A STUDY ON STUDENTS' COMPUTATIONAL THINKING SKILLS AND SELF-EFFICACY OF BLOCK-BASED PROGRAMMING

By

PINAR MIHCI TÜRKER *

FERHAT KADIR PALA **

*-** Department of Computer Education and Instructional Technologies,
Aksaray University, Aksaray, Ankara, Turkey.

Date Received: 15/10/2019 Date Revised: 29/10/2019 Date Accepted: 03/03/2020

ABSTRACT

The aim of this study is to investigate the computational thinking skills of secondary school students and their perceived self-efficacy related to block-based programming. The study group consists of 464 students attending 5 , 6 , 7 , and 8 grades in a province located in the Central Anatolia region, Turkey. Data were collected by Computational Thinking Skills Scale and Self-Efficacy Perception Scale related to Block-Based Programming. Independent sample t-test was used for comparisons of gender, programming education, Scratch and Arduino learning status apart from lessons. According to the findings, it was determined that the computational thinking skills of students were high. Significant differences were found in both computational thinking skills and self-efficacy perceptions of block-based programming according to gender, programming education, and Scratch and Arduino learning status.

Keywords: Computational Thinking, Block-based Programming, Self-efficacy, Secondary School Students.

INTRODUCTION

Nowadays, when technology surrounds us, it is difficult to determine a place where technology does not touch. In agriculture, industry, education, health, art, transportation, technology is used extensively in all areas of our lives. In this direction, changes have taken place in all areas where technology has touched, and took its share in this change in education. Today, it has become one of the primary objectives of education to educate productive individuals who can use the developing technology effectively, who can access information when they need it, by structuring the information, thinking critically and creatively. Recently, in addition to these skills, computational thinking skills have taken place in education as an important feature that individuals should possess.

Computational thinking is one of the new concepts that have been studied in the literature in recent years. Therefore, the definitions for this concept vary and their content is not yet clearly defined (Brennan & Resnick, 2012; García-Peñalvo & Mendes, 2018). However, according to Wing (2006), where the term is frequently mentioned together, computational thinking is a form of analytical thinking that includes elements, such as problem solving, system design and understanding human behavior based on the concepts in computer science. International Society for Technology in Education ISTE (2018) defines this skill as the development and use of strategies to understand and solve problems in order to benefit from the power of technological methods to develop and test solutions to problems.

Yıldız and Çiftçi (2017) think that one of the most effective ways to develop computational thinking skills is programming education. Therefore, students' self-efficacy perceptions are thought to be important in the development of these skills. According to Bandura (1977), individuals' self-efficacy perception towards a task is directly related to their performance and effort in successfully performing that task. So, high self-efficacy perception of individuals in a context is important as it will be related to its performance. As a matter of fact, it is seen in the literature that students' self-efficacy perception scores related to programming affect their programming performance (Aşkar & Davenport, 2009; Mazman & Altun 2013).

1. Related Literature

Brennan and Resnick (2012), on the other hand, made the definition of computational thinking based on Scratch experiences. The researchers identified three key dimensions of computational thinking. The first dimension is called computational concepts and covers the concepts that individuals use to develop programs (sequences, loops, parallelism, events, conditionals, operators, and data, etc.). The second dimension is computational practices and is based on the activities that individuals do while developing programs such as being incremental and iterative, testing and debugging, reusing and remixing, and abstracting and modularizing. The third dimension is computational perspectives. This dimension includes individuals' understanding of the code writing process. Kalelioğlu et al. (2016) identified the most common features used in explaining computational thinking in their literature review. They stated that these features consist of concepts, such as problem solving, abstraction, algorithmic thinking, pattern recognition, and system based thinking.

Many researchers mentioned uncertainties about how to develop computational thinking in individuals and how to transfer it to the classroom environment (Czerkawski & Lyman, 2015; Demir & Seferoğlu, 2017; Grover & Pea, 2013; Guzdial, 2008; Kalelioğlu et al., 2016; Kazimoglu et al., 2012; Lee et al., 2011; Wing, 2008; Yadav et al., 2011). They have emphasized that there are studies which are limited in the literature (Kalelioğlu et al., 2016; Korkmaz et al., 2015).

Yıldız and Çiftçi (2017) think that one of the most effective ways of developing these skills is programming education. Similarly, Sayın and Seferoğlu (2016) emphasized the importance of programming education and stated that programming skill was accepted as one of the 21st century skills. Lye and Koh (2014) also emphasize that programming education plays a key role in computational thinking. Because studies show that programming education develops many high-level thinking skills that are covered by computational thinking. Problem solving (Çatlak et al., 2016; Karabak & Güneş, 2013; Shin et al., 2013), creative thinking, critical analysis, continuous learning through systematic experiment and process (Çatlak et al., 2016; Monroy-Hernández & Resnick, 2008) and product creation (Çatlak et al., 2016) are among these skills. The studies show that students' perceptions of self-efficacy related to programming have changed after programing education (Mazman & Altun, 2013; Davidson et al., 2010), so the situation of taking programming education was included in the study.

In the studies, the importance of programming education is emphasized at an early age and it is stated that programming education can make positive contributions to the development of students (Özçınar et al., 2016; Demirer & Nurcan, 2016). However, reasons such as the need for expertise in text-based programming languages (Esteves & Mendes, 2004), abstract thinking skills, and complex language (Gomes & Mendes, 2007) revealed the idea that block-based programming languages would be more appropriate for young people. Because of the age of secondary school students, concrete programming languages are preferred over abstract programming languages in Turkey. That's why in Information Technology courses, Arduino and Scratch are lectured in secondary schools. Therefore, in line with the aims of the study, research related to block based programming such as Scratch are included.

