Journal of Individual Differences in Education, 2021, 3(2), 97-112

Journal of Individual Differences in Education
EDUGARDEN

Improvement of Pre-Service Teachers' Computational Thinking Skills through an Educational Technology Course

Ebru Albayrak[1] and Şule Yılmaz Özden[1]

[1]*Sakarya University, Faculty of Education, Computer Education and Instructional Technologies Department, Sakarya, Turkey*

ABSTRACT

This study examines the improvement of pre-service teachers' computational thinking skill levels through an educational technology course redesigned within the computational thinking context. 27 pre-service teachers from the Literacy Education Program enrolled in the Instructional Technologies and Material Development course in a public university in Turkey. Pre-service teachers engaged in some structured activities throughout the course and they were asked to complete a final project. Pre and post-survey results showed that pre-service teachers' algorithmic thinking skills and computational thinking skills in general were improved after the course. Analysis of final projects also showed that pre-service teachers were able to use their problem solving, algorithmic thinking, and collaborative skills. However, they had difficulty in using their critical thinking skills and creativity. Findings have implications for the design of an educational technology course that pre-service teachers comprehend and practice computational thinking concepts.

ARTICLE INFO

***Article History*:**
Received:23.11.2021
Received in revised form:29.12.2021
Accepted:29.12.2021
Available online:30.12.2021
Article Type: Standard Paper

Keywords: Computational thinking, computational thinking skills, pre-service teachers, course design, educational technology course design

1. Introduction

Today, societies are heavily influenced by computing at every point of their daily lives from shopping online to analyzing data. Students need to acquire some higher-order skills to understand computing principles which help solve the problems they encountered to succeed in 21st century (Angeli, & Giannakos, 2020; Hodgson, & Riley, 2001; Gretter, & Yadav, 2016). Recent efforts around the world are focused on helping students understand the concepts and skills in computer science at the K-12 level (Author, 2017). These concepts and skills have been introduced under the term computational thinking.

Even though the computational thinking term is new, it has a long history in computer science dating back to the 1950s as algorithmic thinking (Denning, 2009; Denning, & Tedre, 2019). Knuth (1985) defined algorithmic thinking as a method of producing solutions to problems, creation, sequencing, and control of problem solving processes. Also supporting tools that are used for the problem solving process can be planned and designed with algorithmic thinking skills (Barr, Harrison, & Conery, 2011). Computational Thinking concept was first proposed by Wing (2006) and expressed as a fundamental skill as reading, writing, and arithmetic that every child should acquire. Wing described computational thinking as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (Wing 2006, p. 33). Since

[1] Corresponding author's address: Computer Education and Instructional Technologies Department, Sakarya University, Sakarya, Turkey.
e-mail: ealbayrak@sakarya.edu.tr
DOI: 10.47156/jide.1027431

then, several divergent descriptions of computational thinking have been proposed, created confusion between faculty in teacher education programs, K-12 teachers and administrators about what computational thinking implies (Yadav, Stephenson & Hong, 2017). Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) (2011) developed an operational definition of computational thinking for K-12 and described computational thinking as a problem solving process involves nine computational thinking concepts: problem decomposition, abstraction, data collection, data analysis, data representation, algorithms and procedures, automation (using the technology to support abstraction), parallelization (organizing resources to perform tasks at the same time), and simulation. Further, Wing (2014) expressed computational thinking as a thinking process for formulating problems and expressing their solutions in a way that can be solved by a computer or human.

Among the many definitions of computational thinking in the literature (Barr & Stephenson, 2011; Brennan & Resnick, 2012; Grover & Pea, 2013), there is common computational thinking skills including "algorithmic thinking, abstraction, problem solving, decomposition, generalization, and debugging" (Saritepeci & Durak, 2017). Also, ISTE (2015) declared that computational thinking skills are an expression of creative thinking, algorithmic thinking, critical thinking, problem solving, cooperative learning and communication skills. Korkmaz, Cakir ve Ozden (2017) argued that communication skills were essential in the emergence of the above-mentioned skills, therefore they did not separately address the communication skills. In support of this, we redesigned our course and activities based on the five computational thinking skills (Doleck, Bazelais, Lemay, Saxena, & Basnet, 2017; Korkmaz et al. 2017) which are considered as 21st century skills that students should gain (Binkley et al., 2012): problem solving, algorithmic thinking, creativity, critical thinking and cooperativity. These dimensions are complementary to each other. The student defines a problem and determines its boundaries in problem-solving, produces the solution stages of the problem, puts them in the right order and controls them in algorithmic thinking (Knuth, 1985), creates new expressions, different thinking ways in creativity, criticize what he/she and his/her friends produced in critical thinking (Saban, & Saban, 2017) and shares ideas and produces by working together by using communication skills in cooperativity (Doleck et al., 2017).

1.1 Teaching Computational Thinking in Different Disciplines

Computational thinking is not necessarily involve working with the computer technologies but also involve developing skills to solve complex problems, analyzing data, working collaboratively and automating solutions in different disciplines including social sciences, math, art and literacy (Bundy, 2007; Barr & Stephenson, 2011; Voogt et al., 2015). To draw attention to the importance of computational thinking in different disciplines, Wing (2011) stated that "Computational thinking is not just or all about computer science. The educational benefits of being able to think computationally--starting with the use of abstractions--enhance and reinforce intellectual skills, and thus can be transferred to any domain" (p.4). In other words, computational thinking is not only required for learners who majored in computer science related departments, but also essential for learners who majored in other subjects (Wing, 2008; Kim, Kim, & Kim, 2013). Bundy also argued that computational thinking skills can be used in problem-solving processes in different disciplines, and these skills are crucial for every discipline (2007) since computational thinking has the potential to allow students to think in different ways and to develop different solutions for problems in any discipline (Yadav et al., 2014). For example, computational thinking can be applied in social sciences by deducing conclusions from facts; in language arts by identifying patterns of different sentence types; in math by doing long division (algorithms); and in science by applying order of operations in an expression (problem decomposition) (Barr & Stephenson, 2011).

