (2006).

The Davies framework "involves three levels: awareness, praxis, and phronesis" (p. 48). The awareness stage requires teachers to become aware of the basic technologies available to them and "the basic purposes and functions involved" (p. 48). At the praxis level, teachers are increasing their expertise in using the technology and are able to complete simple tasks. This level often involves expert models and practice "involving simulated problem-solving activities" (p. 49). Finally, at the phronesis level, teachers become adept at using the technology and can answer the question, "Why do I use or not use this technology in this specific situation?" (p. 49). In the context of GenCyber education, teachers are introduced to each technology and GenCyber Principle by experienced models and provided opportunities to interact with the technology, both of which help to build participants' self-efficacy (Somekh, 2008; Ertmer, 2005). Finally, teachers would develop a content-based lesson plan implementing the GenCyber Principles and one or more of the technologies.

The TPACK framework, maintains that in order to effectively integrate technology into instruction, teachers need to have a deep understanding of how each of the components (e.g., technology, pedagogy, and content knowledge) "interact, constrain, and afford each other" (Koehler, Mishra, Kereluik, Tae, and Graham, 2014). The TPACK framework depends on "a consideration of the interactions among technology, content, and pedagogy" (Ertmer & Ottenbriet-Leftwich, 2010, p. 259). Like the Davies framework, in order to accomplish this, teachers must understand "(a) the technology tools themselves, combined with (b) the specific affordances of each tool that, when used to teach content, enable difficult concepts to be learned more readily, thus resulting in the achievement of meaningful student outcomes" (Angeli & Valanides, 2009, cited in Ertmer & Ottenbriet-Leftwich, p. 259). Prior research on GenCyber professional development activities have demonstrated a positive impact on teachers' understanding and application of TPACK. A 2016 GenCyber workshop (Ivy, Lee, Fanz, & Crumpton, 2019) showed significant growth in participating teachers' TPACK.

3. Methods

3.1 Program Design

The professional development program was designed through collaboration between three Education faculty, one Computer Science faculty, and two STEM high school teachers. Designed as an opportunity to serve in-service middle and high school teachers across content areas in cyber-focused fields, this program aimed to expand teachers' understanding of and ability to create high quality cyber-focused curricula relevant to their field.

The program was a five-day, non-residential summer camp for teachers from a variety of disciplines. The camp was offered to forty practicing teachers ranging from novice teachers (less than 3 years in the field) to veteran teachers (over 15 years of experience). The camp was designed to provide teachers across all disciplines the opportunities to explore cyber citizenship concepts, programming concepts, and cybersecurity concepts with explicit connections to GenCyber Principles as both learners and educators. Teachers were introduced to ten cybersecurity First Principles but dove deeper into the six Cybersecurity Concepts (GenCyber, 2019) throughout the week.

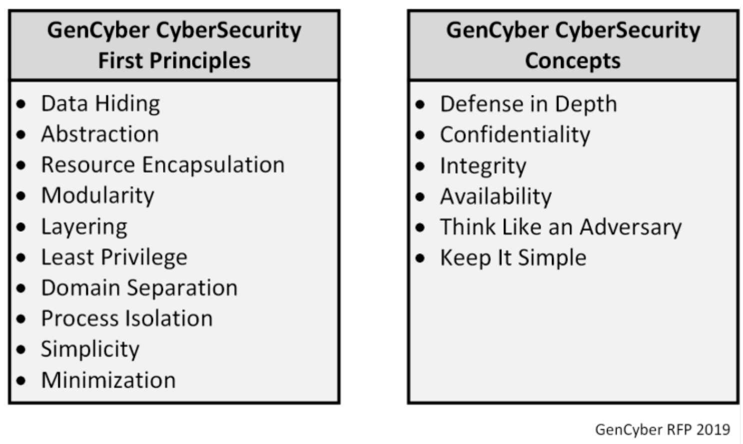


Figure 1. Cyber principles taught in program

The instruction during the professional development was designed to promote inquiry-based learning, focused classroom discourse, and collaborative learning to learn and apply cyber principles. Following the methods and practices of scientists (Keselman, 2003), inquiry-based learning is an effective instructional approach that improves student learning (Alfieri, Brooks, Aldrich, and Tenenbaum (2011). Throughout the week-long experience, teachers were also provided with structured time to design and refine cyber and/or programming lessons to integrate meaningfully and purposefully in their classrooms. Table 1 describes the daily activities of the professional development experience.

Table 1. Daily Schedule Overview

Activity	Description
Sign-In /Daily Warm-Up	Participants sign-in and participate in warm-up exercise in which the instructors present a cybersecurity scenario to consider and discuss.

Topic Presentations	Camp instructors or guest speaker present material to the participants.
Break	
Exploration Activity	Participants engage in a hands-on activity designed to introduce or reinforce GenCyber Principles and Concepts
Daily Reflection	Participants completed a Daily Reflection sheet designed to elicit what they have learned so far.
LUNCH Break	
Exploration Activity / Guest Speaker	Participants engage in a hands-on activity designed to introduce or reinforce GenCyber Principles and Concepts or a guest speaker presents material to them.
Break	
Lesson Planning	Participants explored, created, and shared lessons which connected cybersecurity and cyber citizenship concepts to their discipline.
Exit Ticket	Camp instructors used formative assessment to gauge participants understanding of daily material/activities.

The GenCyber Knights teacher camp curriculum centered on the unifying concept of dilemma. Throughout the week, participants grappled with a series of ethical and moral dilemmas related to cyber citizenship and technology. One example was the exploration of bioprinting: *Is it ethical to create "replacement parts" for living organisms? What are the potential benefits and consequences?* While teachers were learning and applying information about GenCyber Principles, they were also be learning how to effectively engage students in argumentation and inquiry-based learning. Participants explored their own cyber vulnerabilities and those of their students, worked to understand the associated challenges, and developed plans to troubleshoot areas of vulnerabilities. More detailed sampling of lessons is provided in Appendix A.

Participants were provided with a variety of tools and technology to use both as learners and teachers in order to support GenCyber implementation beyond the camp experience:

- Micro:bit and Scratch. During the first afternoon, participants explored Scratch programming with Micro:bits. Each teacher received five Micro:bit bundles to use in their classroom.
- Sphero SPRK+ and SpheroEDU. The SPRK+ robots and SpheroEDU app were integrated into the camp experiences. At the conclusion of the camps, each teacher received four Sphero SPRK+ devices to support classroom implementation.
- Tableau Public. During a session on Thursday, teachers explored Tableau Public as a data visualization tool. Participants discussed how this can be integrated into their classes.
- Flash Forge CreatorPro, Flash Print, and Tinkercad. Teachers explored the function of the 3D printer and its capabilities in relation to their content areas. They printed wheel cyphers for participants to use to encode and decode messages during the workshop.

Table 2. Connections of Tools to GenCyber CyberSecurity First Principles and Concepts

Tool Used for Workshop	**Connection to GenCyber First Principles and Concepts**
Micro:bit and Scratch.	Modularity Resource Encapsulation Simplicity
Sphero SPRK+ and SpheroEDU	Modularity
Tableau Public	Think Like an Adversary Integrity Defense in Depth
Flash Forge CreatorPro, Flash Print, and Tinkercad.	Confidentiality

The learning environment of the camp allowed for the facilitation of cooperative groups and collaboration for task completion. As shown in Table 1, there were situations presented in which teachers were introduced to terminology or conventions requiring direct instruction; however, most lessons across the week began with a problem and were followed by teachers' collaborating on solutions through the problem-solving process (Pólya, 1945). Teachers worked in teams to first understand the problem, devise a strategy, carry out the strategy, then reflect on their strategy and solution, devising a different strategy, when necessary. Small groups were usually self-selected, work was recorded and shared with the whole group. At times, groups were formed purposefully by grouping together similar subject areas/same schools and other times groups were assigned randomly to create diversity across disciplines and cohesiveness among the camp participants. Active exploration time was built into Sphero lessons. There were several hands-on activities, such as creating a prosthetic fingerprint, exploring and defining encryption, and history of and examples various cyphers. Guest speakers, which included an FBI agent in cybersecurity, an attorney in cyber law, and Chief Information Security Officer for a bank shared real-life cyber issues and solutions to further inspire and inform the participants.

Low-tech and high-tech materials and group assignments changed throughout the day, allowing diverse experiences across disciplines.

Teachers were presented with examples of lesson plans which connected GenCyber Cybersecurity First Principles and Concepts to subject area content, including English Language Arts, Mathematics, Science, and Foreign Language. Teachers had the opportunity to create lesson plans for their curricular area and classrooms. Teachers worked on two mini lesson plans at the beginning of the week, which helped them gain confidence and capacity to create a larger lesson plan later in the week. On day two, an hour was scheduled to link GenCyber concepts and principles to content standards and brainstorm ideas for lesson plans. Templates and expectations were also explained at this time (see Appendix B). On the last day of camp, final lesson plans were shared with the class for peer feedback and uploaded to the shared Google Drive.

3.2 Research Design

Guided by the recommendations of Creswell (2013), we used the survey approach to investigate the impact of the weeklong GenCyber camp on inservice teachers' Technology, Pedagogy, and Content Knowledge (TPACK). Survey research was the preferred method of data collection because of its economy, rapid turnaround time, and the standardization of the data (Babbie, 2012). Participating teachers completed a pre and post assessment. The survey is discussed in the Data Source section.

3.3 Participant Description

There were 37 teacher participants, representing 11 different school systems and districts. Of the participants, 75% were female and 25% were male. Teachers taught a variety of grade levels, but 50% taught in grades 6-8 and 50% taught in grades 9-12. Regarding the length of time in the classroom, 25% of teacher had served for 1-5 years, 20% for 6-10 years, 35% for 11-15 years, 5% for 16-20 years, and 15% for over 20 years. Figure 2 shows the specific content areas taught by the participants.

ANSWER CHOICES	RESPONSES	
English	15.00%	3
Math	20.00%	4
Science	20.00%	4
Computer Science	35.00%	7
Social Students	10.00%	2
Business	0.00%	0
Other (please specify)	35.00%	7
Total Respondents: 20		

#	OTHER (PLEASE SPECIFY)	DATE
1	Spanish; Video Production	6/21/2019 1:25 PM
2	Practical Technology	6/21/2019 1:18 PM
3	K-8 Technology	6/21/2019 1:15 PM
4	Innovation	6/21/2019 10:38 AM
5	Technology Coach	6/21/2019 9:50 AM
6	World Language	6/21/2019 9:21 AM
7	Library Media/School Technology Coordinator	6/21/2019 9:11 AM

Figure 2. Content areas taught by participants

Participating teachers indicated their prior involvement with and interest in cybersecurity principles in the following ways:

- I do not know much about cybersecurity and am curious about it. (30%)
- I already teach some cybersecurity and want to learn more. (20%)
- Have you done other things to learn about cybersecurity, e.g., workshops, classes, etc? (75% said no)
- I will be integrating cybersecurity into the subjects/classes I teach in the future. (70%)
- I am formulating plans for how I will integrate cybersecurity in my class or as an extracurricular. (79%)
- I am already taking concrete actions to integrate cybersecurity into my teaching/coaching. (21%)

When asked for more information about why they chose to come to the summer program, one teacher stated, “I currently teach a technology class and am always looking for/gathering resources and 'topics' that I can embed with instruction for that class in order to make sure my students are confident in their world.” Another teacher replied, “Our district is looking into revamping digital citizenship and cyber security is one aspect that has been overlooked in the big scheme of things. I wanted to get more information so that I was prepared and could get my school on board as well.”

3.4 Data Sources

Data sources included: participant reflections, pre-post surveys, and written lessons developed by teachers. The TPACK Developmental Model Self-Assessment Survey was co-developed and adapted by the authors and based on the themes and subthemes of the TPACK Standards and Development Model (Niess, et al., 2009). The TPACK Self-Assessment survey included 11 categories, adapted from the themes and subthemes of the TPACK development model. For each category, five levels of descriptors provided insight into the TPACK levels for participants. The five levels were Recognizing, Accepting, Adapting, Exploring, and Advancing. Each level was correlated with a numerical value from one to five, and the sum of the criteria provided an indexed TPACK rating for each iteration of the TPACK Self-Assessment survey.

3.5 Data Collection and Analysis

Data collection centered on three main sources: participant reflections, pre-post surveys, and written lessons developed by teachers. First, participating teachers were asked to reflect both individually and in group settings at the end of each day as well as overall after the program. Second, the GenCyber Knights project team surveyed participants to determine the impact of the camp. Initial teacher applications assessed their current levels of support and involvement in programming or cyber programs. At the conclusion of the camp, a survey including open-ended reflection questions on cybersecurity concepts was administered. In both instances, participants accessed the online Google Forms survey. No login was required, and participants completed the survey in one sitting. The approximate time for completion was 10-15 minutes, depending on the individual. Data was exported from Google Forms as an

Excel spreadsheet. Finally, the lessons teachers developed were assessed using the EQUIP Quality Review Protocol (Marshall, Smart, & Horton, 2010), which was made available to teachers to ensure key components of exploration, implementation, and discussion were present throughout the lessons. Data analysis of the multiple data sources included using the constant comparative methodology until researcher consensus was attained. Once complete, codes were examined for overlap and redundancy and collapsed into broad themes (Creswell, 2012). Emerging themes were triangulated within and across data sources, with careful attention to maintaining an audit trail back to original data.

4. Results

Results indicated the professional development program enabled teachers to iteratively reflect on best practices in cyber education while learning and applying the content of GenCyber Principles within the context of their own field of study. The GenCyber Knights professional development program resulted in the following outcomes:

4.1 In what ways did participating in the GenCyber Knights experience effect teachers' knowledge of GenCyber Concepts?

During each week, participants explored each of the concepts daily and contributed connections, which were both concrete and abstract. Teachers collaborated to identify content connections for their discipline related to each of the six GenCyber Concepts. GenCyber First Principles and Cybersecurity Concepts were emphasized multiple times per day throughout the camp. It was expected that different teachers would approach the application of these ideas in different ways. For example, an English teacher may plan a small unit focusing on cyberbullying and teach it through a unit culminating in a hearty debate on what defines a "responsible cybercitizen;" whereas an engineering teacher may design a unit on reverse engineering where students are challenged to solve a mock crisis using programming and problem-solving skills.

The implementation of the GenCyber principle, *Layering*, was our most successful. We theorize this is because of the straightforward nature of the concept, which involves using multiple levels of protection, similar to having several locks or layers of fencing to protect your personal property. Although we felt that the GenCyber concepts were so much easier for our attendees to grasp, that after a brief introduction to the principles, we concentrated primarily on the concepts. Our implementation of the "*Think like an Adversary*" concept cut across multiple topics and appeared to be very clearly understood by all participants by the end of the session. This success was likely due to the sleuth-like nature of exploring cases in which the motives of the adversary were discussed and debated. The most challenging concept to communicate was, ironically, "*Keep it Simple"*. The participants understood the concept well enough, but we had difficulty coming up with good cybersecurity-related examples. This is possibly due to the high depth of technical knowledge needed in order to analyze and determine the most efficient (and simple) strategy or configuration.