Yükseltürk and Altıok (2018) state that there are few studies in the field of block-based programming and computational thinking skills, and these studies are generally aimed at facilitating beginners to learn programming and creating projects through problem-solving strategies. Researchers also point out that text-based programming leads to lack of motivation and negative prejudices in students, due to the complexity of its specific rules and syntax. Students create products by combining components, such as animation, sound, painting and music with block based programming. This provides an environment for them to develop their problem-solving and computational thinking skills. As a result of another study conducted by Yükseltürk and Altıok (2017), the researchers stated that block-based programming can be an important way to develop students' computational thinking and creative thinking skills.

García-Peñalvo and Mendes (2018) state that computational thinking is interpreted by researchers in various ways and that this concept is taken into the teaching process with different approaches in the classroom. Examples of these approaches include block-based programming, text-based programming, and physical kits created to control robotic systems or objects.

Howland and Good (2015) examined the effect of students on learning computational thinking using the Flip application, a visual / block based programming language. At the end of the study; it was determined that the students 'motivation increased as a result of producing their own games, the application was effective in the students' learning the concepts related to computational thinking and that the female students were more successful in the programming process. However, Kobsiripat (2015) found that learning activities with Scratch contributed to the development of students' creativity.

Kalelioğlu et al. (2014) examined the problem-solving skills of fifth grade students by using Scratch platform. In the light of the findings, it was observed that Scratch programming language did not affect students' problem-solving skills. However, all students developed a positive attitude towards programming with Scratch and stated that they found the environment easy. Kukul and Gökçearslan (2014) examined the problem-solving skills of the students who received programming training for the first time in their study conducted with 304 students in the fifth and sixth grades. Scratch was used in the programming education of the students. According to the findings, the problem-solving skills of the students were found to be high. However, it is seen that this skill is not meaningful in terms of variables, such as gender, grade level, computer status. The studies showed that the attitudes of female students towards programming could differ according to male students, so gender variable was also examined in this study.

As a result of applications with secondary school students using block based programming, Fadjo (2012), it was determined that the applications play an important role in the development of students' computational thinking skills and conceptual knowledge. Rizvi et al. (2011) state that students programming with Scratch have a high degree of self-efficacy. In the study conducted by Bishop-Clark et al. (2007) using the Alice application with students who do not have experience in programming language, the students were examined from a qualitative and quantitative perspective. The findings showed that students enjoyed programming, increased their self-confidence in programming, and showed significant improvements in understanding programming concepts.

According to Kasalak (2017), in the literature, block-based programming is widely used in the programming education of secondary school students, but there is no study with self-efficacy perception related to block-based programming. Based on these findings, this study was designed to eliminate the uncertainty existing in the literature and to determine the general situation for perceptions of computational thinking skills and self-efficacy regarding block-based programming.

2. Purpose

The aim of this study is to investigate the computational thinking skills and self-efficacy perceptions of secondary school students according to some variables. For this purpose, the following questions were examined.

- What is the level of computational thinking skills of secondary school students?
- Do secondary school students' levels of computational thinking skills differ according to following variables?
  - Gender
  - Programming education
  - Studying status of Scratch apart from lessons
  - Studying status of Arduino apart from lessons
- What are the levels of self-efficacy perceptions of secondary school students towards block-based programming?
- Do secondary school students' self-efficacy perceptions related to block-based programming differ according to following variables?
  - Gender
  - Programming education
  - Studying status of Scratch apart from lessons
  - Studying status of Arduino apart from lessons
- Is there a relationship between secondary school students' computational thinking skills and self-efficacy perceptions of block-based programming?

3. Method

3.1 Model of Research

This study examines the computational thinking skills of 5th, 6th, 7th and 8th grade students in a province in Central Anatolia and their perceptions of self-efficacy for block-based programming and the relationship between these two variables. This study, which aims to describe an existing situation, is an example of a cross-sectional survey model (Büyüköztürk et al., 2018). The dependent variables of the study are students' computational thinking skills and self-efficacy perceptions of block-based programming. On the other hand, gender, programming education, and Scratch and Arduino working conditions outside of the course are the independent variables examined.

3.2 Research Group

The study was carried out with the participation of 464 secondary school students. A stepwise path was followed in selecting the participants, the purposeful sampling method and then the random sampling method was used. Accordingly, the schools in the city centre that provide programming education and do not provide programming education are listed separately. Two schools form each list were randomly selected and the study was carried out in these four schools. In the schools, each class was chosen randomly and scales were applied. Information about the sample of the research is tabulated on the basis of participants. Demographic characteristics of the students are given in Table 1. When the demographic information about the students were examined, 49.6% were female (N = 230) and 50.4% were male (N = 234). It is seen that 73.5% of the students received programming education (N = 341). 56.5% of the students had Scratch (N = 262); 17.0% use Arduino (N = 79) outside the classroom and study on these platforms.

3.3 Data Collection Tools

In this study, Computational Thinking Levels Scale (CTLS) was used by Korkmaz et al. (2015) to determine the computational thinking skills scores of secondary school students. The scale consisted of four items of creativity, four items of algorithmic thinking, four items of cooperativity, four items of critical thinking and six items of problem solving. The scale was prepared as a five-point likert. The internal consistency coefficient Cronbach's alpha values were 0.64 for the creativity dimension; 0.76 for algorithmic thinking; 0.81 for the cooperativity dimension; 0.71 for critical thinking; 0.86 for problem solving and 0.80 for the overall scale. The following formula (Formula 1) is used to score

Variables	Categories	N	%
Gender	Female	230	49.6
	Male	234	50.4
Programming Education	Taken	341	73.5
	Not Taken	123	26.5
Studying status of Scratch a part from lessons	Yes	262	56.5
	No	202	43.5
Studying status of Arduino apart from lessons	Yes	79	17.0
	No	385	83.0

Table 1. Demographic Information of Students

the scale. According to the scores obtained, computational thinking skills of individuals are determined as weak, medium and high.

$$X_{standard\ score} = \frac{X_{raw\ score}}{Scale\ item\ numbers.} X20 \qquad (1)$$

Perceived self-efficacy scale related to block-based programming was developed by Kasalak (2017). In the scale, there are five items for simple block based programming tasks; seven items are complex block based programming tasks; The scale was prepared as a five-point likert. The internal consistency coefficient of the scale was found to be 0.82 for the dimension of simple block-based programming tasks; 0.86 for complex block-based programming tasks.