To promote the implementation of computational thinking concepts in different subject areas in K-12, CSTA and ISTE have provided sources for teachers includes examples of using computational thinking concepts in teaching at K-12 level. In addition, Google, in 2010, has developed an online computational thinking course including computational thinking applications and lesson plans in different subjects, to highlight how computational thinking is applied in different disciplines. Another online resource is Barefoot supported by Department for Education in UK, provides variety of resource for teachers created by teachers to integrate computational thinking in classroom (Barefoot, 2019). On the other hand, some countries made an effort to integrate computational thinking into their K-12 curriculum (Balanskat & Engelhardt, 2014), while most of others do not use computational thinking in all disciplines (Hsu et. al., 2018).

Actually, introducing computational thinking in different subject areas is not easy since it is challenging for teachers to change the curriculum they are familiar with and to adapt new content and teaching strategies. However, it is crucial to adopt new teaching methods to engage learners. Also, applying programming concept in different subject areas could attract the attention of low achievement students and improve their achievement level (Hsu et. al., 2018).

1.2 Computational Thinking for Pre-Service Teachers in Different Disciplines

An important step that can be taken to embed computational thinking into the K-12 curriculum is to prepare future teachers to integrate computational thinking into their teaching (Yadav, Zhou, Mayfield, Hambrusch, & Korb, 2011). Teacher education programs should provide pre-service teachers opportunities to incorporate computational thinking into their practice to ensure their students use computational thinking skills. In the same way, teacher education programs should prepare pre-service teachers to understand computational thinking concepts and to ensure reflect these concepts on their teaching in different subject areas (Yadav et al., 2017).

In order to improve the computational thinking skills of pre-service teachers, computational thinking could be integrated into the courses that employ effective learning strategies. Several studies have suggested different learning strategies to help students improve their computational thinking skills including project-based learning, problem-based learning, collaborative learning, and game-based learning (Hsu et.al., 2018). Project-based learning is one of the most used learning strategies in the studies about promoting computational thinking in educational settings (Hsu et.al., 2018). This learning strategy includes complex tasks to solve challenging problems and give opportunities for students to engage in higher ordered activities such as designing, problem-solving, analyzing, and decision making (Jones, Rasmussen, & Moffitt, 1997; Marquez Lepe, Jimenez-Rodrigo, 2014). In support of this, recent research has also suggested that project-based learning approach is beneficial for the development of computational thinking (Gross, et. al, 2014; Missiroli, Russo, & Ciancarini, 2017; Hsu, Chang, & Hung, 2018) which improves students' problem solving skills in real world settings by requiring them to attend hands on activities (Jumaat, Tasir, Halim & Ashari, 2017). This approach provides student centered and active learning experiences that could be useful for acquisition of such multi-layered skills: defining discipline-specific issues, producing solutions to problems encountered, benefiting from the ideas of others, being able to criticize solutions (Angeli et al., 2016; ISTE 2015), designing solution steps, performing multi-layered tasks (Thomas, 2000; Doppelt, 2003). Therefore the project-based learning approaches can be very useful for pre-service teachers to develop their computational thinking skills as well (Hsu, Chang, & Hung, 2018): planning a course in a systematic way, specifying the boundaries of the course subject, detecting the specific needs of learners, determining the most appropriate teaching method, selecting or developing instructional materials that helps to improve student learning and being able to effectively integrate these materials into the planned course, thinking critically through planning the course and working in collaboratively to complete these tasks. This research also employed the project-based learning strategy to improve pre-service teachers' computational thinking skills and knowledge.

Research on pre-service teachers' computational thinking skills demonstrated that many of the pre-service teachers have difficulty in conceptualizing the computational thinking concepts (Bower & Falkner, 2015; Mouza et al., 2017). For example, Yadav et al. (2014) designed computational thinking modules and incorporated these modules into the required educational psychology course. In this study, the effects of the developed modules were assessed by measuring the computational thinking skill levels of the pre-service teachers. It was found that these modules increased pre-service teachers' awareness of computational thinking and influenced their attitude positively. Also, they have begun to think more about integrating computational thinking concepts into their future teaching. In another study, Mouza et al. (2017) redesigned an educational technology course that pre-service teachers make connections between content, pedagogy, and technology within the context of computational thinking concepts. They found that pre-service teachers' knowledge related to computational thinking concepts improved and pre-service teachers valued implementing these concepts in their future teaching. However, pre-service teachers reported that they do not feel comfortable with integrating computational thinking concepts and practices in classroom teaching. In another study, Pala & Mihci Turker (2019) examined the effects of Arduino IDE and C++ programming languages on the computational thinking skills of pre-service teachers studying on computer education and instructional programming department. Pre-service teachers exposed to basic programming training and then they were asked to design group projects on a voluntary basis. They found that pre-service teachers' algorithmic thinking, creativity and critical thinking skills has improved while problem solving and cooperativity skills remained the same after the course. The participants' background on technology and the course subject may have impact on the results. For example, Bower et al. (2017) observed the development of teachers' conceptual understanding of computational thinking structures in their study which had a project-based approach. According to the data obtained, teachers from different disciplines could only explain the detailed structures of problem solving and algorithmic thinking. Zha, Jin, Moore, & Gaston, (2020) structured a learning environment for preservice teachers with flipped learning approach. At the end of the study they developed participants' computational thinking attitudes.

Several studies in literature have largely focused on the development of the computational thinking skills of students at K-12 level (Sahiner & Kert, 2016). In addition, there is a need for further studies with project-based learning strategy focusing on developing pre-service teachers' computational thinking skills with a comprehensive and organized training process in different subject areas and examining pre-service teachers' experiences in their own field to improve their computational thinking skills. In this aspect, we examined a redesigned educational technology course's effect on computational thinking skills of pre-service teachers who majored in a specific discipline.

1.3 Research Questions

This study addresses the research questions below:

1. How does the educational technology course influence the computational thinking skill levels of pre-service teachers in literacy education?
2. To what extent did the pre-service teachers reflect their computational thinking skills in their course projects?