Figure 3 below shows what teachers considered the most important takeaways from the professional development, indicating the gains in their content, curricular, and pedagogical knowledge along with their motivation to incorporate these concepts into their teaching. These aforementioned types of knowledge are necessary for teachers to transition learning to teaching practice: 1) content knowledge (what to teach), 2) pedagogical content knowledge (how to teach it), and 3) curricular knowledge (when to teach it).

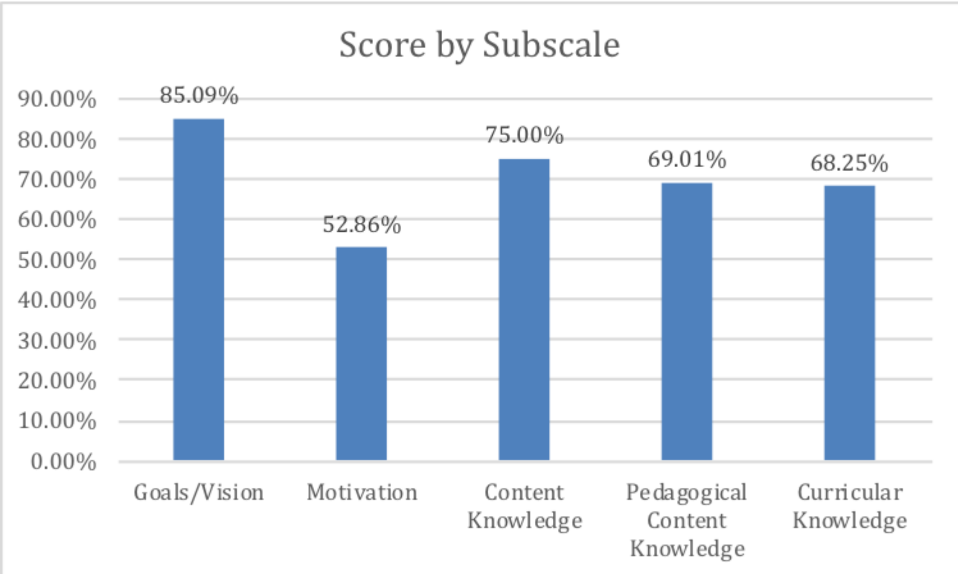


Figure 3. Participants report on the factors they considered the most important takeaways from the professional development

Participants reported relatively high expectations they would be integrating cybersecurity into their lessons at 85.09% of the possible total. This suggested the teachers believed that cybersecurity is important and relevant to their students. The lowest score for this camp is on motivation (52.86%). Motivation is reflective of the individual and is a difficult construct to directly affect within a camp. The lower score reflects concern among the camp attendees about finding the time and resources to meaningfully integrate cybersecurity into their respective curriculum/school. The scores on content knowledge (75.00%), pedagogical content knowledge (69.01%), and curricular knowledge (68.25%) also show possible room for growth.

4.2 In what ways did participating in the GenCyber Knights experience foster teachers' connection of the GenCyber Concepts to their classroom? The GenCyber Lesson Plan template was shared with participating teachers. Teachers collaborated to prepare lesson plans to be used in their classrooms which connected to GenCyber concepts and their content area standards. Teachers developed a variety of lesson plans to include the cybersecurity ideas they learned in the professional development program. The lessons were shared in Google Drive folders, which allows teachers to continue to collaborate.

This content-diverse group does not typically have an opportunity to collaborate on lessons, and the results were incredible. Teachers experienced camp content as learners first, then built connections to their classrooms. One Spanish Teacher, for example, changed the language on the device and students used simple block programming to program the robot. She also engaged students in discussing cyber concepts in Spanish. Social studies teachers explored the history of cybersecurity and built a timeline of societal vulnerabilities over decades to correspond to U.S. History. Table 3 below captures their ideas:

Table 3. Teacher developed lessons integrating GenCyber concepts into area of instruction

Content Area	Lesson Idea	Connection to GenCyber Principles/Concepts
English	Making counterclaims made by an opponent in an argument	Think Like an Adversary
Social Studies?	Examining personal digital footprints	Data Hiding; Defense in Depth; Confidentiality; Think Like an Adversary; Keep It Simple
Science	Create an education synthetic skin model. Students determine what materials they use to represent different receptors in their model and give justifications for their decisions	Domain Separation; Process; Isolation; Modularity; Abstraction; Simplicity; Minimization; Availability; Think Like an Adversary; Integrity; Keep It Simple
Ethics	All students issued iPads, email addresses, and Apple IDs. The goal is to think about cybersecurity, digital citizenship, and connection to real-life situations	Domain Separation; Layering; Simplicity; Minimization; Defense in Depth; Confidentiality; Integrity; Think Like an Adversary; Keep it Simple
Mathematics and Art	Students use their knowledge of Geometry shapes (2 and 3 dimensional) to discover simple shapes in the real world. Using photography, students take photos of architecture or organic objects that represent basic geometrical shapes.	Abstraction; Simplicity; Minimization; Availability; Keep it Simple
Biology	Students navigate a virus (a Sphero) as it enters the body, while encountering various defense systems along the way.	Domain Separation; Modularity; Least Privilege; Data Hiding; Layering;

		Defense in Depth; Think Like an Adversary; Integrity
Any course that requires research	Evaluating websites for research purposes	Data Hiding; Simplicity; Availability; Think Like an Adversary; Integrity; Keep it Simple
Science	Students explore how different surfaces will impact the Sphero when it comes to the concepts of speed, friction, and momentum.	Process Isolation; Simplicity; Data Hiding; Minimization; Least Privilege
Computer Science	Students work in pairs to complete a Digital Breakout that reviews nine Digital Citizenship elements.	Process Isolation; Layering; Least Privilege; Defense in Depth; Availability; Think Like an Adversary; Integrity

The teachers who attended the GenCyber Knights teacher camp planned interactive lessons to integrate into their classrooms. Over time the lessons could improve to be more transdisciplinary; however, the strategies used for planned delivery were interactive and promoted collaboration and problem solving. Lesson plans were analyzed using the EQUIP Rubric for Lessons and Rubrics, with a focus on the Instructional Supports category. Common strengths of lessons included thoughtful integration of technology and media, encouraging the use of academic language and terminology, engaging students in productive struggle through thought-provoking and relevant problems, and providing scaffolding, differentiation, intervention and supports for learners (Marshall, Smart, and Horton, 2010).

4.3 How did the unifying concept of dilemma support participants' engagement in cyber security and cyber citizenship?

Participants began their GenCyber Knights experience by exploring the Case of Terry Childs. As the week progressed, teachers were provided with tools (such as vocabulary, skills, and understanding of concepts) for preventing and navigating ethical dilemma. Three cyber security professionals, including a cyber law attorney, a cyber security bank executive, and an F.B.I. Special Agent specializing in cybercrimes, shared insights and engaged in dialogue with teachers. Participants mentioned many times in their reflections on how important it was to hear from experts in the field. One participant stated, "Guest speakers who are currently working in the field can share examples of threats as well as reinforce the need for cybersecurity professionals." Another teacher noted, "It gave me lots of knowledge and stories to tell my students, while also trying to give me concrete ideas of how to talk and work with my students about them." Understanding real-life cyber-related issues and solutions helped teachers consider how they could use dialogue and argumentation to engage students in relevant discussions.

4.4 How did the GenCyber Knights experience foster growth in participants' programming and technical skills which are transferrable to their classrooms?

Beginning with a discussion of different modalities, participants explored the functions of Spheros through programming in SpheroEDU during the second day of the workshop. After grappling with basic functions, teachers engaged in a series of Sphero challenges including a MazeChallenge, Jump the Shark Challenge, and a Shuffle Sphero Challenge. After persevering through these tasks as learners, participants linked the Spheros to their content standards and began brainstorming for their lesson plan. Reflections from the participants and the project team revealed they had learned a great deal about their own cyber vulnerabilities. Collectively, reflections centered on understanding that the most valuable qualification to becoming a cyber security professional is to be a great problem solver. They also reflected that cybersecurity and cyber vulnerabilities are not always high-tech (such as is the case with social engineering), and teachers also reported that students need to know cyber security careers exist in private and public career sectors and the demand is very high. Table 4 below captures teachers' perspectives about why GenCyber Principles and Concepts are important to teach students. Results indicate the top reasons are because it is an emerging field, students need to understand for their own safety.

Table 4. Teachers' level of agreement with statements on importance of GenCyber

	STRONGLY AGREE 7	6	5	4	3	2	STRONGLY DISAGREE1	TOTAL	WEIGHTED AVERAGE
Teaching cybersecurity is important to me because I want to help my students be safer online.	68.42% 13	31.58% 6	0.00% 0	0.00% 0	0.00% 0	0.00% 0	0.00% 0	19	5.68
Teaching cybersecurity is important to me because I can help build the cybersecurity workforce.	36.84% 7	31.58% 6	21.05% 4	10.53% 2	0.00% 0	0.00% 0	0.00% 0	19	4.95
Teaching cybersecurity is important to me because my school has a need for someone to teach it.	31.58% 6	15.79% 3	21.05% 4	10.53% 2	10.53% 2	10.53% 2	0.00% 0	19	4.16
Teaching cybersecurity is important to me because it is an emerging discipline.	36.84% 7	52.63% 10	0.00% 0	10.53% 2	0.00% 0	0.00% 0	0.00% 0	19	5.16
Not applicable - teaching cybersecurity is not important to me.	0.00% 0	5.26% 1	0.00% 0	0.00% 0	5.26% 1	5.26% 1	84.21% 16	19	0.42

13 / 41

Bellarmine University GenCyber Teacher Camp, June 17-June 21

Despite teachers grasping the importance of GenCyber ideas, they did have concerns about classroom implementation. Table 5 details their perspectives on classroom implementation. While weighted averages overall indicated concerns were minimized, teachers still acknowledge teaching GenCyber concepts would demand time and effort and require a plethora of resources.

Table 5. Teachers' level of agreement with concerns of GenCyber classroom implementation

	STRONGLY AGREE7	6	5	4	3	2	STRONGLY DISAGREE1	TOTAL	WEIGHTED AVERAGE
Teaching/coaching cybersecurity will demand a great deal of my time and effort.	10.53% 2	5.26% 1	47.37% 9	26.32% 5	5.26% 1	5.26% 1	0.00% 0	19	2.26
Teaching/coaching cybersecurity will make my work as a teacher harder.	10.53% 2	5.26% 1	26.32% 5	31.58% 6	10.53% 2	10.53% 2	5.26% 1	19	2.79
I am concerned about not having enough time to prepare to teach/coach cybersecurity.	10.53% 2	31.58% 6	31.58% 6	10.53% 2	15.79% 3	0.00% 0	0.00% 0	19	1.89
I am concerned about not having enough resources to teach/coach cybersecurity.	21.05% 4	5.26% 1	21.05% 4	10.53% 2	26.32% 5	10.53% 2	5.26% 1	19	2.68

5. Discussion

We sought to support teachers across disciplines regarding cyberliteracy skills. Schools and districts around the nation are offering STEM programs to enhance the mathematics and science standards by providing educational opportunities for students through the use of hands-on activities and projects. Computer Education Support programs for schools and districts are building and supporting professional development focused on project and problem-based learning, STEM programming, and robotics. Rigorous and high-quality STEM magnet programs attract many middle and high school students. With the multitude of programming options available in STEM, supporting teachers in their development and implementation of cyber-based across the curricula is paramount. It is not simply the job of STEM educators to incorporate these principles into classrooms. Because cyberliteracy pervades all classrooms, not only those focused around STEM, we assert involving teachers from across disciplines strengthens the efforts of schools and STEM-based programming to ensure students are equipped with GenCyber Principles that will help them navigate life.

Modeling and hand-on application of skills is valued across disciplines and is an ever-present aspect of the GenCyber Knights Teacher Camps. Further, these skills are grounded in ISTE Educator Standards as teachers engage as Learners, Leaders, Citizens, Collaborators, Designers, Facilitators, and Analysists (ISTE, 2000).

As technology has become a more prominent part of our everyday lives in, and outside of, schools, cybersecurity knowledge and skills have become increasingly important. Konak (2018) points out that cybersecurity is "projected to grow to a 170 billion global market by 2020 from $75 billion in 2015 (Morgan, 2015). On the opposite end of the equation a global shortage of 1.5 million cybersecurity professionals is predicted by 2019 (Morgan, 2019)." The move to online instruction as a result of the pandemic has also increased cyberattacks. Lily Hay Neman (2020) reported that in a 30-day period during the pandemic, 60 percent of worldwide corporate and institutional malware incidents were in the education industry. In the U. S., the fourth largest district in the country, Miami-Dade in Florida's suffered malware attacks that interrupted online instruction for several days (Wright, 2020). The growth in technology, especially as a result of recent shifts to remote learning brought about due to the pandemic, have increasedly demanded for teachers to become more technologically savvy and model the GenCyber Cybersecurity First Principles and Concepts. Professional growth opportunities like this can prepare teachers with the necessary knowledge and skills to effectively integrate these principles into their curriculums to ensure students learn and practice the skills.

6. Conclusion

There is a dearth of research on pre- and in-service teachers' knowledge and skills regarding cybersecurity (Pusey and Sadera, 2011-2012) and GenCyber principles as well as how these teachers should go about teaching students the knowledge and skills they will need to meet the demands of private industry for cybersecurity professionals (Thompson et al., 2018). The 2019 GenCyber Knights Teacher Camp provided our team with the opportunity to develop an inquiry-based and collaborative learning experience for middle and high school teachers to learn about the GenCyber Principles and Cybersecurity concepts and strategies for integrating them into their curriculum. The results of our study after the conclusion of the program indicated that it enabled teachers to iteratively reflect on best practices in cyber education while learning and applying the content of GenCyber Principles within the context of their own field of study. Moreover, a high percentage of the participants indicated in follow up surveys that they would integrate cybersecurity education in their curriculum in some fashion using techniques and ideas they acquired from the camp. Our group intends to continue implementing the GenCyber Knights program to train additional pre-service and in-service teachers in the region in the critical area of cybersecurity.

Acknowledgements

This work was jointly supported by the National Security Administration and the National Science Foundation GenCyber Program under Grant Number H98230-19-3-1-017.

References

Alfieri, L., Brooks, P.J., Aldrich, N.J., & Tenenbaum, H.R. (2011). Does discovery-based instruction enhance learning? *Journal of Educational Psychology, 103,* 1-18.

Angeli, C., & Valanides, N. (2009). Epistemological and methodological issues for the conceptualization, development, and assessment of ICT–TPCK: Advances in technological pedagogical content knowledge (TPCK). *Computers & education*, *52*(1), 154-168.