3.4 Implementation Process

At the beginning of the study, necessary permissions were obtained from the Provincial Directorate of National Education and scales were applied in the second semester of the 2018-2019 academic year. Accordingly, four schools were identified with easily accessible sampling method. Scales were applied to the fifth, sixth, seventh, and eighth grades randomly selected for each grade level from these schools and data were collected from individuals willing to participate voluntarily. The scale was applied to 480 students in four secondary schools and the data of 464 of them were analyzed. Findings were obtained in accordance with the analysis.

3.5 Data Analysis

The data obtained from the research were analyzed using SPSS 23 program. Histogram graphs, skewness-kurtosis coefficients were examined for the assumption of normality of self-efficacy perception scores related to block-based programming and CTLS scores. Accordingly, it was found that the skewness and kurtosis coefficients were between -2 and +2 (George & Mallery, 2010; Khan, 2015) and the variables did not deviate from the normal excess.

Independent samples t-test was used for gender, programming training, and Scratch and Arduino study situations outside the classroom. In order to determine the relationship between students' computational thinking skills and self-efficacy perceptions related to block-based programming, Pearson correlation coefficient was used since the data showed normal distribution.

4. Findings

4.1 Computational Thinking Levels of Secondary School Students

Within the scope of the research, in order to determine the computational thinking skills of 5th, 6th, 7th and 8th grade students, CTLS consisting of five factors and a total of 22 items was applied to the students. Descriptive statistics were calculated in accordance with the answers given by the students who participated in the study and the results are given in Table 2.

The raw scores obtained in Table 2 were calculated in accordance with the scoring formula specified by Korkmaz et al. (2015) and standard scores were obtained. Accordingly, secondary school students' creativity ($\overline{X}$ = 80.8), algorithmic thinking is ($\overline{X}$ = 72.3) and cooperativity ($\overline{X}$=78.8) critical thinking ($\overline{X}$=72.9) are at high levels while problem solving ($\overline{X}$=54.7) is seen to be at medium levels. However, it is observed that the students' general computational thinking skills ($\overline{X}$=70.4) score is observed to be high.

4.2 Comparison of Computational Thinking Skills of Secondary School Students According to Variables

The results of the analysis of students' computational thinking skills according to the variables, such as gender, programming education, using Scratch and Arduino platforms outside the school are shown in the tables.

Table 3 shows the t-test results of students' computational thinking skills according to gender. Accordingly, students' creativity [t(462) = 2.59; p <0.05], cooperativity [t(462) = 2.20; p<0.05], and problem solving [t(462) = 4.49; p<0.05] levels differ significantly according to gender. While female students had higher creativity ($\overline{X}$=16.56) than male students ($\overline{X}$=15.79) male students 'problem solving levels ($\overline{X}$=17.70) and cooperativity levels ($\overline{X}$=16.38) are higher than female students' problem solving levels ($\overline{X}$=15.06) and cooperativity levels ($\overline{X}$=16.10).

Table 4 shows the t-test results of students' computational thinking skills according to programming education. Accordingly, the students' computational thinking skills who took programming training differ significantly from students who did not take [t(462) = 2.28; p<0.05]. The average scores of computational thinking ($\overline{X}$=78.26) of students who received programming training were higher than those who did not receive ($\overline{X}$=75.01).

Table 5 shows the t-test results of students' computational thinking skills according to their working status in Scratch platform outside of the classroom. Accordingly, creativity [t(373) = 3.50; p<0.05], algorithmic thinking [t(388) = 5.32; p<0.05], cooperativity [t(402) = 3.18; p<0.05], critical thinking [t(462)=3.58;p<0.05] factors and computational thinking skills [t(462) = 4.76; p<0.05] of students who work Scratch Platform outside of the class differ significantly from students who do not work. Students working on the Scratch platform outside of the class had higher creativity ($\overline{X}$=16.64), algorithmic thinking ($\overline{X}$=15.56), cooperativity ($\overline{X}$=16.25), critical thinking ($\overline{X}$=15.11), and average scores for computational thinking ($\overline{X}$=79.98), than non-working students.

Table 6 shows the t-test results of the students' computational thinking skills according to their working situation on the Arduino platform. According to this, algorithmic thinking [t (462) = 2.53; p<0.05], problem solving [t (462) = 3.04; p<0.05] and computational thinking skills [t (462) = 2.71; p<0.05] of students working

Factors	Item Count	N	Min.	Max.	$\overline{X}$	SD
Creativity	4	464	4.00	20.00	16.17	3.23
Algorithmic Thinking	4	464	4.00	20.00	14.46	3.68
Cooperativity	4	464	4.00	20.00	15.76	3.81
Critical Thinking	4	464	4.00	20.0	14.58	3.66
Problem Solving	6	464	6.00	30.00	16.41	6.52
Computational Thinking	22	464	22.00	110.00	77.47	13.72

Table 2. Descriptive Statistics on Computational Thinking Skills of Students

Variable	Factors	Group	$\bar{X}$	SD	t	df	p
Gender	Creativity	Female	16.56	3.07	2.59	462	.010
		Male	15.79	3.35			
	Algorithmic Thinking	Female	14.30	3.65	.93	462	.351
		Male	14.62	3.72			
	Cooperativity	Female	16.15	3.73	2.20	462	.028
		Male	16.38	3.86			
	Critical Thinking	Female	14.77	3.53	1.09	462	.274
		Male	14.40	3.77			
	Problem Solving	Female	15.06	6.12	4.49	462	.000
		Male	17.70	6.64			
	Computational Thinking	Female	76.86	12.02	.841	462	.401
		Male	77.93	14.98			
		None	75.01	13.57			

Table 3. Computational Thinking Skills T-Test Results According to Gender Variable