2. Methods

2.1 Research Context

This research was conducted within the educational technology course offered by an undergraduate teacher education program in Turkey. This course were redesigned to support the development of pre-service teachers' computational thinking skills and the ways in which computational thinking can be incorporated with specific subject area. A mixed method approach adopted in this study using both quantitative and qualitative data to find more detailed answers to research questions (Creswell, 2009). The experimental design was used that includes one group pretest-posttest (Fraenkel, Wallen, & Hyun, 2012). All data collection procedures were approved by the university's ethical committee and the study followed all the ethical standards.

2.2 Course Description

The educational technology course mandatory for all teacher candidates in the school of education named "Instructional Technologies and Material Development (ITMD)" (The course' name and content is revised as "Instructional Technologies" after this study) aims to make use of instructional technologies to develop correct, up-to-date and effective instructional materials for literacy education (see Table 1). "Using the project-based learning strategy to improve computational thinking skills variety of activities were developed: defining the problems, building knowledge through research, generating problem solving steps, thinking creatively, criticizing peers' work, and working collaboratively.

In the beginning of the course, pre-service teachers were informed about the goals of the course and were expected to be prepared for the course subjects in advance. Pre-service teachers were encouraged to conduct research through the university's online library system and an online search engine in order to improve their research and data collection skills. A quiz program was used to evaluate the pre-service teachers' knowledge about the course content in the beginning of each class and then the content was discussed to reinforce pre-service teachers' knowledge with using critical thinking skills.

Cooperation were emphasized throughout the course. Pre-service teachers were initially divided into groups of 3-4 people to improve their cooperative learning skills. Then, each group were asked to choose a topic in literacy education and to develop a lesson plan and instructional materials (digital and non-digital) for this topic. They were also asked to find subdomains of the topic they have chosen to ensure they could fully capture the framework of these topics and to design an effective lesson plan including the instructional materials. Finally, they were asked to present the plan. It is aimed to improve the pre-service teachers' problem solving and algorithmic thinking skills with these activities.

Pre-service teachers first designed unplugged (non-digital) activities for their lesson plan which enable them to practice real-world experiences (Nishida et al., 2009). The unplugged activities require less technical knowledge and cognitive skills compared to the complexity of the computing environment and also may lead pre-service teachers to look from different and productive ways of designing (Kotsopoulos et al., 2017). Thus, unplugged materials may be useful in taking the first important steps to develop computational thinking skills (Bell, Alexander, Freeman & Grimley, 2008) with less anxiety of a new situation (Meerbaum-Salant, Armoni & Ben-Ari, 2013). Second, they were asked to plan unplugged or plugged (with computer) activities for their group's subject. They were free to choose plugged or unplugged material for this assignment. Third, they were expected to plan two plugged activities. Last, each group presented their lesson plan and materials in detail to their classmates. Each group were expected to criticize the other groups' work in order to develop their critical thinking skills.

2.3 Participants

The participants of this study included 27 pre-service teachers from the Literacy Education Program enrolled in the ITMD course during Spring 2018 in a public university in Turkey. These participants were selected for purposeful sampling in order to obtain rich experiences since their major is different than computer education (Patton, 2015). Computational thinking is a way of thinking often used in the field of computer sciences. However, as its use and importance in other disciplines (Bundy, 2007), this study focus on the development of computational thinking skills of pre-service teachers who have not been trained as computer science educators. Pre-service teachers in a particular discipline were selected for this purpose to provide the transferability (external validity) of the research (Merriam & Grenier, 2019).

Table 1. Course activities and promoted computational thinking skills

Course Activities	Description of the activities	Computational Thinking Skills
Planning the Context	Participants choose a topic for final project, determine a problem related to the topic, decompose the problem in smaller tasks, draw the boundaries of the problem, then plan a lesson addressing this problem, specify appropriate teaching methods, and decide and design appropriate instructional materials for the lesson plan.	Problem solving Algorithmic thinking Cooperativity
CS Unplugged Activities	Participants plan and presents a non-digital instructional material by considering visual design principles and related theories and discuss the appropriateness of selected materials for their topic.	Algorithmic thinking Creativity Critical thinking Cooperativity
Concept Mapping	Participants uncover the major subdomains of the topic they selected for their project and revealed the relationships between these domains.	Algorithmic thinking Critical thinking Cooperativity
Scratch Activities	Participants experience basic coding concepts with Scratch programming tool.	Algorithmic thinking
Digital Storytelling & Sharing	Participants create a digital story with a program and discuss the usability of this digital material. They share the digital story on YouTube.	Algorithmic thinking Creativity Cooperativity
Final projects & Presentations	Participants prepare a 15-minute formal presentation of all the instructional materials they designed for their final project.	Algorithmic thinking Cooperativity

2.4 Data Collection

Both quantitative and qualitative data were collected. First the final projects developed by the pre-service teachers as qualitative data were assessed based on the rubric (see Table 2) to assess their computational thinking skills. The rubric was developed by the researchers with 4 points scale to examine pre-service teachers' computational thinking skills: problem solving, algorithmic thinking, creativity, critical thinking, and cooperativity. The final projects were evaluated by two researchers to provide the internal validity (Merriam, 2015).

Second, in order to assess the participants' computational thinking skills, a computational thinking survey that provides quantitative data was used at the beginning and at the end of the course. Because the course and activities in this study were designed based on the five computational thinking skills, we used this survey which is useful to measure the skills the study focused. The survey developed by Korkmaz et al., (2017) to collect data on how they feel about their computational thinking adequacy with a Cronbach's alpha of .82 (scale, 1: strongly disagree, 2: disagree, 3: no opinion, 4: agree, 5: strongly agree). It consists of 29 items with 5 dimensions; problem solving, algorithmic thinking, creativity, critical thinking, and cooperativity. Also, the scale can be used as one-dimensional scale to identify computational thinking skill levels. Students completed the survey online.

