Educational Policy Analysis and Strategic Research, V 15, N 2, 2020

The Effect of Coding Education on 5th, 6th and 7th Grade Students' programming Self-Efficacy and Attitudes About Technology

Gözde OKAL[1]

University Name

Bekir YILDIRIM[2]

Muş Alparslan University

Serkan TİMUR[3]

Çanakkale Onsekiz Mart University

Abstract

This study aimed to determine the effect of teaching model developed on coding education on students' self-efficacy and attitude towards technology. The research was conducted on 64 students who were the 5th, 6th and 7th grades of a secondary school during the fall semester of 2018-2019 academic years. The research was designed in accordance with the exploratory sequential design from mixed research designs. The data were collected through the programming self-efficacy scale, the technology attitude scale and semi-structured interview form. The results of the analysis of quantitative data indicated that the model developed for coding education had a positive effect on students' programming self-efficacy and attitudes towards technology. It was found that the students expressed positive opinions about coding education. The students think that coding education facilitates the teaching of many different subjects such as mathematics and science. The students also think that they can do many activities such as code/program writing, designing games and robots, and solving problems with their coding education.

Keywords: Coding, Self-Efficacy, Technology, Attitude, Secondary School

DOI: 10.29329/epasr.2020.251.8

[1] Computer Engineer msc, MEF International School, Izmir, Turkey, ORCID: 0000-0002-7899-0070 Email: okalgozde@hotmail.com

[2] Assoc. Prof. Dr., Muş Alparslan University, Faculty of Education, Muş, Turkey, ORCID: 0000-0002-5374-4025 Email: bekir58bekir@gmail.com

[3] Assoc. Prof. Dr., Çanakkale Onsekiz Mart University, Çanakkale, Turkey, ORCID iD: 0000-0002-4949-2275

Correspondence: serkantimur42@gmail.com

Introduction

The introduction of the concepts of Industry 4.0 and Society 5.0, which emerged with the rapid change in technology, has changed the skills and equipment that countries expect from individuals. Countries need individuals who have critical thinking and problem-solving skills, technology literacy, leadership skills, flexibility and adaptability. In addition to these skills, countries expect individuals to have computational thinking skills, including algorithms, necessary for solving encountered problems (Wing, 2006). Computational thinking involves the use of algorithms in the solution of an existing problem (Choi, Lee & Lee, 2016). In addition, algorithmic thinking has an effect on individuals' ability to design and use the algorithm related to the problem. Ziatdinov and Musa (2012) stated that algorithmic thinking skill is to write necessary algorithm for the solution of the problem through logical reasoning about a problem. Therefore, coding skills, which are considered to be one of the main components of logical reasoning and regarded among 21st century skills, are also important for countries (European Commission, 2014).

Coding education is defined as a process that includes algorithm, connection between software and hardware, basic structures such as expressions and loops of programming languages (Yükseltürk & Altıok, 2016). In another definition, coding education involves writing codes via using program languages to solve a given problem (Arabacıoğlu, Bülbül & Filiz, 2007). However, the difficulty in understanding programming tools and programming languages, inadequate student self-efficacy and inadequate infrastructure make code writing process difficult (Byrne & Lyons, 2001; Futschek, 2006). In particular, writing text-based codes is one of the issues that students experience problems and have difficulty (Özmen & Altun, 2014). Difficulty in the code writing process makes it difficult to teach the logic of algorithm. In coding education, the important thing is to teach students the logic of algorithm (Ala-Mutka, 2004). Therefore, planning the process well, making programming languages more visual and interesting, facilitating, and eliminating infrastructure deficiencies will facilitate the provision of coding education.

Countries are carrying out studies in order to provide coding education to students in an easy way. As a result of these studies, countries have facilitated to created different programming languages such as “code.org, Scracth, MBlock” and integrated them into the curriculum. For example, in the studies conducted on code.org, tens of hours of coding curriculum was created in order to enable students to learn programming comfortably and the developed coding education were translated into 34 different languages to reach everyone (Code.org, 2015). The aim of programming languages such as Scracth, MBlock, code.org is to enable students to teach programming languages easily and enable students to practice with the learned information (Resnick et al., 2009). In this way, students can create their own games, animations and stories by using programming languages and logical reasoning (Taylor, Harlow & Forret, 2010).

With coding education, students create their own games, write codes related to a problem and contribute to the development of students' many different skills enabled researchers to conduct studies in this field. When the literature is examined, it is understood that many studies have been conducted for students. In these studies, it has been concluded that coding education enable students to be taught concepts related in mathematics and informatics, and students develop skills such as critical thinking, problem solving, creative and algorithmic thinking (Grover & Pea, 2013; Monroy-Hern´andez & Resnick, 2008; Oluk & Korkmaz, 2016; Penmetcha, 2012; Shin, Park & Bae, 2013).