Variable	Factors	Group	$\bar{X}$	SD	t	df	p
Programming Education	Creativity	Taken	16.28	3.17	1.18	462	.237
		Not Taken	15.87	3.41			
	Algorithmic Thinking	Taken	14.62	3.65	1.52	462	.129
		Not Taken	14.03	3.76			
	Cooperativity	Taken	15.90	3.79	1.27	462	.205
		Not Taken	15.39	3.88			
	Critical Thinking	Taken	14.76	3.55	1.76	462	.079
		Not Taken	14.08	3.89			
	Problem Solving	Taken	16.69	6.56	1.56	462	.118
		Not Taken	15.62	6.38			
	Computational Thinking	Taken	78.26	13.52	2.28	462	.024
		Not Taken	75.01	13.57			

Table 4. Computational Thinking Skills T-Test Results According to Programming Education

Variable	Factors	Group	$\bar{X}$	SD	t	df	p
Studying status of Scratch a partfrom lessons	Creativity	Yes	16.64	2.84	3.50	373	.001
		No	15.56	3.60			
	Algorithmic Thinking	Yes	15.25	3.27	5.32	388	.000
		No	13.43	3.93			
	Cooperativity	Yes	16.25	3.59	3.18	406	.002
		No	15.12	4.01			
	Critical Thinking	Yes	15.11	3.43	3.58	462	.000
		No	13.9	3.83			
	Problem-Solving	Yes	16.7	6.83	1.13	452	.259
		No	16.02	6.09			
	Computational Thinking	Yes	79.98	12.83	4.76	462	.000
		No	74.05	13.85			

Table 5. Computational Thinking Skills T-Test Results According to Scratch Study Situations Outside of Classroom

in Arduino platform outside the class differ significantly from non-working students. Students working on Arduino platform outside of the class had higher algorithmic thinking ($\bar{X}$=15.41), problem solving ($\bar{X}$=18.43), and average scores for computational thinking ($\bar{X}$=81.16) than non-working students.

4.3 Students' Self-Efficacy Perceptions regarding Block Based Programming

Within the scope of the research, in order to determine the selfefficacy perceptions of the students at 5th, 6th, 7th and 8th grade levels, self-efficacy perceptions related to block-based programming consisting of two dimensions and a total of 12 items were applied to the students. Descriptive analyzes were conducted in line with the answers of the students who participated in the research and the results are given in Table 7.

When the results in Table 7 are examined, it is seen that the average score of students for simple block based programming tasks is ($\bar{X}$=18.29) and that the scores for complex block based programming tasks are ($\bar{X}$=22.59). On the other hand, self-efficacy perceptions of the students based on block-based programming ranged from 12 to 60 and the mean ($\bar{X}$= 40.89).

4.4 Comparison of Perceived Self-Efficacy related to Block Based Programming of Secondary School Students according to Variables

The results of the analysis of students' self-efficacy regarding block-based programming according to variables, such as gender, programming education, and studying Scratch and Arduino platforms outside the school are shown in the tables.

Table 8 shows the t-test results of students' perceptions of self-efficacy regarding block-based programming according to gender variable. Students' simple block-based programming tasks [t (462) = 2.38; p<0.05] and complex block-based programming tasks [t (462) = 3.89; p<0.05] factors and self-efficacy perceptions of block-based programming [t(462) = 3.50; p<0.05] differ significantly according to gender. Male students' score of simple block-based programming tasks ($\bar{X}$= 18.80) and complex block-based programming tasks ($\bar{X}$=23.84) and self-efficacy perceptions related to block-based programming ($\bar{X}$=42.72) were higher than female students. Table 9 shows t-test results of students' perceptions of self-efficacy regarding block-based programming according to programming education. According to simple block-based programming tasks [t (462) = 5.64; p<0.05],complex block-based programming tasks [t (462) = 3.70; p<0.05] and self-efficacy perceptions of block-

Variable	Factors	Group	$\overline{X}$	SD	t	df	p
Studying status of Arduino apart from lessons	Creativity	Yes	16.45	3.15	.847	462	.398
		No	16.11	3.25			
	Algorithmic Thinking	Yes	15.41	3.20	2.530	462	.012
		No	14.27	3.75			
	Cooperativity	Yes	15.82	3.67	.147	462	.883
		No	15.75	3.85			
	Critical Thinking	Yes	15.03	3.40	1.200	462	.229
		No	14.49	3.70			
	Problem-Solving	Yes	18.43	7.00	3.040	462	.002
		No	16.00	6.35			
	Computational Thinking	Yes	81.16	15.02	2.710	462	.007
		No	76.63	13.17			

Table 6. Computational Thinking Skills T-Test Results According to Arduino Platform Study Situations Outside of Classroom

Factors	Item Count	N	Min.	Max.	$\overline{X}$	SD
Simple block based programming tasks	5	464	5	25	18.29	5.36
Complex block based programming	7	464	7	35	22.59	7.04
Self-Efficacy Perception on Block Based Programming	12	464	12	60	40.89	11.46

Table 7. Descriptive Statistics of Students' Perceived Self-Efficacy Regarding Block Based Programming

based programming [t (462) = 4.92; p<0.05], there are significant differences between the students who took programming training before and the students who did not take. Simple block-based programming tasks ($\overline{X}$= 19.11), complex block-based programming tasks ($\overline{X}$= 23.31) and self-efficacy perceptions of block-based programming ($\overline{X}$=42.43) scores of students who took programming training before are higher than the students who did not take.