2.5 Data Analysis

The pre and post data collected through the computational thinking survey were analysed with SPSS. Paired samples t-test were conducted to determine significant differences between pre and post-tests since the assumption of normality is met and the sample size is greater than 25 (Springate, 2012; Johanson, & Brooks, 2010). With this analysis, the improvement of participants' computational thinking skills were measured. The final projects were evaluated via the rubric (Table 2) for a deeper understanding what extent the participants reflect their computational thinking skills after the course. In addition, in the context of the reliability of the study, two researchers in the field analysed the data obtained. %96 percent agreement was achieved by the researchers.

Table 2. Final Projects Assessment Rubric

Computational Thinking Concepts	1 point	2 points	3 points	4 points
Problem solving	-Failure to identify the problem statement. -Failure to draw the boundaries of the problem.	-Failure to identify the problem statement. -Draw the boundaries of the problem. -Having difficulties in breaking down tasks into smaller parts to produce solutions.	-Identify the problem statement -Draw the boundaries of the problem. -Having difficulties in breaking down tasks into smaller parts to produce solutions. -Having difficulties to select instructional materials.	-Identify the problem statement -Draw the boundaries of the problem. -Break down tasks into smaller parts to produce solutions. -Select appropriate instructional materials
Algorithmic thinking	-Failure to generate lesson plan steps in the logical order. -Failure to use the instructional material at the right place in the lesson plan. -Unable to use effective (pedagogical) methods and techniques in the lesson plan	-Failure to generate lesson plan steps in the logical order -Use the instructional material in the lesson plan but it is not effective. -Unable to use effective (pedagogical) methods and techniques in the lesson plan	-Generated the lesson plan steps in the logical order. -Planned three different (1.CS unplugged, 2. CS unplugged or plugged, 3. plugged) materials to be used in the lesson plan -Unable to use (pedagogical) methods and techniques used which provide effective flow in the course schedule	-Generated the lesson plan steps in the logical order. -Planned three different(1.CS unplugged, 2. CS unplugged or plugged, 3. plugged) materials to be used in the lesson plan -Able to use (pedagogical) method and techniques usage which provide effective flow in the course schedule
Creativity	-Failure to generate CS unplugged material	-Generate CS unplugged material but it doesn't contain an original theme.	-Generated CS unplugged material but it doesn't contain exactly original theme	-Generated CS unplugged material and it contains an original theme
	-To tell the digital story, performed only one of the elements;emotional connection, the power of music, dramatic question, the use of sound	-To tell the digital story, performed only two of the elements; emotional connection, the power of music, dramatic question, the use of sound	-To tell the digital story, performed only three of the elements; emotional connection, the power of music, dramatic question, the use of sound	-To tell the digital story, performed the elements; emotional connection, the power of music, dramatic question, the use of sound
Critical Thinking	-Course flow is not prepared considering the target group, cognitive load theory and visual design principles(Font, color...).	-Course flow is prepared considering the target group, but not the cognitive load theory and visual design principles.	-Course flow is prepared considering the target group and visual design principles, but not the cognitive load theory.	-Course flow is prepared considering the target group, visual design principles and the cognitive load theory.
	-To tell the digital story; none of the elements has been provided including perspective, economic use of sound, image and words and the rhythm of the story (speed)	-To tell the digital story; only one of the elements has been provided including perspective, economic use of sound, image and words and the rhythm of the story (speed)	-To tell the digital story; only two of the elements has been provided including perspective, economic use of sound, image and words and the rhythm of the story (speed)	-To tell the digital story; the elements provided including perspective, economic use of sound, image and words and the rhythm of the story (speed)
Cooperativity	-Failure to establish collaboration in working with the group	-Collaboration was ensured but original and specific ideas weren't presented by different people.	-Collaboration was ensured but original ideas weren't presented by different people.	-Collaboration was ensured and original ideas were presented by different people in group.

3. Findings

This section addresses the research questions in relation to examine pre-service teachers' computational thinking skills and what extent they use their computational thinking skills in a redesigned educational technology course (ITMD) developed within the computational thinking context.

3.1 Research Question 1: How Does the Educational Technology Course Influence the Computational Thinking Skill Levels of Pre-Service Teachers in Different Subject Area?

To test for the significance of the gain score (post survey-pre survey), paired samples t-test was conducted on each of the scales. The results are tabulated in Table 3. There is a significant gain in algorithmic thinking skill and the instrument as a whole. Contrary, there is no significant gain on pre-service teachers' problem solving, cooperativity, critical thinking and creativity skills. The last column of Table 3 reports the effect size for all scales. The effect size denotes the increase in the mean score in standard deviation units. Medium effect sizes were documented for algorithmic thinking and the instrument as a whole. Overall, participants showed significantly higher achievement levels on the post-test than on the pre-test (t = 2,173, df = 19, p = .043). The effect size of the analyses (M2-M1/SD) was found d = -0.34 indicating a medium effect size (Cohen's, 1988).

Table 3. Statistics for Computational Thinking and Sub-dimensions

Statistic	Pre survey mean	D	Post Survey mean	SD	t	df	P value	d
Problem solving	3.38	.74	3.42	.82	.270	19	.790	.04
Algorithmic thinking	2.36	.76	2.67	.87	2.510	19	.021*	.38
Cooperativity	3.72	.87	3.87	.75	1.007	19	.326	.18
Critical Thinking	3.55	.77	3.67	.47	.870	19	.395	.17
Creativity	4.04	.44	4.08	.41	.451	19	.657	.08
Total	3.43	.43	3.55	.35	2.173	19	.043*	.31

Note. df = degrees of freedom, *d* refers to Cohen's d (effect size), *p< .05

3.2 Research Question 2: To What Extent Did the Pre-Service Teachers Reflect Their Computational Thinking Skills in Their Course Projects?

To answer the second question of the research, pre-service teachers' final projects were evaluated by using the rubric presented in Table 2. Study groups' computational thinking scores from final projects are represented in Table 4.