Babbie, E. (2012). The practice of research. *Belmont: Woodsworth*.

Creswell, J. W. (2012). Qualitative inquiry & research design: Choosing among five approaches (4th ed.). Thousand Oaks, CA: Sage.

Creswell, J. W. (2013). Steps in conducting a scholarly mixed methods study.

Bamford, A. (2005). Cyber-bullying: Students as young as eight are describing cases of cyber-bullying, cyber-stalking and virtual teasing. *Classroom, 1*, 18-19.

Davies, R. S. (2011). Understanding technology literacy: A framework for evaluating educational technology integration. *TechTrends*, *55*(5), 45.

Ertmer, P. A. (2005). Teacher pedagogical beliefs: The final frontier in our quest for technology integration?. *Educational technology research and development*, *53*(4), 25-39.

Ertmer, P. A., & Ottenbreit-Leftwich, A. T. (2010). Teacher technology change: How knowledge, confidence, beliefs, and culture intersect. *Journal of research on Technology in Education*, *42*(3), 255-284.

GenCyber, 2019 GenCyber CPF. (2019). Retrieved July 14, 2020, from https://www.gen-cyber.com/proposals/rfp/gc-2019/

International Society for Technology in Education. (2000). ISTE national educational technology standards (NETS). Eugene, OR: International Society for Technology in Education.

Ivy, J., Lee, S. B., Franz, D., & Crumpton, J. (2019). Seeding Cybersecurity Workforce Pathways With Secondary Education. *Computer*, *52*(3), 67-75.

Keselman, A. (2003). Supporting inquiry learning by promoting normative understanding of multivariable causality *Journal of Research in Science Teaching*, *40*, 898-921.

Kessel Schneider, S., O'Donnell, L., & Smith, E. (2015). Trends in cyberbullying and school bullying victimization in a regional census of high school students, 2006–2012. Journal of School Health, *85*, 611–620.

Koehler, M. J., Mishra, P., Kereluik, K., Shin, T. S., & Graham, C. R. (2014). The technological pedagogical content knowledge framework. In *Handbook of research on educational communications and technology* (pp. 101-111). Springer, New York, NY.

Konak, A. (2018). Experiential learning builds cybersecurity self –efficacy in K-12 students. *Journal of Cybersecurity Education, Research and Practice*, 1(6), 1-16.

Levin, Douglas A. (2020). "The State of K-12 Cybersecurity: 2019 Year in Review." Arlington, VA: EdTech Strategies, LLC/The K-12 Cybersecurity Resource Center. Available online at: https://k12cybersecure.com/year-in-review/

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record*, *108*(6), 1017-1054.

Newman, L. H. (2020, June 1). Schools already struggled with cybersecurity. Then came Covid-19. *Wired*. https://www.wired.com/story/schools-already-struggled-cybersecurity-then-came-covid-19/

National Science Foundation (NSF). (2020). Cybersecurity: Tech, tools and training to safeguard the future. https://www.nsf.gov/news/special_reports/cybersecurity/index.jsp

Marshall, J. C., Smart, J. & Horton, R. M. (2010). The design and validation of EQUIP: An instrument to assess inquiry-based instruction. *International Journal of Science and Mathematics Education*, 8, 299–321. https://doi.org/10.1007/s10763-009-9174-y

Pólya, G. (1945). How to solve it: A new aspect of mathematical method. Princeton, N.J: Princeton University Press.

Pusey, P., & Sadera, W. A. (2011-2012). Cyberethics, cybersafety, and cybersecurity: Preservice teacher knowledge, preparedness, and the need for teacher education to make a difference. *Journal of Digital Learning in Teacher Education, 28*(2), 82–88.

Readiness and Emergency Management for Schools (REMS) Technical Assistance (TA) Center. (2017). "Cybersecurity considerations for K-12 schools and school districts fact sheet". https://rems.ed.gov/docs/Cybersecurity_K-12_Fact_Sheet_508C.PDF.

Somekh, B. (2008). Factors affecting teachers' pedagogical adoption of ICT. In *International handbook of information technology in primary and secondary education* (pp. 449-460). Springer, Boston, MA.

Thompson, J. Herman, G. L., Scheponik, T., Golaszewski, E., Sherman, A. T., DeLatte, D., Patsourakos, K. Phatak, D., & Oliva, L. (2018). Student misconceptions about cybersecurity concepts: Analysis of think-aloud interviews with students. *Journal of Cybersecurity Education, Research and Practice*, 1(5), 1-29.

Wright, C. (2020, September 2). As cyberattacks persist, $15 million contract for online school platform never signed. *Miami Herald*. https://www.miamiherald.com/news/local/education/article245434515.html

Copyrights

Appendix A
Sample Lesson Descriptions

A sampling of lessons descriptions selected from the Camp Overview is included below to highlight the connections made to the project goals, GenCyber Principles, and Cybersecurity concepts.

Sample 1: Big Data Visualization Lesson

For this lesson, we used Microsoft Excel and Tableau to teach participates how to acquire publicly available data from our city's Open Data website and how to visualize it with Tableau (https://www.tableau.com)[1].

Our city's repository for data provides a significant amount of publicly available data on its Open Data website. Examples of available data include a salary schedule for all Metro Employees, housing permit applications, crime statistics and health/environmental data. Moreover, most of the data are provided in raw, downloadable formats that allow citizens the ability to peruse and manipulate it on their own.

While an enormous amount of data is available through the Open Data website, assimilating and visualizing it is not an easy task. Tableau is a commercial program designed specifically to assist non-programmers with visualizing large and complex datasets. Typical visualizations could include bar plots, scatter plots, heat maps, and geographic maps. Tableau provides its software free to educational institutions; therefore, participants would be able to use their school email address to download the software and use it in their classes.

The lesson included an introduction to Open Data and Tableau. Microsoft Excel, with which many people are already familiar, will be used to look the raw data; most hands-on activities will occur in Tableau.

In the interest of serving the widest range of academic disciplines as possible, we focused on the following data sets from Open Data:

1. Employee Salary Data (statistics)
2. Employee Characteristics Data (statistics)
3. Crime Data (there are several years' worth of data) – potential mapping topic (social studies/geography)
4. Historical Marker Data – potential mapping topic (social studies/geography)
5. Public health complaints (health)
6. Air quality (health/science)

Through this lesson, participants will:

1. Create bar charts

2. Create scatter plots
3. Create heat maps
4. Generate simple maps
5. Investigate data - trying different visualizations with a single data set to focus on different aspects of the data. Tableau provides rich support for this type of investigation.

The combination of Tableau and Excel could be used in many ways in the classroom. This lesson will provide participants the tools and ideas to develop approaches to easily integrate big data literacy and visualization techniques in their classroom across a wide variety of disciplines. Furthermore, this lesson would provide participants the tools to introduce the GenCyber principles of data hiding, least privilege, confidentiality, integrity, and availability.

[1] For this lesson we considered using Python and R, both software packages heavily used by data scientists and data analysts. However, we opted for Tableau because its goal is to enable users to easily visualize data without needing to learn a programming language. Python or R would be more appropriate for a week-long course devoted entirely data visualization and analysis.

Sample 2: Micro:bit Lesson

The micro:bit (www.microbit.org) is a small computing device that contains 25 individually programmable LED lights (in a 5x5 grid), 2 programmable buttons, 3 input/output connections, accelerometer, compass, and Bluetooth capability within a 4 cm x 5 cm circuit board with a cost of approximately $15. It can be powered with AA batteries and programmed through a microUSB connector. As an educational tool, this is highly scalable to different educational levels, because it can be programmed through a drag-and-drop block programming interface (similar to that used with Scratch and Lego robotics) to teach younger students but can also be programmed with a variant of Python for students who are familiar with that programming language.

The micro:bit was used by participants to program nametags for themselves. This will demonstrate three components of the GenCyber curriculum, as we initially introduce the programming language (block or Python) as a way to flash individual LEDs on and off, and then demonstrate the tedium of coding individual LEDs to show a scrolling name before discussing the advantages and disadvantages of programming the scrolling name using prebuilt letters.

This then led into a discussion of what the micro:bit can be used for in the classroom, and one project the instructor suggested making is a password keeper. Cybersecurity best practices demands different passwords for different websites/applications, and preferably passwords made of random combinations of many letters, numbers, and punctuation. People who do not follow this practice often explain that they would have a hard time remembering even one long random password, much less many of them. The participants were asked, how can we use the 5x5 LED display to display a password in a way that only the owner can correctly interpret? This can range from pictograms for younger students, to Morse-code style short and long flashes for higher level students.

This lesson provided the participants with tools to easily introduce GenCyber principles of abstraction and data hiding at different ability levels and lays the groundwork for a later session on encryption.

Appendix B
Lesson Planning Template

GenCyber

LESSON TITLE

Lesson Description: This is a general description of the lesson that provides an overview of the topics included in the lesson along with an indication of the depth of coverage. It should be 3-5 sentences long.

Prerequisite Knowledge: These are statements of what students are expected to know or be able to do in order to start this lesson.

Length of Completion: What is the length of the lesson?

Level of Instruction: Is this lesson intended for high school, middle school, or elementary school. This the lesson appropriate for advanced, intermediate, or beginner learners?

Applicable First Principles &/or Concepts: Please select the Principles or Concepts covered in this lesson.

GenCyber First Principles

Domain Separation	Abstraction
Process Isolation	Data Hiding
Resource Encapsulation	Layering
Modularity	Simplicity
Least Privilege	Minimization

GenCyber Cybersecurity Concepts

Defense in Depth	Availability
Confidentiality	Think Like an Adversary

Page | 1

Integrity Keep it Simple

Resources that are Needed: Specify the resources needed to complete the lesson. This includes a list of PowerPoint slides, data files, supplementary reading, assessment activities, videos, etc. needed for this lesson. Please properly cite and acknowledge sources.

Accommodations Needed: Specify the any accommodations needed to complete the lesson. Examples include closed captioning for hearing impaired students, accommodations for students with disabilities, etc. (Please note all products – i.e. videos - created in support of the GenCyber grant must have closed captioning. If the lesson is directing the user to an already existing video, please note if the video is available closed captioning.)

LEARNING OUTCOMES

LESSON LEARNING OUTCOMES

- The learning outcomes should be statements of what students should be able to DO at the end of instruction. Use outcome examples such as Design/Built, Test/Defend, Compare/Contrast, Apply/Use, Explain/Discuss, Identify/Describe
- It would be typical for a lesson to have 1-3 learning outcomes

LESSON DETAILS

Interconnection: If the lesson is part of a series of lessons or activities include a list here with the titles to the other activities.

Assessment: This includes a brief description of any assessment activities used, formally or informally. Examples include Presentations, Projects, Writing Assignments, Observations, Walk around, Oral Questions, Labs, Other.

Extension Activities: This includes a brief description of any follow-up activities used, formally or informally.

Page | 2

Computational Thinking Skills, Programming Self-Efficacies and Programming Attitudes of the Students

Mustafa Serkan Günbatar[1]

[1] Van Yüzüncü Yıl University

DOI: https://doi.org/10.21585/ijcses.v4i2.96

Abstract

The purpose of this research is to examine Computational Thinking (CT) skills, Computer Programming Self-Efficacy (CPSE), and Computer Programming Attitude (CPA) of middle school students who took the Information Technologies & Software (IT&S) courses and those who did not and make various analyses according to the relationships between these variables. It is a correlational study and participants are 5th, 6th, 7th and 8th-grade middle school students. Computational Thinking Levels Scale, Computer Programming Self-Efficacy Scale, and Educational Computer Games Assisted Learning Coding Attitude Scale were used as data collection tools. Multivariable regression analysis and Multivariate Analysis of Variance were performed as statistical technics. As a result of the analysis, it was found that CPA and CPSE variables are significant predictors for CT skills. Both students who took IT&S course and those who did not take have moderate CPSE levels. But students who took IT&S course have a statistically significantly higher CPSE scores. In addition, it was observed that both those who took IT&S course and those who did not have moderate CPA and did not differ statistically, and students who took IT&S course had a high level of CT, while those who did not take the course were moderate and statistically different. It was also found that students who took IT&S courses made cumulative progress in terms of CPSE, CPA and CT variables compared to those who did not. The obtained results are discussed within the framework of middle school IT & S course curriculum held in Turkey in 2018.

Keywords: computational thinking skills; computer programming self-efficacy; computer programming attitude; middle school students; information technologies & software course

1. Introduction

Almost all of today's middle school students are growing in a world surrounded by digital equipments. Even if they do not have these digital equipments, they have information about what they do and how they are used. One of the theories that can best explain this is the Social Learning Theory by Albert Bandura (1971). Bandura mentions learning through modelling in social environment. Thanks to the informative function of the model, the learning product emerges, and the learners learn by observing. The learner first perceives and interprets the situations in the social environment; then he remembers that they interpret and code into their long-term memory; then he practices his observations; finally, if he has the motivation to repeat this observation, he repeats the behavior at different times (Bandura, 1971). We can call this generation a digital native, born into the digital world and speaking the language of the digital world as its mother tongue (Prensky, 2001a). Communicating with digital natives who learn to use the technological language as their mother tongue in a social environment can be through well-designed games or activities in teaching environments (Prensky, 2001b). At this point, teaching by activities such as problem solving in the process comes to mind.

P21 (Partnership for 21st century learning), which has international members and aims to develop educational policies, has provided a framework for 21st century learners. The framework covers elements under two main themes: Learning outcomes and Support systems (P21, 2020). These elements, which are essential for all students at the point of being ready for the 21st century, are directly or indirectly related to technology. In parallel, institutions such as ISTE (International Society for Technology in Education) and CSTA (Computer Science Teachers Association) also provided an opinion on the concept of CT brought up by Wing (2006) and how to develop it at the K-12 level (ISTE, 2018; ISTE & CSTA, 2011). Turkish MoE (Ministry of Education), which was not indifferent to such studies, updated its middle school IT&S course curriculum in 2018. The feature of this new program is that it gives a large space to the problem solving and programming unit. When looking at their specific goals, it is seen that great importance is given to problem solving and CT skills (MoE, 2018a; MoE, 2018b).

In the literature, there are studies examining the relationship between CT and programming performance according to various variables (Shute, Sun & Asbell-Clarke, 2017; Oluk & Korkmaz, 2016; Durak & Saritepeci, 2018). These studies generally deal with variables related to cognitive structure. However, affective variables are also important for CT skills (Román-González, Pérez-González & Jiménez-Fernández, 2017; Román-González, Pérez-González, Moreno-León, & Robles, 2018) and it is recommended to establish regression models in the future studies to predict CT skills (Avcu & Ayverdi, 2020). In the literature review, the first study that comes to mind about the CT skills at the middle school level, CPSE (Computer Programming Self-Efficacy) and CPA (Computer Programming Attitude) were not studied in the same experiment together. From this point of view, the CPA, CPSE and CT of middle school students who took the IT&S course and those who did not take were examined, and various analyses were made according to the relationships between these variables. In this context, research questions of the study can be listed as follows.