Table 10 shows the t-test results of the students' perceptions of self-efficacy regarding block-based programming according to their studying status on Scratch platform. According to simple block-based programming tasks [t (386)=7.20; p<0.05], complex block-based programming tasks [t (462) = 5.70; p<0.05] and self-efficacy perceptions of block-based programming [t (462) = 7.04; p<0.05] scores, there are significant differences between students who studied Scratch platform outside of the course and students who did not study. Simple block-based programming tasks ($\overline{X}$=19.82), complex block-based programming tasks ($\overline{X}$=24.20), and self-efficacy

Variable	Factors	Group	$\overline{X}$	SD	t	df	p
Gender	Simple block-based programming tasks	Female	17.70	5.55	2.38	462	.018
		Male	18.80	5.11			
	Complex block based programming	Female	21.33	6.95	3.89	462	.000
		Male	23.84	6.92			
	Self-Efficacy Perception on Block Based Programming	Female	39.00	11.49	3.50	462	.000
		Male	42.72	11.16			

Table 8. T-Test Results of Self-Efficacy Perception related to Block Based Programming According to Gender

perceptions of block-based programming ($\overline{X}$=44.03) scores of students who studied Scratch platform apart from lessons are higher than students who did not study on Scratch platform apart from lessons.

Table 11 shows the t-test results of the students' perceptions of self-efficacy regarding the block-based programming according to the working situation on the Arduino platform. According to complex block-based programming tasks [t (462) = 2.82; p<0.05] and self-efficacy perceptions of block-based programming [t (462) = 2.43; p<0.05] scores, there are significant differences between students who studied Arduino platform outside of the class and students who did not study. Complex block-based programming tasks ($\overline{X}$=24.62) and block-based programming self-efficacy perceptions ($\overline{X}$=43.74) scores of students who studied Arduino platform apart from lessons are higher than students who did not study Arduino platform apart from lessons.

4.5 Findings on the Relationship Between Students' Computational Thinking Skills and Self-Efficacy Perceptions of Block Based Programming

Within the scope of the study, in order to determine the computational thinking skills and the self-efficacy perception related to block-based programming of 5^{th}, 6^{th}, 7^{th} and 8^{th} grade students, CTLS and Perceived self-efficacy scale related to block-based programming were used. Pearson correlation coefficient was used to determine the relationship between students' computational thinking skills and self-efficacy perceptions related to block-based programming and the results are given in Table 12.

Table 12 presents the Pearson Product Moment Correlation Analysis between students' computational thinking skills

Variable	Factors	Group	$\overline{X}$	SD	t	df	p
Programming Education	Simple block based programming tasks	Yes	19.11	4.99	5.64	462	.000
		No	16.03	5.71			
	Complex block based programming	Yes	23.31	7.02	3.70	462	.000
		No	20.60	6.74			
	Perceived self-efficacy related to block based programming	Yes	42.43	11.06	4.92	462	.000
		No	36.64	11.52			

Table 9. T-Test Results of Self-Efficacy Perception Related to Block Based Programming According to Programming Education

Variable	Factors	Group	$\overline{X}$	SD	t	df	p
Studying status of Scratch apart from lessons	Basic block based programming tasks	Yes	19.82	4.64	7.20	386	.000
		No	16.31	5.60			
	Complex block based programming tasks	Yes	24.20	6.64	5.77	462	.000
		No	20.51	7.02			
	Perceived self-efficacy related to block based programming	Yes	44.03	10.26	7.04	462	.000
		No	36.83	11.68			

Table 10. T-Test Results of Self-Efficacy Perception Related to Block Based Programming According to Studying Status of Scratch Apart from Lessons.

Variable	Factors	Group	$\overline{X}$	SD	t	df	p
Studying status of Arduino apart from lessons	Basic block based programming tasks	Yes	19.12	4.87	1.06	462	.133
		No	18.12	5.45			
	Complex block based programming tasks	Yes	24.62	6.59	2.82	462	.005
		No	22.18	7.07			
	Perceived self-efficacy related to block based programming	Yes	43.74	10.64	2.43	462	.015
		No	40.31	11.55			

Table 11. T-Test Results of Self-Efficacy Perception Related to Block Based Programming According to Studying Status of Arduino Apart from Lessons

and self-efficacy perceptions related to block-based programming. Accordingly, there are low positive correlations between students' cooperativity scores and complex block-based programming tasks scores [$r = .267$; $p<.05$], and problem-solving scores and self-efficacy perception scores related to block-based programming [$r = 149$; $p<.05$]. No significant relationship was found between students' problem-solving scores and simple block based programming scores ($p>.05$). However, a significant and positive correlation was found between the other dimensions. In other words, there is a significant positive and moderate relationship between students' computational thinking skills and self-efficacy perceptions related to block-based programming.

5. Results and Discussion

Computational thinking and programming skills have become two of the 21st century skills that are often referred to as highly regarded in the field of education. However, there are not enough studies in the literature to identify and develop computational thinking skills (Brennan & Resnick, 2012; García-Penalvo & Mendes, 2018). However, it is possible to come across research that advocates the idea that programming education is effective in computational thinking skills (Lye & Koh, 2017; Oluk & Korkmaz, 2016; Yıldız & Çiftçi, 2017). In this direction, this study aimed to contribute to the literature on computational thinking skills, and the relationship between computational thinking skills and perceived self-efficacy regarding block-based programming of the secondary school students were examined.

Students' creativity, algorithmic thinking, cooperativity, critical thinking and general computational thinking skills are high level. However, it was found that problem solving factor was moderate. Korkmaz et al. (2015) reported a similar situation in their study and found that the problem-

Computational Thinking	Self-Efficacy Perception According to Block Based Programming	N	r	p
Creativity	Simple block based programming	464	.324	.000
	Complex block based programming	464	.391	.000
	Perceived self-efficacy related to block based programming	464	.392	.000
Algorithmic Thinking	Simple block based programming	464	.435	.000
	Complex block based programming	464	.551	.000
	Perceived self-efficacy related to block based programming	464	.542	.000
Cooperativity	Simple block based programming	464	.306	.000
	Complex block based programming	464	.267	.000
	Perceived self-efficacy related to block based programming	464	.307	.000
Critical Thinking	Simple block based programming	464	.362	.000
	Complex block based programming	464	.438	.000
	Perceived self-efficacy related to block based programming	464	.439	.000
Problem Solving	Simple block based programming	464	.051	.270
	Complex block based programming	464	.204	.000
	Perceived self-efficacy related to block based programming	464	.149	.001
Computational Thinking	Simple block based programming	464	.403	.000
	Complex block based programming	464	.533	.000
	Perceived self-efficacy related to block based programming	464	.516	.000

Table 12. Pearson Product Moment Correlation Analysis Results of Students' Computation a Thinking Skills and Self-Efficacy Perceptions Related toBlock-Based Programming

solving factor of the students was moderate. Since the schools in which the study was conducted are located in the city center, the students' opportunities are higher than the other schools and the majority of them have received programming training, which may be one of the reasons why the scale scores of the students are high or medium level.