Table 4. Computational Thinking Scores of Study Groups for Final Projects

Groups	Problem Solving	Algorithmic Thinking	Creativity Unplugged	Creativity Plugged	Critical Thinking Unplugged	Critical Thinking Plugged	Cooperativity
Group 1	4	4	3	2	2	3	4
Group 2	4	4	4	2	2	3	4
Group 3	4	3	3	2	2	3	4
Group 4	3	3	3	2	2	3	4
Group 5	4	4	2	1	2	2	3
Group 6	4	3	4	3	3	3	4
Group 7	4	4	4	3	3	2	4
Group 8	3	3	4	3	4	3	4
Mean	3.75	3.5	2.81		2.63		3.87

Analysis of the projects showed that, participants had high score on problem solving (3.75 out of 4), algorithmic thinking (3.5 out of 4) and cooperativity skills (3.87 out of 4), and low score on creativity (2.81 out of 4) and critical thinking skills (Table 4). Most participants were able to use problem solving skills to overcome the issues they faced when designing the lesson plan. Nearly all of the study groups had competence on defining the problem situation, determining the limits of the problems, generating solutions by breaking down the complex problems into smaller parts that are more manageable, deciding instructional materials to integrate in the lesson plan the participants designed. Only two groups initially experienced difficulties in using problem-solving skills through working on the final project. They had a very difficult time to reach a consensus on what kind of material to design for their lesson plan. Participants' were also had high score on using their algorithmic thinking skills through designing their final project. They had successfully implemented step by step procedures to complete their lesson plan they prepared for their final project including plugged and unplugged instructional materials. Participants were also received high score on cooperativity skills. They were able to work together for their group success especially for the problems they encounter while working on their final project.

Even though pre-service teachers were able to use problem solving, algorithmic thinking, and cooperative working skills, they had some difficulty in using creativity and critical thinking skills. While some groups have developed instructional materials that were relatively unique, some other groups designed materials not creative. For example, study group 3 designed three different cardboards to describe all of the noun types (Figure 1). The purpose of these materials was to teach noun types by playing a game. The rule of the game was to match the pictures with correct labels on the board. In this context, each carton were divided into two labels (e.g. singular and plural names). The pictures were placed in a cardboard box. Singular object in the picture that was to be drawn from the box could be placed under the label called singular noun, the plural noun could be placed under the label called plural noun. The material overall was not creative and did not reflect any specific theme. The material was limited to measure learning outcomes other than to promote learning. This group also created a digital story with an online program has also been examined. Even though the digital story used strength of the music and dramatic question items, it did not compose an emotional connection with the audience (see Fig. 2).

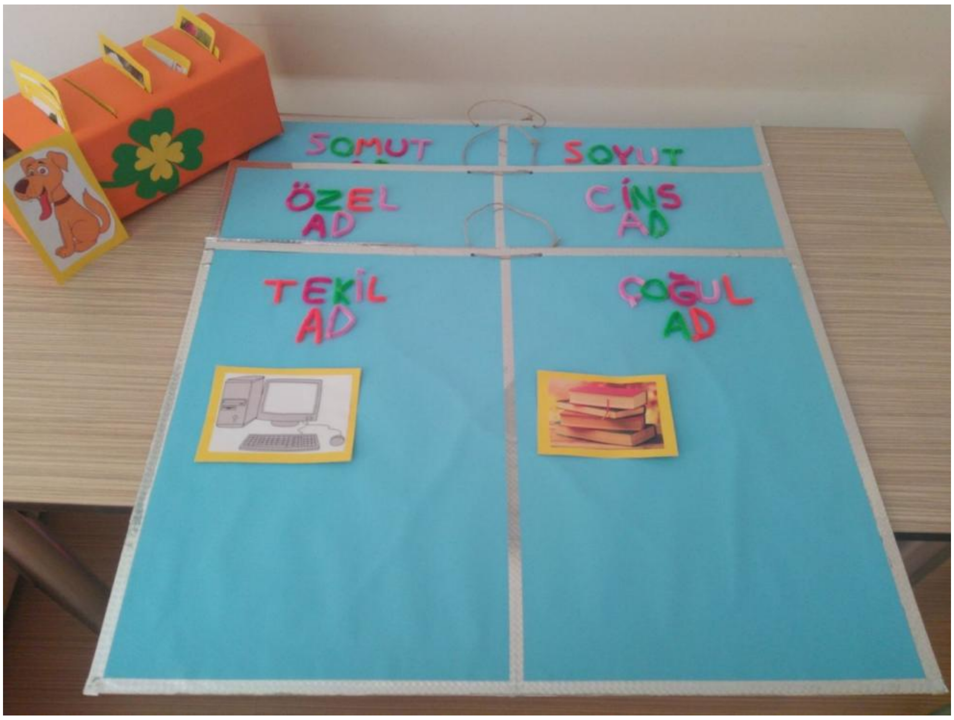

Figure 1. Sample of an unplugged material, designed by group 3

Note: Translation of words has been inserted into the image.

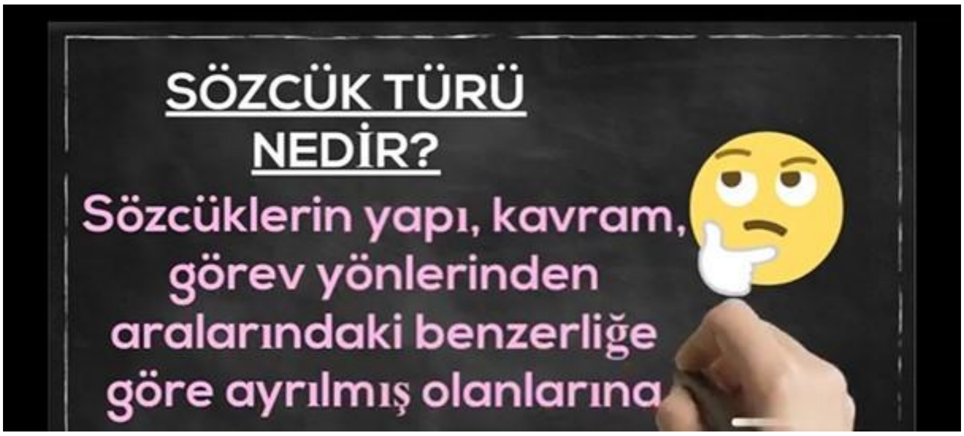

Figure 2. Sample of a plugged material (Digital story), designed by group 3

Note: Translation of the text; What is word form? The words that are separated according to their similarities in terms of structure, concept, and task…

Another group (Group 6) project with the highest score on creativity. The group designed the instructional material with a unique theme. The ladder-shaped material reflects a tale. The characters and objects in the tale are placed on each step and the story text was written in pieces on small papers placed on the material. In this way, different elements were brought together in a creative way by making use of both text and pictures.