RQ1: Are CPA and CPSE variables significant predictors for CT skills?
RQ2: What is the distribution of CT skills, CPA and CPSE measurements of students who took IT&S course and those who did not?

RQ3: Do the students who took the IT&S course and those who did not take differ according to the common effect of the CPA, CPSE and CT skills variables?

2. The study

2.1.Research Design

In this study, CT skills, CPSE and CPA of middle school students were measured. Then, the relationships between these variables were examined and it was determined whether the aforementioned measurements differ according to whether students took IT&S course or not. Therefore, it is a correlational study. As in this study, studies to reveal the relationships between variables are evaluated within the scope of the relational screening model (Fraenkel & Wallen, 2009).

2.2. Participants

Participants are the students who were studying in 5, 6, 7 and 8th-grade middle school in 2018-2019 spring semester in Van province of Turkey. Van is in the east of Turkey. The data were collected in the first quarter of 2019. Students were from four different middle public schools. Analysis was carried out with the data of 506 students in total, 221 of them are female and 285 of them are male. Students took IT&S course according to the infrastructure of computer laboratory in their schools.

2.3. Instrumentation

2.3.1. Computational Thinking Levels Scale

Computational Thinking Levels Scale (CTLS) was used to measure students' CT skills. The scale is a five-point Likert type and adapted by Korkmaz, Çakır & Özden (2016) for middle school students. The scale has five sub-dimensions. The Cronbach Alpha coefficient of the Creativity sub-dimension consisting of four items is .640; the four-point Algorithmic Thinking sub-dimension was .762; the four-point Collaboration sub-dimension .811; the four-point Critical Thinking sub-dimension was .714; the six-item Problem Solving sub-dimension was calculated as .867. The Cronbach Alpha coefficient of the 22-item scale is .809. As a result of Confirmatory Factor Analysis performed in the adaptation process, goodness of fit values were determined as RMSEA = .074, S-RMR = .078, GFI = .89, AGFI = .84, CFI = .91, NNFI = .91, IFI = .90. According to these results, it has been stated by the researchers that the scale model has acceptable fit values. In addition, according to the results, it was found that the scale is a reliable and valid measurement tool to measure Turkish students' CT skills.

2.3.2. Computer Programming Self-Efficacy Scale

Computer Programming Self-Efficacy Scale (CPSES) developed by Kukul, Gökçearslan & Günbatar (2017) was aimed to measure students' CPSE. As a result of the exploratory factor analysis, the scale took its final shape with 31 items and consists of one factor. Cronbach Alpha coefficient of the five-point likert type scale is .950. As a result of confirmatory factor analysis Goodness of fit values were found as: X^2 / df = 1.84; RMSEA = .06; NFI = .95; NNFI = .98; RMR = .068; CFI = .98; IFI = .98; GFI = .82 and AGFI = .79. According to the obtained data model for scale, it was found that Turkish middle school students have compliance with acceptable values and the model has a high level of reliability.

2.3.3. Computer Programming Attitude Scale

Educational Computer Games Assisted Learning Coding Attitude Scale was used to determine the students' CPAs. The Scale was developed by Keçeci, Alan & Kırbağ-Zengin (2016). The scale, consisting of a total of 28 items, is a five-point Likert type. To determine the sub-factor structure of the scale, exploratory factor analysis was performed. The scale consists of three sub-factors. The Cronbach Alpha coefficient of the 12-item sub-factor, which is called as a request against coding learning, is .840; the 11-item sub-factor, which is called the interest for the use of computer games in education for education purposes is .736; the five-item sub-factor called concern about the asocialization of the computer is .481. The Cronbach Alpha coefficient of the whole scale is .833. As a result of the analysis, it has been revealed that the scale is sufficiently valid and reliable.

2.4. Data Analysis

Firstly, descriptive statistics related to CT, CPSE and CPA measurements of the students were presented. Then, multivariable regression analysis was performed to determine significant predictors for CT. Multivariate Analysis of Variance (MANOVA) was performed to determine if students who took IT&S courses and those who did not take differed according to the common effect of CT, CPSE and CPA measurements. Finally, One-way analysis of Variance (ANOVA) was performed to compare the CTS, CPSE and CPA measurements of students who took IT&S courses and those who did not.

3. Findings

RQ1: Are CPA and CPSE variables significant predictors for CT skills?

Table 1. Result of multiple regression analysis conducted for CT skills predict

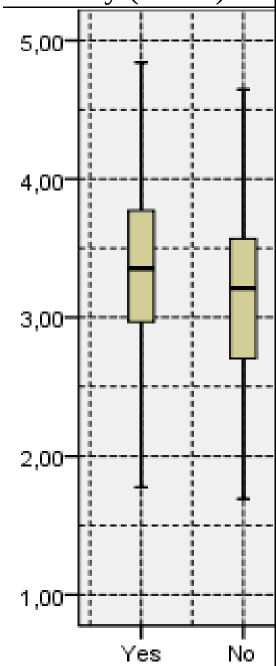
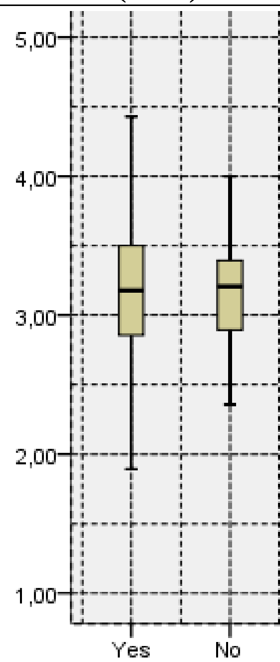
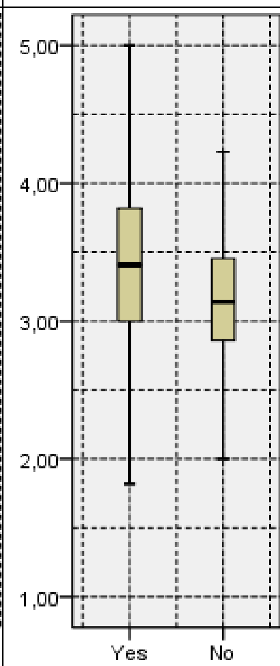
Variable	B	Standard Error	β	t	p	Zero-order r	Partial r
Fixed Value	1.188	.166		7.169	.000		
CPSE	.465	.037	.485	12.668	.000	.530	.492
CPA	.210	.046	.175	4.578	.000	.300	.200
R=.557		R^2=.310					
$F_{(2,503)}$=113.026		p=.000					

For the reference points, Pearson correlation coeffients (r) can take values from -1 to +1. We can interpret values .10 to .29 as small (low), .30 to .49 as medium (moderate), .50 to 1.0 as large (high) (Pallant, 2007). According to Table 1, it is seen that there is a positive and moderate (r = .530) correlation between CT and CPSE according to bilateral and partial correlations between predictor variables and predicted variable, and when CPA variable is controlled, partial correlation between CT skill and CPSE is similarly positive and moderate (r = .492). It is seen that there is a positive and moderate (r = .300) relationship between CT skill and CPA, and when the CPSE variable is controlled, there is a positive and low (r = .200) relationship between CT skill and CPA.

When CPSE and CPA variables are considered together, they offer a moderate and significant relationship with CT skills scores, R = .557, $R^2 = .310$, $p < .01$. The CPSE and CPA variables explain approximately 31% of the total variance in the CT skills.
According to the standardized regression coefficient (β), the significance level of the CPSE variable for CT skills is higher than for CPA. When the t-test results related to the significance of the regression coefficient are examined, it is seen that both variables are significant predictors of CT skills ($p < .01$). Regression equation for predicting CT skills is given below.

$$CT = 1.188 + .465 * CPSE + .210 * CPA \quad (1)$$

RQ2: What is the distribution of CT skills, CPA and CPSE measurements of students who took IT&S course and those who did not?

Taking Information Technologies and software course		Computer programming self-efficacy (CPSE)				Computer programming attitude (CPA)				Computational thinking (CT)			
													
	N	Min	Max	$\bar{X}$	Sd	Min	Max	$\bar{X}$	Sd	Min	Max	$\bar{X}$	Sd
Yes	426	1.26	4.84	3.36	.60	1.86	4.50	3.18	.51	1.68	5.00	3.45	.59
No	80	1.69	4.65	3.16	.63	2.36	4.18	3.18	.40	2.00	4.45	3.17	.49

Total	506	1.26	4.84	3.33	.61	1.86	4.50	3.18	.49	1.68	5.00	3.40	.59

Figure 1. Distribution of CPSE, CPA and CT measurements of students who took IT&S courses and those who did not

In figure 1, there are values related to the distribution of CPSE, CPA and CT skills measurements of students who took IT&S course and those who did not take. 426 of the students participating in the study took IT&S course and 80 of them did not. Mean ranges from five-point likert scales can be interpreted for 1.00-1.79 as *very low*; 1.80-2.59 as *low*; 2.60-3.39 as *moderate*; 3.40-4.19 as *high*; 4.20-5.00 as *very high.* While the average CPSE of students who took the course is $\bar{X}$ = 3.36 (moderate), the average of those who did not take the course is $\bar{X}$ = 3.16 (moderate). CPA averages of students who took the course and those who did not are very similar, and their average is $\bar{X}$ = 3.18 (moderate). While the average of the CT skills of the students who took the course is $\bar{X}$ = 3.45 (high), the average of those who did not take is $\bar{X}$ = 3.17 (moderate).

RQ3: Do the students who took the IT&S course and those who did not take differ according to the common effect of the CPA, CPSE and CT skill variables?
It was tested whether the data set belonging to the students, who took IT&S course and those who did not, provided the assumptions of the MANOVA analysis. It was seen that assumptions were provided. As a result of the analysis, according to the common effect of the dependent variables a significant difference was found among the students in terms of whether they took IT&S courses ($F_{(3,502)}$ = 6.071, p = .000; Wilks' Lambda (ʌ) = .965; Partial eta squared = .035).

Table 2. Mean, Standard Deviation Values and ANOVA Results of CT, CPSE and CPA Measurements according to taking IT&S course

	IT&S course	N	$\bar{x}$	Sd	df	F	p	Partial Eta Squared
CT	Yes	426	3.4481	.59261	1-504	15.737	.000*	.030
	No	80	3.1687	.49326				
CPSE	Yes	426	3.3645	.60484	1-504	7.890	.005*	.015
	No	80	3.1564	.62530				
CPA	Yes	426	3.1799	.50704	1-504	.002	.965	.000
	No	80	3.1773	.39746				

*p< 0.01

Table 2 shows the ANOVA results in which dependent variables are compared according to whether students took IT&S course or not. CT and CPSE variables of the students who took IT&S course have significantly higher levels than those who did not ($p < .01$). In terms of CPA, students who took the IT&S course and those who did not take do not differ ($p > .05$).

4. Discussion of Findings

This research was carried out to examine CT, CPSE, and CPA of middle school students who took IT&S course and those who did not take and to make various analyzes according to the relationships between these variables. The results obtained revealed that the variables CPA and CPSE are significant predictors for CT skills. In terms of CT skills predict, the CPSE variable is stronger than the CPA variable. CT, which is derived from computer science (Yadav, Stephenson & Hong, 2017) and identified with programming, has no clear definition (Computing at school, 2020).

However, it is known that CT is not the same as programming skills. It can be said that making a computer program creates an infrastructure to think computationally (Shute, Sun & Asbell-Clarke, 2017). In parallel, There are studies that show results regarding the relationship between Programming performance and CT such as; students who use block-based programming tools develop their CT skills as much as their programming skills improve (Oluk & Korkmaz, 2016); the variable of academic success of Information technologies can significantly predict CT skills (Durak & Saritepeci, 2018). These variables, which are related to programming performance, can be described as cognitive factors. For CT skills, non-cognitive factors such as self-efficacy and attitude are determinative as well as cognitive factors (Román-González et al., 2018). In the literature, there are studies on non-cognitive factors that support the results of this study. It was found that there is a moderate significant relationship between CPA and CPSE (Çoban, Korkmaz, Çakır & Erdoğmuş, 2020); and a moderate significant relationship between CT and CPSE (Durak, Yılmaz, & Yılmaz, 2019) and CPSE predicted CT significantly (Avcu & Ayverdi, 2020). The importance of the data obtained within the scope of this study is that the CPSE and CPA variables, which come to mind as the non-cognitive factor when it comes to CT, are considered together for the middle school students, and this study reveals the order of their importance.

When CPSE levels were analyzed, it was seen that both students who took IT&S course and those who did not take had moderate self-efficacy. Although both groups seem to be at the same level, comparisons in terms of their averages showed that the students who took the course had higher self-efficacy statistically significantly. The concept of self-efficacy can be explained as “the individual's perception of the capacity to perform the required effort to perform a particular task” (Bandura, 1997) and there is a mutual relationship with the performance variable (Lishinski, Yadav, Good & Enbody, 2016). Prior programming experience is an important variable for CPSE (Mazman & Altun, 2013; Kittur, 2020). Effect of programming experience on self-efficacy was revealed, after four weeks of STEM-supported training (Feldhausen, Weese, & Bean, 2018) or for a period of algorithm and coding training (Mıhçı-Türker & Pala, 2020; Okal, Yıldırım, & Timur, 2020). Considering the situation that people think that computer programming is complex and difficult (Yükseltürk & Altıok, 2017), it is an expected result that students who did not take IT&S course before and therefore did never perform computer-programming performance would have lower CPSE perceptions.

When the CPAs of the students were examined, it was seen that both the students who took IT&S course and those who did not had very close averages. Both groups have moderate CPA. There is no statistically significant difference according to the comparisons over their averages. In the literature, there are studies that demonstrate that students' programming attitudes have increased as a result of the programming instruction carried out with computer programming tools such as mBlock and Scratch working with drag and drop logic (Günbatar & Karalar, 2018; Yükseltürk & Altıok, 2017). In addition, there are studies revealing that tools such as scratch used in programming instruction do not provide similar attitude development with the traditional method (Çetin, 2016) and that the attitudes of both gifted and normal students towards computer-aided coding are similar (Toklu, 2019). Based on these results, it is thought that CPA can develop at the end of the instruction, which is carried out either with traditional method or with various programming tools. However, it has been demonstrated that people who are interested in computer programming for many years do not differ from those who are new in this business in terms of attitude (Günbatar, 2018).