Computational thinking skills differed according to gender. According to this, female students in the study group are more creative than men, and male students have better problem solving and cooperativity skills than women. Depending on this result, creativity of male students, problem solving and cooperativity skills of female students should be improved. Kirmit et al. (2018), in their study at secondary school level, found that male students' creative thinking, algorithmic thinking, cooperativity and critical thinking averages were higher than women; found that the problem solving means of female students were higher than male students. The findings of the study differ from the current study findings in terms of factors. Among the reasons why the students' general computational thinking skills scores are similar, but differ in the sub-factors, the cultural structures such as methods of access to information in the provinces where the study was conducted, educational status of parents, socio-economic status, etc. may be different.

In Turkey, as in other countries in recent years, the importance is given to programing education at early age. Research results showed that programming education is important in terms of computational thinking skills. Computational thinking skills of students who receive programming education are higher than others. Parallel to this result, students who use Scratch platform, which is one of the block based programming languages and develop applications, have higher creativity, algorithmic thinking, cooperativity, critical thinking, and computational thinking skills. It is possible to come across many studies that support the findings and show that the use of Scratch is effective in the development of students' computational thinking skills in general and sub-factors (Fadjo, 2012; Kobsiripat, 2015; Kukul & Gökçearslan, 2014; Yüksekeltürk & Altıok, 2017). In addition, students who are interested in the Arduino platform and make applications apart from lessons also have higher algorithmic thinking, problem solving and computational thinking skills. Similarly, in the study conducted by Pala and Mıhcı-Türker (2019a), it was determined that the training using the Arduino IDE platform was effective on the creativity, algorithmic thinking, critical thinking sub-factors and general computational thinking skills of the teacher candidates. Gülbahar (2018) stated that robotic coding developed students' many skills, such as logical inquiry, algorithmic thinking, decomposition, abstraction, and generalization. Briefly, students develop all their skills in computational thinking. Similarly, Penmetcha (2012) states that robotic programs such as Arduino IDE are effective in improving computational thinking skills.

In addition to computational thinking skills, students' perceptions of self-efficacy regarding block-based programming were examined and significant results were obtained. According to this, male students have higher perceptions about block-based programming. In a study conducted by Pala and Mıhcı-Türker (2019b), it was found that programming education had an effect on the gender variable and that female students described programming as more difficult. Similarly, Carter & Jenkins (1999) state that there is a perception that women are weaker in programming education than men. However, it is possible to come across different studies suggesting that women have a weaker confidence in programming (Scragg & Smith, 1998). On the other hand, Isa and Derus (2017) stated that in their study on programming education and gender, males had less understanding of functions than females and females had less desire to learn programming languages than males. However, in the study conducted by Lau and Yuen (2009), it was determined that student gender had no effect on programming ability. As Isa and Derus (2017) point out, women may be reluctant to learn programming languages.

In addition, it was found that programming education was effective in students' self-efficacy perceptions related to block-based programming. Students who are trained in programming have higher self-efficacy perceptions related to block based programming. Rizvi et al. (2011) stated that students who have programming training have a high degree of self-efficacy. This finding is in line with the research results. It is thought that this situation may be due to the prejudices about programming and the perception of the programming process as difficult. Because students who do not take the programming course consider the process difficult. For this reason, it is important to change the prejudices of the students and to start the process with easy practices that increase the self-efficacy perceptions regarding block-based programming. As a matter of fact, it is possible to find studies in the literature that students' self-efficacy perceptions have changed after the block-based programming process (Bishop-Clark et al., 2007; Rizvi et al., 2011).

Students' use of Scratch and Arduino platform outside the classroom and developing applications in this environment are also effective in their self-efficacy perceptions and sub-dimensions related to block-based programming. This may be due to the fact that programming languages such as Scratch and Arduino are more concrete, easier and fun to use than traditional programming languages. As a matter of fact, in the study conducted by Ozoran et al. (2012), positive opinions such as "the program was entertaining, it made the programming visual, it helped to learn the concept of algorithm and it increased creativity" were reached for the Scratch platform. However, there is a moderate positive correlation between students' computational thinking skills and self-efficacy perceptions of block-based programming in almost all dimensions.

6. Suggestions

Based on all these results, it is seen that there is a need for applications that increase the problem-solving skills of the students. Male students should be developed in terms of creativity, female students in problem solving and cooperativity skills. All students should be provided with programming training to improve their computational thinking skills; students should be able to spend time outside the school and practice on Scratch and Arduino platforms. In subsequent studies, research on the causes of these results may be conducted.

In the findings related to the study, the problem solving levels of the students who developed applications with the Scratch platform outside the class and who were not interested in this subject were similar. Detailed research can be conducted in other studies on the causes of this finding and the causes can be examined.

There is a need to increase students' self-efficacy perceptions related to block-based programming, especially of female students. In addition to this, students should be allowed to practice outside the school on platforms such as Scratch and Arduino.

Conclusions

This study aimed to contribute to the literature on computational thinking skills, and the relationship between computational thinking skills and perceived self-efficacy regarding block-based programming of the secondary school students were examined. As a result of the study, computational thinking skills differed according to gender. According to this, female students in the study group are more creative than men, and male students have better problem solving and cooperativity skills than women. On the other hand, computational thinking skills of students who receive programming education are higher than others. Parallel to this result, students who use Scratch platform which is one of the block based programming languages and develop applications, have higher creativity, algorithmic thinking, cooperativity, critical thinking and computational thinking skills.