Also, some groups had some difficulties in using critical thinking skills. While planning the materials, they did not sufficiently implement the visual design principles and take precautions to reduce the cognitive load. For example, the unplugged material of group 3 (Figure 1) had beautiful colors, but the color of the background and the text were not sufficiently contrasting, so it could be difficult to read. In addition, there were long sentences in many scenes in the digital story created by group 3 (Figure 2). This problem would cause excessive cognitive load especially considering that their target audience is 8th grade students. Besides, they developed their digital story like a simple presentation rather than storytelling and did not use an original theme.

Unlike, group 6 who scored high in critical thinking was able to prepare the material by considering the principles of visual design, cognitive load theory and the characteristics of the target audience. Their digital story was creative. The audio, view, and pictures that included in the story were sufficient in terms of content with a specific viewpoint and rhythm (Figure 3). They provided a digital story that reflects a clear main idea and flow reflects their creativity skills.

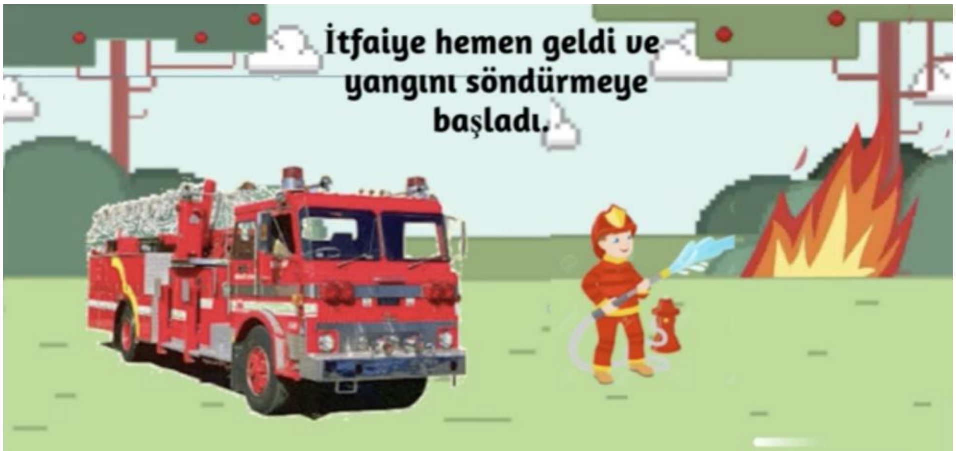

Figure 3. Sample of a digital story material, designed by group 6

4. Discussion

In this study, an educational technology course (ITMD) was redesigned in the context of computational thinking to help pre-service teachers understand and implement computational thinking concepts within the context of their own discipline area. This course adopted a project-based approach and included structured activities throughout the course, aiming to provide a deeper understanding of computational thinking and to ensure using computational thinking related skills: problem solving, algorithmic thinking, cooperativity, critical thinking and creativity.

Data analysis from the computational thinking survey revealed that pre-service teachers' algorithmic thinking skills and computational thinking skills in general were improved throughout the course. According to Brown (2015), algorithmic thinking includes understanding a problem, explaining it clearly in sentences, making a plan for solving it, implementing the problem solving steps, accessing the useful tools and using them appropriately and evaluating the effectiveness of the problem solving process. In this respect, the development of the algorithmic thinking skill could have a positive effect on the problem-solving process. Similarly, Bower et. al. (2017) found that in-service teachers improved only their algorithmic thinking skills after a training program they designed which might suggests that developing each computational thinking skills may be difficult for pre-service teachers in specific disciplines that not related to the technology education. Yadav et al. (2014) and Mouza et al. (2017) observed the improvement of pre-service teachers' computational thinking skills in general, however, they did not focus on the improvement of any specific computational thinking skill.

Findings from the analysis of final projects indicated that pre-service teachers were competent to use their problem solving, algorithmic thinking and cooperativity skills through working on their projects. On the other hand, some computational thinking skills were not well represented in pre-service teachers' final projects. They had difficulty in using their critical thinking skills and creativity while developing the instructional materials. For instance, they did not efficiently follow some considerations: characteristics of the target audience, visual design principles, and related theories for designing instructional materials. Since creativity and critical thinking is a high-level skill that one can put forward, there might be need for different settings and structured activities to ensure pre-service teachers reveal their skills.

4.1 Implications

The findings of current study have important implications for teacher educators and future research. This study has moved beyond the studies in the literature by engaging pre-service teachers in a course that they practiced their computational thinking skills in a specific subject area. The participants developed a lesson plan in literacy education. Through the planning process they were asked to complete some structured activities to ensure that they use their problem solving, algorithmic thinking, cooperativity, critical thinking and creativity skills. After attending the course redesigned in the context of computational thinking, the participants improved their problem solving, algorithmic thinking and cooperativity skills. The course activities and projects helped them build a realization of the computational thinking skills and concepts. Considering that pre-service teachers should use their computational thinking skills in their own discipline, this study would be useful to examine their skills in such a practical environment.

As one of the computational thinking skills, algorithmic thinking that is essential for every stage of the problem solving process (Futschek, 2006) has been clearly observed with two different measurements, survey and rubric in this study. Previous studies on computational thinking have little contribution on development of the algorithmic thinking (Mumcu and Yildiz, 2018). With the development of this skill, pre-service teachers have taken an important step in comprehend and utilize computational thinking skills (Knuth, 1985). In addition, it will be beneficial for pre-service teachers to teach students computational thinking skills in their future professional life (Burton, 2010).