Attitude is the mental state in which the individual pre-chooses the behaviour in the face of certain situations (Gagne, 1985). Attitudes guide human behaviour, and they constitute an important place within the scope of affective domain learning (Balaban-Salı, 2006). There are several levels of affective domain learning, ranging from a basic level, such as having knowledge about a particular situation, to a complex situation ranging from creating a life philosophy (Krathwohl, Bloom & Masia, 1964). Getting information about the attitude of an individual about a certain subject directly reveals his attitude. However, it is not possible for everyone to present objective information, and different approaches to obtaining information (such as collecting information from others, reviewing records) can also be used (Balaban-Salı, 2006). In the literature, it is mentioned that there is a development in programming attitude after basic programming training, and there is no difference between those who are interested in programming for many years and those who are new to this business in terms of programming attitude. However, the number of studies on computer programming and student attitudes is not high (Çetin & Ozden, 2015). Considering the complex nature of the attitude, it is recommended to carry out new studies (using different approaches to obtain information if possible).

When CT measurements are examined, students who took IT&S course had high level of CT skills; those who did not take the course were found to be at a moderate level. Comparison results based on averages also support this level differentiation. Students who took the course have significantly higher CT skills than those who did not take the course. In the process of collecting data within the scope of the study, MoE Board of Education Discipline has recently organized the IT&S course curriculum at middle school level. Therefore, the data were collected during the first applications of the renewed curriculum. Within the scope of the specific objectives of the renewed curriculum, main expectations are: The students' acquisition and development of problem solving and computational thinking skills; To be able to express their verbal and visual expressions by developing an understanding of algorithm design; To choose the appropriate programming approach to solve problems and apply them; Creating technical knowledge in programming; Ability to use at least one of the programming languages; Developing innovative and original projects for the solution of the problems encountered in daily life (problems faced by elderly and disabled people, etc.) ". In this context, a "Problem solving and programming" unit covers a total of 72 hours, covering 50% for the 5th and 6th grades and 44% for the 7th and 8th grades. It is planned to achieve the above-mentioned special objectives within the scope of this unit (MoE, 2018a; MoE, 2018b). The CT skills difference between the students who took IT&S course and those who did not take the course, and the problem solving and programming unit, which is given an important place within the scope of the curriculum, are an indication that the new curriculum can achieve its purpose.

Students who took the course and those who did not differ according to the common effect of CPSE, CPA and CT skills. In other words, the students who took this course make progress cumulatively in terms of these three variables compared to those who did not. As a result of their study, Román-González, Pérez-González, Moreno-León, & Robles (2018) emphasized non-cognitive factors related to CT. They provided suggestions for training policies for the development of CT skills. Accordingly, besides cognitive psychological construct such as problem-solving skill, non-cognitive factors such as self-efficacy and personality should also be considered. Within the scope of this study, this differentiation obtained with the common effect of self-efficacy, attitude and CT skills confirms the idea of Román-González, Pérez-González, Moreno-León, & Robles (2018). Turkey revealed the importance of the CPSE and CPA variables as well as CT skills to achieve the specific objectives within the scope of the problem-solving and programming unit with the its renewed IT & S course curriculum. Based on these results, especially the importance of problem-solving and programming unit's objectives has emerged. Updated middle school IT&S course curriculum requirements must be accomplished by all institutions in Turkey. Curriculum developers in other countries must recognize the importance of the problem-solving and programming subjects. They should make decisions to equip students with the skills with this context.

References

Avcu, Y.E. & Ayverdi, L. (2020). Examination of the computer programming self-efficacy's prediction towards the computational thinking skills of the gifted and talented students. *International Journal of Educational Methodology, 6*(2). 259-270. doi: 10.12973/ijem.6.2.259

Balaban-Salı, J. (2006). Tutumların öğretimi. In A. Şimşek (Ed.), *İçerik Türlerine Dayalı Öğretim* (pp 133-162). Ankara: Nobel Yayın Dağıtım.

Bandura, A. (1971). *Social learning theory*. New York: General Learning Press.

Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: W. H. Freeman.

Çetin, I. (2016). Preservice teachers' introduction to computing: exploring utilization of scratch. *Journal of Educational Computing Research, 54*(7), 997-1021. doi: 10.1177/0735633116642774

Çetin, I., & Özden, M. Y. (2015). Development of computer programming attitude scale for university students. *Computer Applications in Engineering Education*, *23*(5), 667-672.

Computing at school (2020). *Computational thinking*. Retrieved July 9, 2020 from https://community.computingatschool.org.uk/files/8261/original.pdf

Çoban, E., Korkmaz, Ö., Çakır, R., & Erdoğmuş, F. U. (2020). Attitudes of IT teacher candidates towards computer programming and their self-efficacy and opinions regarding to block-based programming. *Education and Information Technologies*, *25*(5), 4097-4114. doi: 10.1007/s10639-020-10164-w

Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, *116*, 191-202. doi: 10.1016/j.compedu.2017.09.004

Durak, H. Y., Yılmaz, F. G. K., & Yılmaz, R. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Educational Technology*, *10*(2), 173-197. doi: 10.30935/cet.554493

Feldhausen, R., Weese, J. L., & Bean, N. H. (2018, February). Increasing student self-efficacy in computational thinking via STEM outreach programs. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, Baltimore Maryland, USA.

Freankel, J.R. & Wallen, N. E. (2009). *How to design and evaluate research in education* (7th edition). New York: McGraw-Hill.

Gagne, R. M. (1985). *The conditions of Learning and theory of instruction* (4th ed.). New York, NY: CBS College Publishing.

Günbatar, M. S. (2018). Examination of undergraduate and associate degree students' computer programming attitude and self-efficacy according to thinking style, gender and experience. *Contemporary Educational Technology*, *9*(4), 354-373. doi: 10.30935/cet.471004

Günbatar, M. S., & Karalar, H. (2018). Gender differences in middle school students' attitudes and self-efficacy perceptions towards mBlock programming. *European Journal of Educational Research*, *7*(4), 925-933. doi: 10.12973/eu-jer.7.4.923

ISTE (2018). *ISTE standards for students.* Retrieved April 11, 2018 from https://www.iste.org/standards/for-students

ISTE and CSTA (2011). *Operational definition of computational thinking for K-12 thinking operational-definition-flyer.* Retrieved July 16, 2020 from http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn¼2

Keçeci, G., Alan, B., & Kırbağ-Zengin, F. (2016). Educational computer games assisted learning coding attitude scale: validity and reliability study. *Education Sciences*, *11*(3), 184-194. doi: 10.12739/NWSA.2016.11.3.1C0661

Kittur, J. (2020). *Measuring the programming self-efficacy of Electrical and Electronics Engineering students*. Retrieved July 27, 2020 from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9031745

Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2016). Computational thinking levels scale (CTLS) adaptation for secondary school level. *Gazi Journal of Educational Science*, *1*(2).

Krathwohl, D. R., Bloom, B. S., & Masia, B. B. (1964). *Taxonomy of Educational Objectives, Handbook II: Affective Domain.* New York: Longman.

Kukul, V., Gökçearslan, Ş., & Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Educational Technology Theory and Practice*, *7*(1), 158-179. doi: 10.17943/etku.288493

Lishinski, A., Yadav, A., Good, J., & Enbody, R. (2016, August). Learning to program: Gender differences and interactive effects of students' motivation, goals, and self-efficacy on performance. In *Proceedings of the 12th Annual International ACM Conference on International Computing Education Research,* Melbourne VIC, Australia.

Mazman, S. G., & Altun, A. (2013). Programlama–I Dersinin BÖTE Bölümü Öğrencilerinin Programlamaya İlişkin Öz Yeterlilik Algıları Üzerine Etkisi. *Journal of Instructional Technologies & Teacher Education*, *2*(3), 24-29.

MoE (2018a). *Bilişim teknolojileri ve yazılım dersi öğretim programı (5. ve 6. Sınıflar).* Retrieved June 25, 2020 from http://mufredat.meb.gov.tr/Dosyalar/2018124103559587-Bilişim Teknolojileri ve Yazılım 5-6. Sınıflar.pdf

MoE (2018b). *Bilişim teknolojileri ve yazılım dersi öğretim programı (7. ve 8. Sınıflar).* Retrieved June 25, 2020 from http://mufredat.meb.gov.tr/Dosyalar/2018813171426130-2-2018-81Bilişim Teknolojileri ve Yazılım Dersi (7 ve 8. Sınıflar).pdf

Mıhçı-Türker, P. & Pala, F.K. (2020). The effect of algorithm education on students' computer programming self-efficacy perceptions and computational thinking skills. *International Journal of Computer Science Education in Schools. 3*(3). 1-11. doi: 10.21585/ijcses.v3i3.69

Okal, G., Yıldırım, B. & Timur, S. (2020). The effect of coding education on 5th, 6th and 7th grade students' programming self-efficacy and attitudes about technology. *Educational Policy Analysis and Strategic Research, 15*(2), 143-165. doi: 10.29329/epasr.2020.251.8

Oluk, A. & Korkmaz, Ö. (2016). Comparing students' scratch skills with their computational thinking skills in terms of different variables. *International Journal of Modern Education and Computer Science. 11.* 1-7. doi: 10.5815/ijmecs.2016.11.01

P21 (2020). *P21 Frameworks & Resources*. Retrieved February 28, 2020 from http://www.battelleforkids.org/networks/p21/frameworks-resources

Pallant, J. F. (2007). *SPSS survival manual: A step-by-step guide to data analysis with SPSS* (3rd edition). New York: Open University Press.

Prensky, M. (2001a). Digital natives, digital immigrants. *On The Horizon, 9*(5).

Prensky, M. (2001b). Digital natives, digital immigrants part 2: Do they really think differently?. *On The Horizon*, *9*(6).

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in Human Behavior*, *72*, 678-691. doi: 10.1016/j.chb.2016.08.047

Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, *80*, 441-459. doi: 10.1016/j.chb.2017.09.030

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142-158. doi: 10.1016/j.edurev.2017.09.003

Toklu, E. (2019). *Game Desing and Coding Training For 7-11 Year Old Gifted Students – Kodu Game Lab Example* (Unpublished master's thesis). Uludağ University, Bursa-Turkey.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*, 33-35.

Yadav, A., Stephenson, C. & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, *60*,(4), 55-62. doi: 10.1145/2994591

Yükseltürk, E., & Altıok, S. (2017). An investigation of the effects of programming with scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*. *48*,(3). 789-801. doi: 10.1111/bjet.12453

Exploring the Development of Primary School Students' Computational Thinking and 21st Century Skills through Scaffolding: Voices from the Stakeholders

Volkan Kukul[1]

Recep Çakır[1]

[1]Amasya University

DOI: https://doi.org/10.21585/ijcses.v4i1.84

Abstract

Computational thinking (CT) has become a skill that is taught starting from an early age with its increasing popularity. In addition, the opinion that CT is related to other 21st century skills finds its place in the literature. The main purpose of this study is to identify the contributions of scaffolding-based game programming activities to students' CT and 21st-century skills. In line with this purpose, the study was designed as a qualitative case study. The participants of the study consisted of 16 primary school students, 10 pre-service ICT Teachers, and 2 primary school teachers. The research results reveal that game programming has a positive effect on students' CT skills, 21st century skills and some psychometric variables like self-confidence and motivation. In future studies and implementations, educators may support their instructions of programming through different scaffolding strategies. In addition, it should be taken into consideration that students can become innovative designers with content that they find interesting.

Keywords: Computational Thinking, 21st Century Skills, Scaffolding, Game Programming

1. Introduction

In modern societies, people need diverse skills to maintain their lives more conveniently. These skills can vary according to the approaches of institutions and researchers, yet there are common points between the skills that are defined. The common points involve creative, innovative, collaborative and computational thinking skills (PFC, 2018; van Laar, van Deursen, van Dijk, & de Haan, 2017; Chalkiadaki, 2018). Teaching these skills requires updating education programs along with the advancing technology and enabling students to perform the necessary activities. Coding skills and programming education have taken place among the rapidly growing trends with the advancing technology.

Coding skills are known as the skills that emerge as an outcome of reasoning and they are given a place in the study programs of different countries (Angeli et al., 2016; European Commission, 2016). In addition to computer programming skills, students are also taught skills to find different solutions to problems and find out practical answers (Yukselturk and Altiok, 2016). The aim is to raise number of students who can use technology well and develop their own technology.

Recent studies have shown that children can learn coding even at early ages and coding education improves the skills that students should have in the 21st century (Bermingham et al., 2013; Zimmerman, 2007). Cetin (2012) has shown in his study that was carried out with 5th-grade students that teaching coding to students had a positive impact on their problem-solving skills. Another study conducted by Lin and Kuo (2010) employed the robot kits for education in teaching the programming language course at primary and secondary school levels, and their study reported that students can gain programming skills with these kits and acquire basic architectural knowledge. Lindh and Holgersson (2007) investigated the impacts of robotic toys (legos) on mathematics and problem-solving skills in their study, yet did not find a significant difference between the control and experimental groups. However, the study revealed that students who enjoyed problem-solving activities had higher levels of achievement, and students who had lego-logo training became more successful in the following years.

In the initial phases of the implementation, students who are new to programming education had considered programming education as a difficult process (Genc and Karakus, 2011; Gomes and Mendes, 2007), yet, recently, programming education has become more entertaining for students with ready codes and visualized tools (code hour, code org, scratch, etc.) and perceived as less difficult by the learners. One of the approaches adopted to make programming education an entertaining process for students is enabling them to design their own games. Programming games is reported as an effective method to teach technical skills to students (Kafai and Burke, 2015; Hainey, Baxter, & Ford, 2019). Kafai and Burke (2015) argued that coding education can reach to a larger audience and become more successful with games and suggested to teach coding via games. Furthermore, in the literature, game programming is defined as one of the most common approaches to improve computational thinking skill (Weinberg, 2013; Hainey et al., 2019; Allsop, 2019) which is among the 21st-century skills (ISTE, 2016). Therefore, this aim of this study is to analyze the impacts of game programming on 21st-century skills and computational thinking skills from the perspectives of all relevant stakeholders.

2. Theoretical Framework

2.1 21st Century Skills

Researchers have emphasized that individuals should be equipped with different skills to be able to accommodate themselves in knowledge societies. Digital literacy lay at the core of these skills (Perkovic and Settle, 2010). Considering the globalization process and the rapid development of ICTs in our lives, improving digital skills has become an imperative (Van Laar, Van Deursen, Van Dijk, & De Haan, 2017). According to Perkovic and Settle (2010), to catch up with 21st-century societies, individuals should be computer literate and possess computer fluency ve computational thinking skills. On the other hand, for various organizations, 21st-century skills are not only based on digital literacy and they should be viewed from a broader perspective. Partnership for 21st Century Skills analyzed the skills that individuals must have in this century under three main themes in the framework published in 2007 (Partnership for 21st Century Learning, 2007). In framework, the first theme is Learning and Innovation Skills (creativity and innovation, critical thinking and problem solving, communication and collaboration), the second theme is Information, Media and Technology Skills (information, media and ICT literacy), and the third theme is Life and Career Skills'tir (flexibility and adaptability, initiative and self-direction, social and cross-cultural skills, productivity and accountability, leadership and responsibility). International Society for Technology in Education (ISTE) has also established 21st century standards for students in different periods. First of all, in 1998, they improved individuals' use of technology based on their learning, afterward they developed standards for using technology for learning, and finally, they developed standards for transformative learning with technology in 2016. The recent standards include the skills as follows: Empowered Learner, Digital Citizen, Knowledge Constructor, Innovative Designer, Computational Thinker, Creative Communicator and Global Collaborator (ISTE, 2016).