In addition to computational thinking skills, students' perceptions of self-efficacy regarding block-based programming were examined and significant results were obtained. According to this, male students have higher perceptions about block-based programming. Also, it was found that programming education was effective in students' self-efficacy perceptions related to block-based programming. Students who are trained in programming have higher self-efficacy perceptions related to block based programming. Students' use of Scratch and Arduino platform outside the classroom and developing applications in this environment are also effective in their self-efficacy perceptions and sub-dimensions related to block-based programming.

References

[1]. **Askar, P., & Davenport, D. (2009).** *An investigation of factors related to self-efficacy for Java Programming among engineering students.* Online Submission, 8(1).

[2]. **Bandura, A. (1977).** Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review,* 84(2), 191-215. https://doi.org/doi/10.1037/0033-295X.84.2.191

[3]. **Bishop-Clark, C., Courte, J., Evans, D., & Howard, E. V. (2007).** A quantitative and qualitative investigation of using Alice programming to improve confidence, enjoyment and achievement among non-majors. *Journal of Educational Computing Research,* 37(2), 193-207. https://doi.org/10.2190FJ8W3-74U6-Q064-12J5

[4]. **Brennan, K., & Resnick, M. (2012, April).** New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association,* (Vol. 1, p. 25).

[5]. **Büyüköztürk, Ş., Kılıç, Ç. E., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2018).** *Scientific Research Methods.* Pegem Citation Index, 1-360.

[6]. **Çatlak, Ş., Tekdal, M., & Fatih, B. A. Z. (2015).** Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3), 13-25.

[7]. **Carter, J., & Jenkins, T. (1999, June).** Gender and programming: What's going on? In *Proceedings of the 4th Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education* (pp. 1-4). https://doi.org/10.1145/384267.305824

[8]. **Czerkawski, B. C., & Lyman, E. W. (2015).** Exploring issues about computational thinking in higher education. *Tech Trends,* 59(2), 57- 65. https://doi.org/10.1007/s11528-015-0840-3

[9]. **Davidson, K., Larzon, L., & Ljunggren, K. (2010).** Selfefficacy in programming among STS students.

[10]. **Demir, Ö., & Seferoğlu, S. S. (2017).** Yeni kavramlar, farklı kullanımlar: Bilgi-işlemsel düşünmeyle ilgili bir değerlendirme. Eğitim teknolojileri okumaları içinde, 801-830.

[11]. **Demirer, V., & Nurcan, S. A. K. (2016).** Programming

education and new approaches around the world and in Turkey/Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama,* 12(3), 521-546.

[12]. Esteves, M., & Mendes, A. J. (2004, October). A simulation tool to help learning of object oriented programming basics. In *34th Annual Frontiers in Education* (pp. F4C-7). IEEE. https://doi.org/10.1109/FIE.2004.1408649

[13]. Fadjo, C. L. (2012). *Developing computational thinking through grounded embodied cognition* (Doctoral dissertation, Columbia University). https://doi.org/10.7916/D88058PP

[14]. García-Peñalvo, F. J., & Mendes, A. J. (2018), Exploring the computational thinking effects in pre-university education, *Computers in Human Behavior,* 80, 407-411 https://doi.org/10.1016/j.chb.2017.12.005 .

[15]. George, D., & Mallery, M. (2010). *SPSS for Windows Step by Step: A Simple Guide and Reference,* 17.0 (10th Ed.) Boston: Pearson.

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

[17]. Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher,* 42(1), 38-43. https://doi.org/10.3102%2F0013189X12463051

[18].Gülbahar, Y. (2018). Bilgiişlem seldüşün meden programlamaya. In Y Gülbahar (Ed.), *Bilgiişlem seldüşünmeve programlama konusundadeğişim vedönüşümler* (pp. 395-410). Ankara: Pegem Akademi. 16. Bölüm.

[19]. Guzdial, M. (2008). Paving the way for computational thinking. *Communications of the ACM,* 51(8), 25-27. https://doi.org/10.1145/1378704.1378713

[20]. Howland, K., & Good, J. (2015). Learning to communicate computationally with Flip: A bi-modal programming language for game creation. *Computers & Education,* 80, 224-240. https://doi.org/10.1016/j.compedu.2014.08.014

[21]. International Society for Technology in Education-ISTE. (2018). *Computational Thinking for All*.

[22]. Isa, N. A. M., & Derus, S. R. M. (2017). Students experience in learning Fundamental Programming: An analysis by gender perception. *Advanced Journal of Technical and Vocational Education,* 1(1), 240-248.

[23]. Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education,* 13(1), 33-50.

[24]. Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic J. Modern Computing,* 4(3), 583-596.

[25]. Kasalak, İ. (2017). *Effects of robotic coding activities on the effectiveness of secondary school students' self-efficacy and student experience about activities,* (Master Thesis, Hacettepe University,)

[26]. Karabak, D. & Güneş, A . (2013) OrtaokulBirinciSınıf Öğrencileriİç in Yazılım Geliştirme A lanın da Müfredat Önerisi. *Eğitimve ÖğretimA raştırmaları Dergisi* 2(3), 175-181.

[27]. Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory *computer programming. Procedia-Social and Behavioural Sciences,* 47, 1991-1999. https://doi.org/10.1016 /j.sbspro.2012. 06.938

[28]. Khan, A. F. (2015). *Assessment of midlife career stress on Indian managers.* (Master Thesis. Aligarh Muslim University).

[29]. Kirmit, Ş., Dönmez, İ., & Çataltaş, H. E. (2018). Üstün Yeteekli Öğrencilerin Bilgisayarca Düşünme Becerilerinin İncelenmesi. *Journal of STEM Education,* 1(2), 17-26.