4.2 Limitations and Future Research

There are some inherent limitations in the study. The sample of this research is limited to the pre-service teachers studying in the literacy education programs. In addition, the purposeful sampling method adopted in this study is limited in terms of providing generalisability. To strengthen the empirical base of this study, further studies could be conducted with larger size of samples and pre-service teachers in different disciplines including math, science, social sciences, language arts. Also, the current study used a survey to analyze pre-service teachers' computational thinking skill levels and used a rubric to examine what extent pre-service teachers used their computational thinking skills in final projects. In future research, different measurement tools can be developed to analyze their computational thinking skills in more detail. For instance, in-process evaluation and observation method could be used, therefore it would be possible to observe pre-service teachers' efforts, attitudes and challenges at each problem solving step (Tang, Yin, Lin, Hadad, & Zhai, 2020).

In this study pre-service teachers were not able use their critical thinking and creativity as expected. Pre-service teachers may need to practice higher level activities in a long time period to improve these skills. According to Aksoy (2004), creativity is closely related to bringing together different combinations of ideas, colors and words. To develop creativity skills of pre-service teachers, encouraging and inspiring environments, tools and activities can be provided for creating original, effective, and enjoyable products (McKenney, Kali, Markauskaite, & Voogt, 2015) with unusual visual representations, instead of repeating existing samples. In addition, critical thinking skills can be developed via encouraging pre-service teachers to evaluate the competence of the products they designed for a project or an activity. They might be encouraged to criticize their work on specific issues such as the usefulness of the product, its appropriateness to the target group and redesign the product by addressing the issues identified.

In summary, scholars and teacher education programs should focus on development of pre-service teachers' comprehension of computational thinking for successful integration of computational thinking in K-12 settings. Beyond this, teacher educators may integrate computational thinking into different courses including content-specific method courses. Finally, teacher education programs should require preservice teachers to design and implement lesson plans in which students develop their CT skills.

References

Aksoy, B. (2004). *Problem based learning approach in geography teaching.* (Doctoral Thesis). Ankara: Institute of Educational Sciences Gazi University in Turkey https://tez.yok.gov.tr/UlusalTezMerkezi/

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior, 105*, 106185.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society, 19*(3), 47-58.

Balanskat, A., & Engelhardt, K. (2014). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. Brussels, Belgium: European Schoolnet. Retrieved from https://goo.gl/i5aQiv

Barefoot, (2019). *Classroom Resources*. Retrieved from https://www.barefootcomputing.org/about-barefoot

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

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?, *ACM Inroads, 2*(1), 48-54. doi: 10.1145/1929887.1929905

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2008, July). *Computer science without computers: new outreach methods from old tricks.* Paper presented at the 21st Annual Conference of the National Advisory Committee on Computing Qualifications, Auckland, New Zealand. Retrieved from

https://www.citrenz.ac.nz/conferences/2008/127.pdf

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., & Rumble, M. (2012). Defining twenty-first century skills. In P. Griffin, B. McGaw, & E. Care (Eds.) *Assessment and Teaching of 21st Century Skills* (pp. 17–66). New York, NY: Springer.

Bower, M., & Falkner, K. (2015, January). *Computational thinking, the notional machine, pre-service teachers, and research opportunities*, Paper presented at the 17th Australasian Computer Education Conference, Sydney. Retrieved from

https://pdfs.semanticscholar.org/c2df/f4fdd833c44015fedff1e9ae480740894a7b.pdf

Bower, M., Wood, L. N., Lai, J. W., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the Computational Thinking Pedagogical Capabilities of School Teachers. *Australian Journal of Teacher Education, 42*(3). doi: 10.14221/ajte.2017v42n3.4

Brennan, K., & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada. Retrieved from https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Brown, W. (2015). Introduction to algorithmic thinking. Retrieved from https://raptor.martincarlisle.com/Introduction%20to%20Algorithmic%20Thinking.doc

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing, 1*(2), 67-69. Retrieved from https://core.ac.uk/download/pdf/28961399.pdf

Burton, B. A. (2010). Encouraging algorithmic thinking without a computer. *Olympiads in Informatics, 4*, 3-14.

Cook, T. D. (2003). Why have educational evaluators chosen not to do randomized experiments?. *The Annals of the American Academy of Political and Social Science*, *589*(1), 114-149.

Creswell, J. W. (2009). Mapping the field of mixed methods research. *Journal of Mixed Methods Research, 3*(2), 95-108. doi: 10.1177/1558689808330883

CSTA & ISTE (2011). *Operational definition of computational thinking for K-12 education*. Retrieved from https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

Denning, P. J. (June 2009). The profession of IT. Beyond computational thinking. *Communications of the ACM, (52)*6, 28–30. Retrieved from https://sgd.cs.colorado.edu/wiki/images/7/71/Denning.pdf

Denning, P. J., & Tedre, M. (2019). *Computational thinking*. Cambridge, MA: MIT Press.

Doleck, T., Bazelais, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: Exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education, 4*(4), 355-369. doi: 10.1007/s40692-017-0090-9

Doppelt, Y. (2003). Implementation and assessment of project-based learning in a flexible environment. *International journal of technology and design education, 13*(3), 255-272.

Ehrenberg, R. G., Brewer, D. J., Gamoran, A., & Willms, J. D. (2001). Does class size matter?. *Scientific American*, *285*(5), 78-85.

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2012). *How to design and evaluate research in education*. (8th ed.). New York: McGraw-Hill Education.

Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. Algorithmic thinking: the key for understanding computer science. *Lecture Notes in Computer Science* (pp. 159-168). doi: 10.1007/11915355_15

Gretter, S., & Yadav, A. (2016). Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills. *TechTrends, 60*(5), 510-516. doi: 10.1007/s11528-016-0098-4

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38-43. doi: 10.3102/0013189X12463051

Gross, S., Kim, M., Schlosser, J., Mohtadi, C., Lluch, D., & Schneider, D. (2014, April). Fostering computational thinking in engineering education: Challenges, examples, and best practices. In IEEE *Global Engineering Education Conference (EDUCON)* (pp. 450-459). Istanbul, Turkey.