2.2 Computational Thinking

The concept of computational thinking (CT) was used by Papert for the first time in the 1980s and it was brought up again by Jeannette Wing in 2006 (Kalelioglu, Gulbahar, & Kukul, 2016; Kukul & Karatas, 2019; Roman-Gonzalez, Perez-Gonzalez, & Jimenez-Fernandez, 2017). After that date, many countries have updated their curriculums to improve students' computational thinking skills (Mannila et al., 2014). For a long time, there was no consensus on the definition of CT (Grover & Pea, 2013; Kalelioglu et al., 2016). The definition of Wing, which made the concept of computational thinking popular again which defines the concept as "involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science" (Jeannette M. Wing, 2006). However, the article which presented this definition also raised many questions. In 2010, Wing has clarified the definition of CT in another study and reformulate it as CT is the "thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (J. Wing, 2010, p.1). The given definitions were not found sufficient to present the structure of CT skill as a psychological variable, and therefore to be evaluated and improved (Roman-Gonzalez, Perez-Gonzalez, Moreno-Leon, & Robles, 2018). Therefore, this need has tried to be fulfilled with studies on definitions towards implementation (Kalelioglu et al., 2016) and evaluation (Brennan & Resnick, 2012; Dolgopolovas, Jevsikova, Savulionienė, &

Dagienė, 2015; Kukul & Karatas, 2019; Román-González, 2015; Allsop, 2019; Hainey et al., 2019). According to Weinberg (2013), the updates in the curriculum based on teaching CT skills and the most common approach can be addressed under four main headlines. These approaches include block-based programming environments (Scratch, AppInventor, Alice, etc.), game programming (KoduGameLab, etc.) or Robotic (Arduino, Lego, etc.) applications, computer-less applications (CS Unplugged, Bebras, etc.) and CT with other disciplines (STEM, etc.). In the scope of this study, a game programming approach, which is a widely used approach in teaching the CT skill, was used.

2.3 Game Programming and Kodu Game Lab

In the literature, some studies focused on the potential impacts of game programming on CT (Lee et al., 2011), and it is evident that game programming improves CT skills (Garneli & Chorianopoulos, 2018). The meta-synthesis study conducted by Denner, Campe, and Werner (2019) revealed that students improved diverse skills while programming games and also enjoyed the process of programming. Vos, Van der Meijden, and Denessen, (2011) compared two groups of students (aged 10-12 years old). One group only played games and the other group created their own games. According to the results, the learning motivation of the students in the group who created their own games is higher than the others. The given situation makes game programming a strong alternative for enhancing CT skills. Technology firms are also aware of this fact and have created platforms where children can develop games. KoduGameLab, which was used in the scope of this study, is an example of these platforms.

KoduGameLab is an environment that was developed by Microsoft that enables to program games based on the drag-and-drop system. The system has been designed due to the increasing popularity of teaching programming to young children, as an environment where children can design their own games. The program was introduced as a desktop version and it can run with different operating systems. The tool that has object-oriented programming features (Yukselturk & Uçgul, 2018) also includes libraries consisted of different game environments, and individuals can choose one of the environments and use different components. It is possible to share the game designed with KoduGameLab with others, and also to access other games that are shared. KoduGameLab enables the design of three-dimensional games and in this environment, the games are designed using condition structures frequently. In the literature, several studies are designed by using KoduGameLab for game programming. The experimental study conducted by Akcaoglu and Green (2019) revealed that games developed with KoduGameLab improve students design skills. Fowler and Khosmood (2018) conducted a study that employed KoduGameLab and showed that game programming had a positive impact on students' perceptions of Computer Science and permanent learning abilities. Chiazzese et al. (2018) revealed in their study that game programming had a positive impact on participants understanding of the basic principles of programming.

2.4 Scaffolding

The concept of Scaffolding was defined by Vygotsky for the first time in 1978 (Dennen, 2004). According to Vygotsky, students should be guided by a more talented peer to overcome a problem or fulfil a task (Vygotsky, 1978). From this aspect, the concept can be considered as a strategy that fits to group studies. Scaffolding strategies, which are based on learners' needs, may increase achievement in learning concepts, procedures and metacognitive skills (McLoughlin, 2002). In addition to metacognitive skills, the concept also plays a role in increasing achievement in affective development spheres such as motivation and self-confidence (Dennen, 2004). It can also support students to overcome their feelings of failure (Bean & Patel Stevens, 2002). In principle, various strategies such as modelling, coaching, prompting and questioning are used for scaffolding implementations based on social dialog and interaction (Ge & Land, 2003). It is believed that the concept of scaffolding implemented with these strategies may contribute to CT components. For example, Rogoff (1990) defined scaffolding as a meta-activity that assists learners to separate given tasks into pieces, and this definition can explain supporting learners in 'decomposition' which is among the sub-components of CT.

Considering the age levels and lack of programming experience of the students who participated in this study, implementation of scaffolding strategies with talented and experienced peers on programming, and with pre-service teachers will be useful for students. Feng and Chen (2014) emphasized that scaffolding strategies that will be implemented with learners can enable students to analyse the problem effectively before solving them. From this aspect, scaffolding may also contribute to the abstraction skill that is seen as the most important component of CT (Jeannette M Wing, 2008).

3. Aim of the Study

CT has become a skill that is taught starting from an early age with its increasing popularity. In the literature, scaffolding-based programming education has been emphasized as a frequently used method (Lye & Koh, 2014), in the evaluation of these studies computational concepts such as variables and loops were employed (e.g. Kazakoff & Bers, 2012; Wang & Chen, 2010), yet, computational practices (problem-solving practices during programming activities (Brennan & Resnick, 2012) was not stressed (Lye & Koh, 2014). The research studies conducted on the issue suggested evaluating students in applications/practices that aim to teach the CT skill to students through think-aloud protocol or observation of the student outcomes (Lye & Koh, 2014). The main purpose of this study is to identify the contributions of scaffolding-based game programming activities to students' CT and 21st-century skills. In line with this purpose, answers were sought to the following research questions:

- Considering the CT components, which components have been affected by the contributions of game programming activities?
- What were the contributions of game programming activities to students' 21st-century skills? In the case that there were contributions, which skills are affected by these contributions and in which way they were affected?

4. Method

The purpose of the research study is to identify how game programming may affect students' learning of CT and 21st-century skills. In line with this purpose, the study was designed as a qualitative case study. Creswell (2007) defined a case study as “a qualitative approach in which the investigator explores a bounded system (a case) or multiple bounded systems (cases) over time, through detailed, in-depth data collection involving multiple sources of information”. The case that was chosen in the scope of this study was the game programming activities organized by STEM and Coding Student Club and performed by students via KoduGame Lab as a part of the end of year activities of a primary school located in Amasya province. In the scope of this activity, students programmed their games in the framework of the selected theme under the guidance of Information Technologies teachers. The data sources of the study included the interviews conducted with the students who participated in the activity, with the teachers who were in charge of the student clubs, and with pre-service Information Technologies teachers who guided students. The observations made by pre-service teachers were also used in this process. The ‘computational thinking evaluation framework’ developed by Brennan and Resnick (2012) constituted the theoretical basis for the interview questions, given that the study aimed to evaluate students' CT and 21st-century skills via programming.

4.1 Participants

The participants of the study consisted of 16 primary school students of a public school located in Amasya province who participated on the activities organized by STEM and Coding Student Club, two teachers who were in charge of the students club, and 10 pre-service Information Technologies teachers who were in the third year of study. The distribution of the students according to the grade-levels showed that 4 students were second-grade students, 6 students were third-grade students, and six students were fourth-grade students. Some of these students previously joined coding activities organized by the club in the previous year and therefore they were experienced in coding, however, the majority of the students participated in club activities for the first time. On the other hand, the teachers who were in charge of the clubs were class teachers and had teaching experience over 20 years. The pre-service Information Technologies teachers who guided students participated in this study voluntarily on the scope of the practical social work course.

4.2 Content

Due to the inadequate physical capacities of their school, students used the computer laboratories of Amasya University for the club activities. In the first phase, students were separated into groups by the teachers who were in charge of the clubs, and afterward, at least one teacher was assigned to each group to guide students. Before introducing the university environment to the students, basic knowledge on the KoduGameLab interface and the logic of the system was provided to the students via online education programs. In the university environment, the students were asked to design a game related to the ‘space’ theme that was defined by the teachers. Students coded their games for 9 weeks and two hours were allocated per week for the coding activities. As a result of the activities, the students presented their games within Amasya University. Some of the games that were created by students were selected to take part in a national competition. The pre-service teachers who guided students in this process used the scaffolding method when students had difficulties, and also observed the students on a weekly basis during the coding process and filled an observation form for each student.

4.3 Data Collection Tools

The data were collected from different data sources as a requirement of the case study method. In this way the findings could be analysed in detail. The data of the study were collected from four different data sources. The Figure presented below visualizes the data that were collected via data collection tools.

4.4 Interviews with Teachers

One-to-one interviews, that are commonly used in educational research studies (Creswell, 2012), were employed in this study to conduct interviews with teachers. The interviews were performed in the framework the semi-structured interview form of questions developed by the researchers. More detailed data can be collected in semi-structured interviews through spontaneous questions which provides flexibility to researchers (Creswell, 2012). The questions provided in the semi-structured interview form were formed based on the ‘computational thinking evaluation framework’ that was developed by Brennan and Resnick (2012). In addition to this framework, questions that aimed to reveal the changes that occurred in the 21st century skills that students had were included in the form. Moreover, the spontaneous questions were used to obtain more data. In this context, the following interview questions were addressed to the teachers during the interviews:

- How did students begin the game developing process?
- What kind of path did they follow to develop their games?
- How did students identify the data that can be useful for them while transferring the data to their games? (Abstraction)
- Did students program the games in modules? (Decomposition)
- Did students recognize the dysfunctional points in game codes on their own? What kind of solutions did they find? (Evaluation – Testing)
- Were students capable of recognizing the dysfunctional points in game codes and solving the problem? (Generalisation)
- Which skills of students have been improved as a result of coding their own games? (21st Century Skills)

4.5 Focus group interviews with students

Focus group interview is a method that can be used to receive the common views of a specific group on a case or event. Focus group interview is considered as a data collection method that is effective in the case that the interaction among the interviewees is high and when this interaction is expected to create more data (Creswell, 2012), also in the case that interviewees are working together on a specific task. Therefore, the interviews conducted with students were planned as a focus group interview.

4.6 Focus group interviews with pre-service ICT Teachers

Focus group interview method was used to receive the opinions of the pre-service Information Technologies teachers who were guided students while they were programming the games. Focus group interview method was chosen to reveal and compared the situation occurred in different groups given that pre-service teachers were performing the same task on different groups. The following questions were addressed to the pre-service teachers:

- How did students begin the game developing process? What kind of paths did they follow to develop their games?
    - How did students identify the data that can be useful for them while transferring the data to their games? (Abstraction)
    - Did students program the games in modules? (Decomposition)
    - Did students recognize the dysfunctional points in game codes on their own? What kind of solutions did they find? (Evaluation – Testing)
    - Were students capable of recognizing the dysfunctional points in game codes and solving the problem? (Generalisation)
- Which skills of students have been improved as a result of coding their own games? (21st Century Skills)
    - Do you think that the students can work in harmony with other students attending another school? (Collaboration)
    - How do you evaluate the students on creating ideas to design their own games? (Creativity)

4.7 Observation forms filled by pre-service ICT Teachers

Computational Thinking Self-Efficacy Scale developed by Kukul and Karatas (2019) was used as the baseline to observe students within the frame of computational thinking in the programming process. The observation form aimed to observe six different behaviours. The items included in the observation form are presented in Table 1. Each behaviour had a column for explanation where observers could use to take notes about the behaviour. These explanations provided detailed information about the observed or non-observed behaviour. The behaviours included in the observation form were scored as 1 or 0 which referred to the situations of being present or absent to control the development of the behaviours, and students' average scores for the observation form were calculated. In this way, the impact of each week on the behaviours was aimed to be observed using average scores.

Table 1. Items of the Observation Form and Related Concepts

Observed Behaviour	**Related Concept**
The student is able to create the steps necessary for solving the problem.	Algorithmic Thinking
The student is able to identify the dysfunctional points in the steps specified for solving the problem.	Evaluation or Testing/Debugging

The student is able to break up the problem into sub-problems.	Decomposition
The student is able to use previous experiences to solve the problems.	Generalization
The student is able to overlook the unnecessary information/data while solving the problem and focus on the solution.	Abstraction
The student is able to visualize the data to be used for solving the problem or the solution itself.	Abstraction

4.8 Data Collection and Analysis Process

In the data collection phase, different data sources were employed due to the nature of the case study method. In the first place, the contributions of game programming in computational thinking skills aimed to be identified through three different focus group interviews conducted with students. Each focus group interview took around 15 minutes and both students and their parents were informed about the objectives of the interviews. Students were also informed that they can opt-out from the interviews. After receiving the required consents, each interview was recorded.

Afterward, semi-structured interviews were conducted with the participation of the teachers who were in charge of the student clubs, who also had an opportunity to observe the students in different environments apart from the club activities. The interviews conducted with two teachers lasted for 70 minutes in total. Finally, a focus group interview was conducted with three pre-service Information Technologies teachers who observed the students during the game coding activity. The interview took around 35 minutes.

The interviews were analysed in three phases including: (1) organization of the data, (2) coding and reducing data (3) representing data with tables and discussion (Creswell, 2007). In the first phase, the data were transformed into text files from sound files and each participant was given a nickname. In the second phase, the data were separated into sections using the constant comparative method (Glaser, 1965) and the data sections were continuously compared according to their similarities and differences. To name the data sections, both the framework developed by Brennan and Resnick (2012) and the concepts of computational thinking included in the literature were considered. In the analysis of the data, opinions of different data sources were compared, and themes were created. In the final phase, the codes and themes were presented in tables and discussed within their own contexts. As described by Creswell (2007), these three phases proceeded interactively in a spiral structure.

To ensure the credibility of the findings, the principles proposed by Lincoln and Guba (1985) were adopted. The data were also analysed by another researcher to avoid the prejudice of the researchers towards the research problem, and a consensus regarding the results was reached after the discussions with researchers based on negotiated coding approach developed by Garrison et al. (2006). The obtained codes and themes were explained detail within their own

context and the transferability of the findings to other contexts was ensured. Furthermore, the data obtained from different data sources, apart from the interviews, were compared with the findings of the interviews. In this way, the findings were supported and discussed in detail.
In addition to the interviews, the pre-service teachers were asked to fill observation forms each week for students in their groups. The observation forms filled by the pre-service teachers were analysed and weekly behaviour changes were identified. Furthermore, observation notes of each pre-service teacher were used to support the interview data. Given that each pre-service teacher observed another student, no comparisons were made between the observation scores, and the improvement of each observation item was followed on a weekly basis.