[30]. Kobsiripat, W., (2015). Effects of the Media to Promote the Scratch Programming Capabilities Creativity of Elementary School Students. *Procedia-Social and Behavioral Sciences,* 174, 227-232. https://doi.org/10.1016/j.sbspro.2015.01.651

[31]. Korkmaz, Ö., Çakır, R., Özden, M. Y., Oluk, A., & Sarıoğlu, S. (2015). Bireylerin bilgisayarca düşünme becerilerinin farklı değişkenler açısından incelenmesi. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi,* 34(2), 68-87. https//doi.org/10.7822/omuefd.34.2.5

[32]. **Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2015).** Computational thinking levels scale adaptation for secondary school level. *Journal on Educational Science* 1(2), 143-162.

[33]. **Kukul, V., & Gökçearslan, Ş. (2014 September).** Investigatıng the problem solving skills of students attended scratch programming course. *8th International Computer & Instructional Technologies Symposium,* (pp.18-20), Trakya University, Edirne, Turkey.

[34]. **Lau, W. W. F., & Yuen, A. H. K. (2009),** Exploring the effects of gender and learning styles on computer programming performance: Implications for programming pedagogy. *British Journal of Educational Technology,* 40(4), 696-712. https://doi.org/10.1111/ j.1467-8535.2008.00847.x

[35]. **Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J. & Werner, L. (2011).** Computational thinking for youth in practice. *Acm Inroads,* 2(1), 32-37. https://doi.org/10.1145/1929887.1929902

[36]. **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. https://doi.org/10.1016/j.chb.20 14.09.012

[37]. **Mazman, S. G., & Altun, A. (2013).** The effect of Introductory to programming course on programming self-efficacy of CEIT students. *Journal of Instructional Technologies and Teacher Education,* 2(3), 24-29.

[38]. **Monroy-Hernández, A., & Resnick, M. (2008).** Feature empowering kids to create and share programmable media. *Interactions,* 15(2), 50-53. https://doi.org/10.1145/ 1340961.1340974

[39]. **Mıhcı-Türker, P., & Pala, F. K. (2018).** Opinions of secondary school students, teachers and parents about coding. *Elementary Education Online,* 17(4), 2013-2029.

[40]. **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,* 8(11), 1-7. https// doi.org/10.5815 /ijmecs.2 016.11.01

[41]. **Ozoran, D., Cagiltay, N., & Topalli, D. (2012)**. Using scratch in introduction to programming course for engineering students. In *2nd International Engineering Education Conference* (IEEC2012), (Vol. 2, pp. 125-132).

[42]. **Ozcinar, H., Yecan, E., & Tanyeri, T. (2016).** Öğretmen gözüyle görsel programlama öğretimi. *Proceeding Book,* 71.

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

[44]. **Pala, F. K., & Mıhcı-Türker, P. (2019b).** Öğretmen Adaylarının Programlama Eğitimine Yönelik Görüşleri. *Journal of Theoretical Educational Science*, 12(1), 116-134. https://doi.org/10.30831/akukeg.399921

[45]. **Penmetcha, M. R. (2012).** *Exploring the effectiveness of robotics as a vehicle for computational thinking* (Doctoral dissertation. Purdue University).

[46]. **Rizvi, M., Humphries, T., Major, D., Jones, M., & Lauzun, H. (2011).** A CS0 course using Scratch. *Journal of Computing Sciences in Colleges,* 26(3), 19-27.

[47]. **Sayın, Z., & Seferoğlu, S. S. (2016).** Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilişim Konferansı*, 3-5.

[48]. **Scragg, G., & Smith, J. (1998, March).** A study of barriers to women in undergraduate computer science. In *Proceedings of the twenty-ninth SIGCSE Technical Symposium on Computer Science Education* (pp. 82-86). https://doi.org/10.1145/273133.273167

[49]. **Shin, S., Park, P., & Bae, Y. (2013).** The effects of an information-technology gifted program on friendship using scratch programming language and clutter. *International Journal of Computer and Communication Engineering,* 2(3), 246-249. https://doi.org/10.7763/IJCCE.2013.V2.181

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

[51]. **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. https://doi.org/10.1098/rsta.2008.0118

[52]. Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education,* (pp. 465-470). https://doi.org/10.1145/1953163.1953297

[53]. Yıldız, M., & Çiftçi, E. (2017). BilişimselDüşünme and Programlama. In H. F. Odabaşı, B. Akkoyunlu, & A. İşman (Eds.), *Eğitimteknolojileriokumaları 2017* (Chapter 5, pp. 75-86). Adapazarı: TOJET and SakaryaÜniversitesi.

[54]. Yukselturk, E., & Altiok, 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. https://doi.org/10.1111/bjet.12453

[55]. Yükseltürk, E., & Altıok, S. (2018). Blok tabanlı programlama. *Pegem Atıf İndeksi,* 241-266.

ABOUT THE AUTHORS

Dr. Pınar Mıhcı Türker is currently working as an Assistant Professor in the Department of Computer Education and Instructional Technologies at Aksaray University, Aksaray, Turkey. She received her Ph.D Degree in Department of Computer Education and Instructional Technologies, Faculty of Education, Gazi University, Ankara, Turkey, Master's Degree in Department of Computer Education and Instructional Technologies Faculty of Education Ankara University, Ankara, Turkey, and Undergraduate Degree in Department of Computer Education and Instructional Technologies, Faculty of Education ,Gazi University, Ankara, Turkey. Her main areas of interest are Safe and Responsible Internet Use, Scale Development, Haptic Technologies in Education and Computational Thinking.

Dr. Ferhat Kadir Pala is working as an Assistant Professor in the Department of Computer Education and Instructional Technologies at Aksaray University, Ankara, Turkey. He worked as a visiting scholar in the Department of Educational Psychology and Learning Systems at Florida State University, Tallahassee, Florida and took part in some projects. He received his Ph.D Degree in Faculty of Education. and Instructional Technology Education, Hacettepe University, Ankara, Turkey. His Master's Degreeis in Department of Electronics and Computer Education, Faculty of Technical Education, Gazi University, Ankara, Turkey, and received his undergraduate degree in Department of Computer Education, and Instructional Technologies, Faculty of Education, Gazi University, Ankara, Turkey,