Hodgson, T., & Riley, K. J. (2001). Real-World Problems as Contexts for Proof. *Mathematics Teacher, 94*(9), 724-728. Retrieved from

https://search.proquest.com/openview/5df029d3c44a057002183041da9ba239/1?pqorigsite=gscholar&cbl=41299

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296-310. doi: 10.1016/j.compedu.2018.07.004

ISTE (2015). *CT Leadership Toolkit*. Retrieved from https://id.iste.org/docs/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4

Johanson, G. A., & Brooks, G. P. (2010). Initial scale development: sample size for pilot studies. *Educational and psychological measurement*, *70*(3), 394-400

Jones, B. F., Rasmussen, C. M., & Moffitt, M. C. (1997). *Real-life problem solving: A collaborative approach to interdisciplinary learning*. Washington DC: American Psychological Association. doi:10.1037/10266-000

Jumaat, N.F., Tasir, Z., Halim, N.D.A. & Ashari, Z.M. (2017). Project-based learning from constructivism point of view. *Advanced Science Letters*, 23 (8), 7904-7906.

Kim, B., Kim, T., & Kim, J. (2013). Paper and pencil programming strategy toward computational thinking for non-majors: Design your solution. *Journal of Educational Computing Research*, *49*(4), 437-459.

Knuth, D.E. (1985). Algorithmic thinking and mathematical thinking, *The American Mathematical Monthly, 92*(3), 170-181, doi: 10.1080/00029890.1985.11971572

Korkmaz, O., Cakir, R., & Ozden, M.Y. (2017). A validity and reliability study of the Computational Thinking Scales (CTS). *Computers in Human Behavior*, *72*, 558-569. doi: 10.1016/j.chb.2017.01.005

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education*, *3*(2), 154-171. doi: 10.1007/s40751-017-0031-2

Lou, Y., Abrami, P. C., & d'Apollonia, S. (2001). Small group and individual learning with technology: A meta-analysis. *Review of educational research*, *71*(3), 449-521.

Marquez Lepe, E., Jimenez-Rodrigo, M.L. (2014). Project-based learning in virtual environments: a case study of a university teaching experience. *International Journal of Educational Technology in Higher Education* 11(1), 76–90. doi: 10.7238/rusc.v11i1.1762

McKenney, S., Kali, Y., Markauskaite, L., & Voogt, J. (2015). Teacher design knowledge for technology enhanced learning: an ecological framework for investigating assets and needs. *Instructional Science*, *43*(2), 181-202. doi: 10.1007/s11251-014-9337-2

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, *23*(3), 239-264. doi: 10.1007/s11251-014-9337-2

Merriam, S. B. (2015). Qualitative research: Designing, implementing, and publishing a study. In Victor X. Wang (Ed.), *Handbook of Research on Scholarly Publishing and Research Methods* (pp. 125-140). Hershey, PA: IGI Global.

Merriam, S. B., & Grenier, R. S. (Eds.). (2019). Assessing and evaluating qualitative research. *Qualitative Research in Practice: Examples for Discussion and Analysis* (pp. 19-32). USA: Jossey-Bass.

Missiroli, M., Russo, D., & Ciancarini, P. (2017, November). Cooperative Thinking, or: Computational Thinking meets Agile. In *Proceedings of the 30th Conference on Software Engineering Education and Training (CSEE&T)* (pp. 187-191). Savannah, USA.

Mumcu, H. Y., & Yildiz, S. (2018). The Investigation of Algorithmic Thinking Skills of 5th and 6th Graders at a Theoretical Dimension. *MATDER Journal of Mathematics Education,* (1), 41-48.

Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. (2009). A CS unplugged design pattern. *ACM SIGCSE Bulletin*, *41*(1), 231-235. doi: 10.1145/1539024.1508951

Pala, F. K., & Mihci Turker, P. (2019). The effects of different programming trainings on the computational thinking skills. *Interactive Learning Environments,* 1-11. doi: 10.1080/10494820.2019.1635495

Patton, M. Q. (2015). *Qualitative research & evaluation methods: Integrating theory and practice* (4th ed.). Thousand Oaks, CA: Sage.

Saban, A., & Saban, A. I. (2017). Examination of Science Perceptions of Teacher Candidates. In I. Koleva, & G. Duman (Eds.), *Educational Research and Practice*, (pp. 214-224). Sofia: St. Kliment Ohridski University Press.

Sahiner, A., & Kert, S. B. (2016). Examining studies related with the concept of computational thinking between the years of 2006-2015. *EJOSAT: European Journal of Science and Technology, 5*(9). Retrieved from http://dergipark.gov.tr/download/issue-full-file/30420

Sarıtepeci, M., & Durak, H. (2017). Analyzing the effect of block and robotic coding activities on computational thinking in programming education. In I. Koleva, & G. Duman (Eds.), *Educational Research and Practice*, (pp. 490-501). Sofia: St. Kliment Ohridski University Press.

Springate, S. D. (2012). The effect of sample size and bias on the reliability of estimates of error: a comparative study of Dahlberg's formula. *The European Journal of Orthodontics*, *34*(2), 158-163.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education, 148*, 103798.

Thomas, J. W. (2000). *A review of research on project-based learning*, San Rafael, CA: Autodesk Foundation.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715-728. doi: 10.1007/s10639-015-9412-6

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. doi: 10.1145/1118178.1118215

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. (2011). Research notebook: Computational thinking—What and why?. *The Link Magazine*, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from http://link.cs.cmu.edu/article.php?a=600

Wing, J. M. (2014). Computational thinking benefits society. *40th Anniversary Blog of Social Issues in Computing*, *2014*. Retrieved from http://socialissues.cs.toronto.edu/index.html%3Fp=279.html

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education, 14*(1). doi: 10.1145/2576872

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

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011, March). Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 465-470). ACM. doi: 10.1145/1953163.1953297