5. Findings

5.1 Game Programming and CT

To identify the impact of game programming on students' CT skills, interviews were conducted with all relevant stakeholders. The concepts concerning CT that were changed as a result of the interviews were presented in Table 2.

Table 2. Game Programming and CT Components

	According to the Teachers	**According to the Pre-Service Teachers**	**According to the Students**
Data collection	X	X	X
Abstraction	X	X	X
Testing - Debugging	X		X
Decomposition	X	X	X
Generalization	X	X	

The Table shows that teachers who were in charge of the clubs and pre-service teachers shared similar opinions. However, given that pre-service teachers observed different groups, there were differences regarding the impact on students' CT skills. For example, students in a group attended the club activities two years in a row, and almost all of the changes regarding the CT components were present for this group. However, this change was not observed for students on another group.
In analysis of the CT skills, based on the paths that were chosen by the children while programming their games, the teachers stated that the students began to the process by collecting data.

> *"We define the theme of the game. Afterward, the research process about the theme takes place. We turn on the projector in the laboratory together. We make a detailed research about the theme. At the end of the day, we give tasks to students about the theme. Everyone make research on a specific topic. In our next meeting, they present their research" (Teacher 1).*

One of the pre-service teachers shared a similar opinion and stated that students began the process with collecting data.

> *"The topic was about the space and therefore we needed to collect primary data. At first, we collected data about space." (Pre-Service Teacher 3)*

Students also stated that most of the groups began the process with data collection. The interviews conducted with students showed that teachers' guidance towards making research was effective. On the other hand, the results of the interviews showed that some of the groups avoided the guidance and did not make any research.

> *"At first, we made computer-based research to decide where our work will take place." (Student 2)*
>
> *" We started by creating the background. Afterward, we decided on the character and started to write the codes" (Student 4)*

However, it drew attention that students who began the process with making research were highly interested in the selected theme for the game, which was 'Space'. The expressions of the same pre-service teacher revealed this situation clearly:

> *"Some of these children are very interested in space. When they had a chance to design a game related to space, they felt the need of making more research and to acquire more information. They read magazines and read about space. They even watch the movie called Interstellar. In the end, they were asking about going there, what they can see in Titan, and if they can make different research projects?" (Pre-Service Teacher 3)*

On the other hand, students in the other group were not interested in the theme 'space' and therefore, they had a motivation level at the beginning of the programming activities. This situation was expressed by a pre-service teacher as follows;

> *"... we imposed the space theme for the ones who are not interested in. In the end, we got games irrelevant to space. They had prejudices towards the selected theme. We created a Head Ball game, it caught students' attention and they all wanted to do it. They are more comfortable when they developed games related to themes they choose. They should be free to choose their themes. " (Pre-Service Teacher 3)*

About the problem in the selection of the theme, another pre-service teacher stated that;

> *"They were not familiar with the theme. The theme chosen by the teacher was not interesting for them and they had no information about it. The internet-based research was not enough at that point." (Pre-Service Teacher 1)*

Abstraction is among the key components of the CT components (J. Wing, 2010; Jeannette M Wing, 2008). Teachers' statements revealed that students were capable of making an abstraction.

> *"Although students did not take the data that can use from during the research process by saying 'I can use it later'', I saw that they were influenced by the research process while they were designing their games." (Teacher 1)*

The student expressed that they made abstraction in different dimensions.

> *"We take some parts of the research and used them in the game. We designed it on paper." (Student 2)*
>
> *"Our research helped us to figure out what kind of a world we want to design and the research process was useful for that." (Student 6)*

However, this situation was not valid for every student. One of the pre-service teachers expressed that;

> *"Even we have just gained the habit of talking about the planning phase of a project in detail. The fourth-grade students do not have this habit. They want to insert all objects in the program in the game nut they don't do it with a certain logic. For example, we made the planning on paper at first. We write down our thoughts on paper. " (Pre-Service Teacher 1)*

The teachers' expressions showed that students were in the phase of testing and error debugging processes related to their games. However, pre-service teachers did not mention such a skill. The interviews made with students showed that some of the students made debugging, and some of them could see the error yet did not understand the source of the mistake and spend hours for trial and error.

> *"After designing the game they combine their modules and they play the game for an outsider. They correct their games according to the feedback. As they design their own games, they are very aware of the track of the game. Therefore, they can correct their own mistakes. They know where the error is and they also know the reason for it better than others." (Teacher 2)*
>
> *"We placed the stones and we wanted to blow up all of them but they didn't. Then I wanted to see why they didn't blow up but we find anything. We put a lot of effort but then we found the reason and solved the problem. "(Student 3)*
>
> *"We had a difficult time while coding. When we faced a difficulty we discussed with the group members and solved the problem." (Student 8)*
>
> *"We made a lot of mistakes in the coding phase. We corrected them through the trial and error method." (Student 7)*

During the activities, students were working in groups and they had cooperation within the groups. This situation contributed to their skills to break the problem into sub-problems. The given situation was realized both by the pre-service and teachers. Some of the students also stated that they separated the tasks into different sections.

> *"Of course they break the game into pieces. Everyone is working on a section. In this way, they could show their talents better and easily in their own fields. Afterward, they presented their results to their friends. They worked in a way that 2-3 students could design a game." (Teacher 1)*
>
> *"Each student got a task and started to work" (Student 2)*
>
> *"…we gave students individual tasks. They had to complete 3 sections." (Pre-Service Teacher 3)*

The teachers stated that students' generalization skills were improved through game programming activities.

> *"They can absolutely develop ideas. They can express opinions like this is missing here or this code is absent, so we need to add it. They can absolutely generalize what they have learned by considering other problems. The children are very good at using KoduGame. Therefore, I don't think that they will have difficulties if they use scratch" (Teacher 2)*

Pre-service teachers believed that the given situation is all about having experience. They expressed that the group that was joining programming club activities previously has enabled them to generalize their knowledge to new learning situations and programming problems very easily. On the other hand, the students in the other group did not have such kind of generalization skills. Given that the theme did not draw the attention of the students in terms of improving generalization skills, it was believed that students would not be capable of doing it. However, related to the generalization skills of students, the advisor pre-service teacher of the experienced group said that;

> *"4th and 5th-grade students were saying 'we can handle this' and they could even code the parts which I found difficult. They started to code directly." (Pre-Service Teacher 3)*

5.2 Observation of CT

Figure 1 shows the improvements occurred in students' skills based on the average scores calculated for each item in the observation form on weekly basis according to the observations made by pre-service teachers.

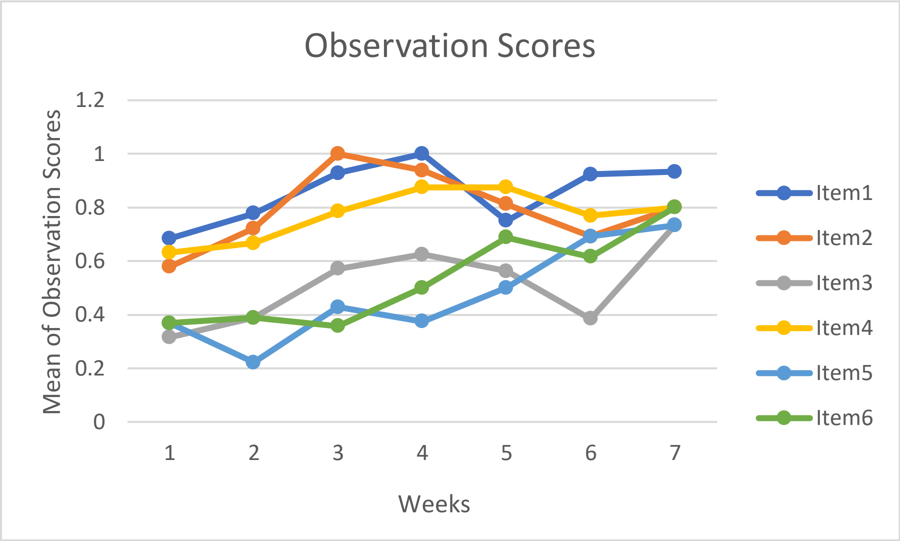


Figure 1. Observation Scores

The Figure shows that skills that were observed at the lowest level starting from week one were students' abstraction and decomposition skills. The results showed that other skills could be observed more frequently. Considering the whole CT skills, it is seen that students' CT skills showed an increase in scaffolding-based activities. These results support the interviews made with the stakeholders. Furthermore, although not all of these behaviours showed a regular increase, it is possible to say that there were a difference and improvement in all behaviours between the first and the last week.

The separate analysis of these behaviours showed that students' algorithmic thinking skills were increased between the first and the last week. However, it was the most frequently observed behaviour in students at the beginner level, and therefore the increase was relatively lower in the last week. In this case, the students with programming experience might have an impact. Furthermore, students' algorithmic thinking skills might be improved in different courses and therefore the level of students was found high at the beginner level. Evaluation or Testing/Debugging item showed that there was an increase as in the case of algorithmic thinking, however, the beginner level was high. It can be argued that the scaffolding activities performed by the pre-service teachers were effective for students to debug the errors and make evaluations.

The analysis of the third item showed that decomposition skill was the least observed skill in the first week. This situation can be explained by the fact that students did not face such kind of problems before. In addition, the lack of problem-solving skills might play a role in this situation. As time progressed, the skill also had been improved and the given situation can be explained with the impact of the scaffolding activities on decomposition skills as mentioned by Rogoff (1990).

The results also showed that there were an improvement in students' generalization skills, however, the increase observed in this skill in the first week was relatively lower. The skill of students that showed the highest improvement is abstraction skill. From this aspect, it can be argued that scaffolding-based game programming activities help to improve abstraction skill. The abstraction skill, which was observed at considerably low levels, in the beginning, caught up with other skill levels in the last week. Given that the abstraction skill was among the key skills for CT (Jeannette M Wing, 2008), it can be argued that scaffolding-based game programming activities made a contribution to the improvement of CT.

5.3 Game Programming and 21st Century Skills

The study aimed to examine whether game programming had an impact on students' 21st-century skills, besides their CT skills. At this juncture, the 21st Century Skills that was specified by International Society for Technology in Education was used as a framework. The study aimed to identify which skills of students that were presented in the framework showed an improvement in this process.

Table 3. Game Programming and 21st Century Skills

	According to the Teachers	**According to the Pre-Service Teachers**	**According to the Students**
Innovative Designer	X	X	X
Knowledge Constructor	X	X	
Creative Communicator	X		X
Global Collaborator	X	X	X
Computational Thinker	X	X	X

Other Variables			
Motivation	X	X	
Self-Confidence	X		X

The teachers expressed that many 21st Century Skills of students who joined the club activities were improved. The teachers also stated that students were able to come up with creative ideas. They expressed their opinions as follows;

> *"…when were with them in the coding process they design the space and they create a story within that. There are 16 children and they work on STEM on Wednesday and on coding on Friday. There is a considerable improvement in their creativity" (Teacher 1)*

On the other hand, pre-service teachers did not describe the situation from a very positive perspective. The main reason for this situation is the fact that the theme of the games which will be designed by the students was chosen by a teacher. This situation might hamper students to produce original outcomes. Furthermore, a pre-service teacher argued that there is a gender-based difference in game programming activities. According to the pre-service teacher, girls produced fewer creative ideas compared to boys.

> *"Except one group, students could not produce original outcomes. I think the inability of the teacher who was in charge of the club to plan the process played a role in this process. " (Pre-Service Teacher 1)*
>
> *"There is a game which is already existed, our students wanted to program the same game." (Pre-Service Teacher 2)*
>
> *"Three of our students were girls. I think that girls are less capable of thinking of games and creating them. They think very simply, probably I am also like that. Boys play more games; maybe too often, and therefore they are better in that. Not in the programming phase, but I think there is a gender-based difference specifically in creating games. Boys are really successful in creating them. Girls generally play with dolls so they think more simply while creating games. They combine two-three objects and then they leave it. " (Pre-Service Teacher 1)*

The interviews made with the students showed that their opinions were similar to the opinions of the pre-service teachers. The students who gained experience in the previous year emphasized that their creativity was enhanced. However, this situation cannot be generalized to all students.

> *"My creativity has been widened. I started to help people. For example, I thought about an elderly person who cannot walk and I wanted to design a tool to help that person to walk. " (Student 2)*
>
> *"I have become more creative. Do you know the game Ben10? While I was watching it I said myself 'I can do it' and I programmed the game. Before the club activities I was not thinking about designing games." (Student 1)*

In the research and data collection process which students performed before programming their games they could structure the information required for their games. However, the groups who were interested in the context of the game had a more efficient research process as well. On the other hand, other groups also made a progress under the guidance of teachers and pre-service teachers.

> *"But now they can search for the things they are want to find. They created a YouTube channel. They share their outcome on their channel. " (Teacher 1)*

In the game programming process, the teachers put the most emphasis on the positive impact on students' communication skills. Both teachers highlighted that students' communication skills were considerably improved in the process of developing games and also afterwards.

> *"There was a big difference in their relationship with their friends. It is very obvious in their conversations. Other teachers are very surprised to see them talking in different places. " (Teacher 2)*
>
> *"We can say that children's language skills and expression skills were improved in the coding process. Because they communicate about their experiences. They make presentations and share with us." (Teacher 1)*

According to teachers, students improved communication skills and their efforts in club activities help them to become global collaborators. Similar opinions were also shared by pre-service teachers. Even the students who were not used to work in groups could work with group awareness.

> *"There is cooperation and counselling. It helps students to be less individualistic. These children are not special children selected from their classrooms; they are voluntary children who came here randomly. " (Teacher 2)*
>
> *"They could get rid of their selfish feelings. Now there is no I, but there is we. We are working for us." (Teacher 1)*
>
> *"The students in my group can definitely work with students even they don't know them" (Pre-Service Teacher 3)*

Students also expressed opinions that can support the observations of pre-service teachers and teachers and stated that group work was a positive experience for them. Given that students' perspectives on the positive aspects of the group work were essentially the same, they indicated different points about these positive aspects.

> *"It was better for us that we were just two. Because we came up with different ideas and opinions while designing the game." (Student 2)*
>
> *"It is difficult to be alone, I prefer to be in a team. Because as a team we can make the task with the help of different opinions and in the end we can make a decision. But when we are alone, we have to do it with only one idea" (Student 1)*
>
> *"I think it is better to do it in a group. Everyone comes up with a different idea and I think it is a nice thing. Everyone can have an idea and we realize them, sometimes it creates problems. But we combine the ideas and solve the problem. " (Student 9)*
>
> *"We can add different ideas. Some points remain missing when you work alone. Our friends can let us know about what is missing." (Student 5)*
>
> *"We all have different characteristics. We all have parts that we are good at. It is good to combine them."(Student 6)*

As a result of the impacts on the CT components provided above, it can be also argued that the CT skill has been also improved as a 21st-century skill. In addition to these skills, the teachers particularly stressed the impacts on psychometric variables such as motivation and self-confidence. The opinions of the students supported the expressions of their teachers and they stated that there was an increase in their self-confidence levels.

> *"There is a burst of self-confidence. They are extremely happy. When they are happy they laugh, they gain self-confidence. We are exactly at this point. It happened with the help of the club activities but it reached a peak with the coding activities. They could produce an outcome with club activities but with the coding activities, they found a chance to show it to others. This led to a burst of self-confidence in students. " (Teacher 2)*
>
> *"Their self-confidence levels have been increased when they started to crease something. Developing their own products is an important factor." (Teacher 1)*
>
> *"They believed that someone can be happy to play the games they designed. When other like their games they are more motivated. They try to make a better game." (Pre-Service Teacher 3)*
>
> *"I am more comfortable in group work now. Normally I am very shy but I was not embarrassed while we were presenting the game in the exhibition." (Student 8)*

The given statement of the student can be interpreted as a sign of self-confidence and it can be also considered as an impact of coding on communication skills.

6. Discussion

This research study aimed to analyse the impact of scaffolding-based game programming activity on students' CT and 21st century skills. According to the results of the study, it is seen that some psychometric variables can be affected during the game programming process. In this direction, the findings of the study are summarized in Figure 2.

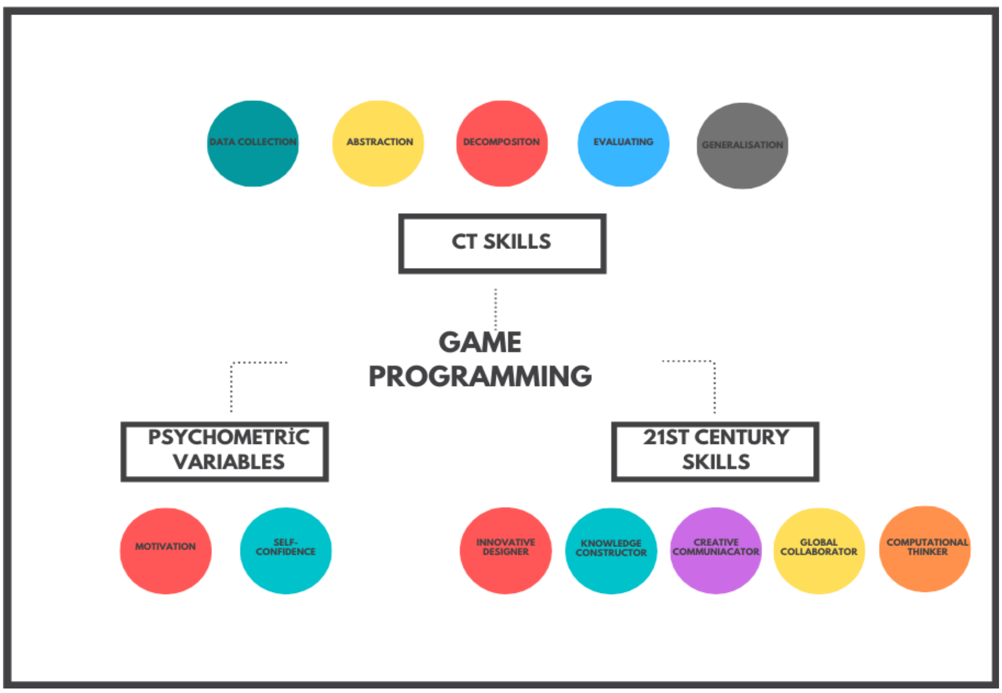


Figure 1. Summary of the Research Findings

The study findings revealed that students began the programming with the data collection process. This skill is also presented in the operational CT definition made by the Computer Science Teacher Association and International Society for Technology in Education (ISTE, 2011). According to the findings, scaffolding performed by pre-service teachers was effective in the data collection process. However, in the case that students are not interested in theme they need to make a research on, scaffolding performed by teachers may remain insufficient. However, it is stated in the literature that scaffolding can be more effective if performed by peers (Thomas, Ge, & Greene, 2011; Vygotsky, 1978). This situation can be investigated with different studies.

Individuals who can program computers are thought to have high level abstraction skills (Wing, 2006). It was tried to help students gain this skill in a shorter time with scaffolding. The scaffolding performed by the pre-service teachers was effective in improving students' abstraction skills. This finding coincides with the findings of the study (Feng, & Chen, 2014), which revealed that programming with scaffolding is better understood and thus contributes to the development of abstraction skill. However, the expressions of the pre-service teachers showed that they had gained this skill very late. Furthermore, the programming experience of the pre-service IT teachers might also lead to an improvement in students' abstraction skills. Therefore, if scaffolding was performed by individuals with low abstraction skills rather than pre-service IT teachers, it might make a negative impact on students.

The research findings showed that students (particularly the ones with programming experience) had generalization skills. However, the increase in the generalization skills was found low, and this situation was associated with the insufficient levels of scaffolding performed by the pre-service teachers. The literature shows that without scaffolding, students face problems in transferring their previous experiences in solving new problems (Lye & Koh, 2014), and they tend towards the trial and error method (Biesta & Burbules, 2003). The same situation is also relevant in the case of debugging, and it was seen that students tried to debug the error instead of trying to find the source of it.

The findings related to the 21st-century skills of the students showed that all stakeholders emphasized the increase occurred in students' innovative designer, global collaborator and computational thinker skills with the game programming process. It is possible to see similar studies in the literature (e.g., Bermingham et al., 2013; Zimmerman, 2007). In addition, it was seen that students had become knowledge constructors and creative communicators, although not all stakeholders shared the same opinion. It seems that game programming attracts students' attention and helps them communicate with each other (Thomas, Ge, & Greene, 2011). The fact that the theme of the game designed by the students was chosen by the teachers, which did not draw their attention, might prevent them from becoming knowledge constructors.

The study findings showed that students' motivation and self-confidence levels were increased after the programming activities and presenting their outcomes. This finding is in parallel with the studies in the literature (e.g., Vos et al., 2011; Bermingham et al., 2013; Garneli, & Chorianopoulos, 2018). The students' designing their own games allowed them to see that they could achieve something. This situation also affects the belief that they can achieve different things in different areas (Guay, Boggiano, & Vallerand, 2001). At this point, teachers should take into account that self-confidence will not increase in all students. For this reason, teachers should support all individuals to increase their self-confidence.

7. Conclusion

One of the limitations of this study was related to the fact that there was no instruction design regarding the club activities performed by the students. In this way, the impact of student club activities on students' CT and 21st-century skills were analyzed. By doing so, the study aimed to make recommendations for teachers who deliver the courses and who are in charge of the student clubs on the implementation process. In future studies and implementations, educators may support their instructions of programming through different scaffolding strategies. Furthermore, experimental studies can be conducted to identify the impacts of different strategies. Teaching programming via real-life problems that can attract students' attention may also enhance the improvement of CT skills. In addition, it should be taken into consideration that students can become innovative designers with content that they find interesting.

Conducting research studies to analyse CT skills of teachers who will perform scaffolding for students, and implementations to improve teachers CT skills may assist to perform a more appropriate scaffolding. In this way, more effective results can be received from students' CT skills.

Considering that students' generalization skills are related to their experiences, designing instruction processes that will enhance their experiences may be useful. Enabling students to face more problems will increase their programming skills and this situation may also improve their CT skills.

The results of the study revealed that students were content with working in collaboration. Educators should take this point into consideration in their instruction designs. In fact, instruction designs that will allow using collaborative work in different phases of instruction, such as evaluation, can be planned.

8. References

Akcaoglu, M., & Green, L. S. (2019). Teaching systems thinking through game design. *Educational Technology Research and Development*, *67*(1), 1-19.

Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International journal of child-computer interaction*, *19*, 30-55.

Bean, T. W., & Patel Stevens, L. (2002). Scaffolding reflection for pre-service and inservice teachers. *Reflective Practice*, *3*(2), 205–218.

Bermingham, S., Charlier, N., Dagnino, F., Duggan, J., Earp, J., Kiili, K., ... & Whitton, N. (2013). Approaches to collaborative game-making for fostering 21st century skills.

In *European Conference on Games Based Learning* (p. 45). Academic Conferences International Limited.

Biesta, G. J. J., & Burbules, N. C. (2003). Pragmatism and educational research. Lanham, MD: Rowman & Littlefield.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking.* Paper presented at the American Educational Research Association, Vancouver, Canada.

Chiazzese, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2018). Engaging primary school children in computational thinking: Designing and developing videogames. *Education in the Knowledge Society*, *19*(2), 63-81.

Creswell, J. W. (2012). *Educational Research Planning, Conducting, and Evaluating Quantitative and Qualitative Research* (4th Edition ed.). Boston: Pierson.

Dennen, V. P. (2004). Cognitive Apprenticeship In Educational Practice: Research On Scaffolding, Modeling, Mentoring, And Coaching As Instructional Strategi es. In D. H. Jonassen (Ed.), *Handbook of Research on Educational Communications and Technology* (Second Edition ed.). New Jersey: LAWRENCE ERLBAUM ASSOCIATES.

Denner, J., Campe, S., & Werner, L. (2019). Does computer game design and programming benefit children? A meta-synthesis of research. *ACM Transactions on Computing Education (TOCE), 19*(3), 1-35.

Dolgopolovas, V., Jevsikova, T., Savulionienė, L., & Dagienė, V. (2015). *On Evaluation of Computational Thinking of Software Engineering Novice Students*.

Feng, C. Y., & Chen, M. P. (2014). The effects of goal specificity and scaffolding on programming performance and self-regulation in game design. *British Journal of Educational Technology, 45*(2), 285-302.

Fowler, A., & Khosmood, F. (2018). The Potential of Young Learners Making Games: An Exploratory Study. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*(pp. 1-9). IEEE.

Garneli, V., & Chorianopoulos, K. (2018). Programming video games and simulations in science education: exploring computational thinking through code analysis. *Interactive Learning Environments, 26*(3), 386-401.

Ge, X., & Land, S. M. (2003). Scaffolding students' problem-solving processes in an ill-structured task using question prompts and peer interactions. *Educational technology research and development, 51*(1), 21-38.

Genç, Z., & Karakuş, S. (2011). Learning by Design: Using Scratch in the Design of Educational Computer Games. 5th International Computer & Instructional Technologies Symposium (ICITS), Elazığ, Turkey. Retrieved From http://goo.gl/KX5Psz

Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE* (Vol. 2007).

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

Guay, F., Boggiano, A. K., & Vallerand, R. J. (2001). Autonomy support, intrinsic motivation, and perceived competence: Conceptual and empirical linkages. *Personality and Social Psychology Bulletin*, *27*(6), 643-650.

Hainey, T., Baxter, G., & Ford, A. (2019). An evaluation of the introduction of games-based construction learning in upper primary education using a developed game codification scheme for scratch. *Journal of Applied Research in Higher Education, 12* (3), 377-402.

International Society of Technology in Education [ISTE] (2011). *Computational Thinking Leadership Toolkit*. Retrieved from https://cdn.iste.org/www-root/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4

International Society of Technology in Education [ISTE]. (2016). 21st Century Standarts for Students. Retrieved from https://www.iste.org/standards/for-students

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

Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, *21*(4), 371-391.

Kukul, V., & Karatas, S. (2019). Computational Thinking Self-Efficacy Scale: Development, Validity and Reliability. *Informatics in Education, 18*(1), 151-164.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., . . . Werner, L. (2011). Computational Thinking for Youth in Practice. *ACM Inroads*, *2*(1), 32-37.

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

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). *Computational Thinking in K-9 Education.* In Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference (pp. 1-29)

McLoughlin, C. (2002). Learner support in distance and networked learning environments: Ten dimensions for successful design. *Distance Education*, *23*(2), 149-162.

Partnership for 21st Century Learning (2007). Framework for 21st Century Learning. Retrieved From:
http://static.battelleforkids.org/documents/p21/P21_framework_0816_2pgs.pdf

Perkovic, L., & Settle, A. (2010). Computational thinking across the curriculum: A conceptual framework. *College of Computing and Digital Media Technical Report*, 10-001.

Rogoff, B. (1990). *Apprenticeship in thinking: Cognitive development in social context*. New York: Oxford University Press

Román-González, M. (2015). *Computational Thinking Test: Design Guidelines And Content Validation*. In Proceedings of EDULEARN15 conference (pp. 2436-2444).

Roman-Gonzalez, M., Perez-Gonzalez, J.-C., & Jimenez-Fernandez, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72*, 678-691.

Roman-Gonzalez, M., Perez-Gonzalez, J.-C., Moreno-Leon, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, *80*, 441-459.

Thomas, M. K., Ge, X., & Greene, B. A. (2011). Fostering 21st century skill development by engaging students in authentic game design projects in a high school computer programming class. *Journal of Educational Computing Research*, *44*(4), 391-408.

Van Laar, E., Van Deursen, A. J., Van Dijk, J. A., & De Haan, J. (2017). The relation between 21st-century skills and digital skills: A systematic literature review. *Computers in Human Behavior*, *72*, 577-588.

Vos, N.H., van der Meijden, H. and Denessen, E. (2011), "Effects of constructing versus playing an educational game on student motivation and deep learning strategy use", *Computers & Education, 56* (1), 127-137.

Vygotsky, L. (1978). Mind in Society: Development of Higher Psychological Processes (Cole M., Jolm-Steiner V., Scribner S., & Souberman E., Eds.). Cambridge, Massachusetts; London, England: Harvard University Press. doi:10.2307/j.ctvjf9vz4

Yükseltürk, E., & Üçgül, M. (2018). Blok Tabanlı Programlama. In Y. Gülbahar & H. Karal (Ed.), *Kuramdan Uygulamaya Programlama Öğretimi.* Ankara: Pegem Akademi.

Wang, L. C., & Chen, M. P. (2010). The effects of game strategy and preference-matching on flow experience and programming performance in game-based learning. *Innovations in Education and Teaching International*, *47*(1), 39-52.

Weinberg, A. E. (2013). *Computational Thinking: An Investigation Of The Existing Scholarship And Research.* [Unpublished doctoral dissertation]. Colorado State University,

Wing, J. M. (2006). Computational Thinking. *Communıcatıons of the ACM, 49*(3), 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725.

Wing, J. (2010). *Computational Thinking: What and Why?* https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Zimmerman, E. (2007) Gaming Literacy: game design as a model for literacy in the 21st century. Harvard Interactive Media Review, *1*(1), 30-35.