In this context, in today's world where Community 5.0 is experienced, coding education has an important role in having necessary skills and equipment for the survival of individuals. Coding education is important to adapt to the community 5.0 process. Therefore, many countries such as USA, UK, Belgium, Spain have coding education in their curriculum. Countries have integrated coding education into teaching programs to develop problem solving, logical reasoning, computational and algorithmic thinking skills (Balanskat & Engelhardt, 2014). When the literature is examined, many studies on coding education have been made at national and international level. It is observed that the studies on coding education have increased especially in the last two decades. In these studies, the possible effects of coding education were examined on various dependent variables (e.g. academic achievement, problem solving skills, programming self-efficacy and opinions about coding) (Calder, 2010; Çetin, 2012; Shin & Park, 2014). However, there are no studies in which the effect of coding education on programming self-efficacy and technology attitudes are considered together and different class levels are examined comparatively. Moreover, when the studies on coding education are analyzed methodically, it is understood that they are done with qualitative or quantitative research methods. However, mixed method studies were not very common which use both quantitative and qualitative data. In this study, exploratory sequential design method was used among mixed method researches in order to reveal whether coding education is effective or not. The aim of this study is to determine the effect of teaching model developed for coding education on students' programming self-efficacy and attitudes towards technology. To this end, the following research questions were sought for the answer:

1. Is there a significant difference between the programming self-efficacy scores of 5th, 6th and 7th grade students in which the model developed for coding education is applied?

2. Is there a meaningful difference between the attitudes towards technology of 5th, 6th and 7th grade students in which the model developed for coding education is applied?

3. What are the opinions of 5th, 6th and 7th grade students regarding the coding education in which the model developed for coding education is applied?

Method

Research Design

Mixed method was used in the study. Mixed method research is a type of research in which qualitative and quantitative research approaches are combined together (Johnson et. Al., 2007). In this study, exploratory sequential design is used among the mixed research methods. In this pattern, quantitative data is primarily collected and analyzed. Then, qualitative data is collected and analyzed to support quantitative data. The results obtained from the quantitative and qualitative data are interpreted together. Within the scope of this study, primarily data were collected and analyzed through the quantitative data collection tools. Afterwards, next stage was processed in which qualitative data were collected and analyzed. Quantitative and qualitative data were interpreted together and the research results were presented. The quantitative dimension of the study was designed in accordance with the single-group pre-test and post-test quasi-experimental design. In this design, the independent variable is applied. In this design, measurements are made before and after the application (Cohen & Manion, 1997). This design is one of the weakest designs among the quasi-experimental designs (Yamak, Bulut & Dündar, 2014). However, the use of this design is appropriate to the nature of the research in studies where a new training module is developed and applied (Creswell, 2012). Within the scope of this study, an instructional model for coding education was developed and applied to the 5th, 6th and 7th grades. Therefore, due to the nature of this study, single group pre-test-post-test quasi-experimental design was used. The qualitative part of the study was designed in accordance with the phenomenology design. Phenomenology is preferred to obtain detailed information from the people who have past experience about the phenomenon or event being focused (Yıldırım & Şimşek, 2011). Within the scope of this study, it was aimed to reveal the opinions of secondary school students about coding education.

Research Group

The research group of the study consisted of 64 people attending 5th, 6th and 7th grades of a secondary school in Malatya during the fall semester of 2018-2019 academic year. The purposeful sampling method among the non-probable sampling methods and easily accessible case sampling among these sampling was selected to set the research group. In this sampling, the researchers decide who will be selected and choose the most appropriate sample for the purpose of the research (Balcı, 2016). In this sampling, it provides the researchers time and effort in order to select the most suitable group for which the study will be performed (Platton, 2002). Within the scope of this study, there are certain reasons for choosing an easily accessible sampling method. These are: (1) the implementation will take place during a semester, (2) providing the necessary infrastructure for coding education, and (3) meetings were hold with the school administration and parents, information was provided about the coding education and the agreement of school administration and parents to conduct this research are among the reasons.

Data Collection Tools

“Programming Self-Efficacy Scale”, “Attitude Scale towards Technology” and “Semi-Structured Interview Form” were used within the scope of the study. In this study, quantitative data collection tools are supported by qualitative data collection tools.

Programming Self-efficacy Scale (PSS)

“The Programming Self-Efficacy Scale” with 5-point Likert type items by Kukul, Gökçearslan and Günbatar (2017) was used within scope of the research. The scale was developed to measure the programming self-efficacy levels of secondary school students. The final version of the programming self-efficacy scale was 31-items one-factor model. For the validity and reliability of the scale, exploratory and confirmatory factor analyzes were performed. As a result of the analysis, the researchers found that the scale was a valid and reliable one to measure the programming self-efficacy levels of secondary school students. In the scope of this study, the reliability of the PSS scale was also examined. The Cronbach's alpha value of the scale was found to be “0.70” suffesting that that it can be used as a reliable instrument in this study. Sample items from the Programming Self-efficacy Scale are presented in Table 1.

Table 1. Programming Self-efficacy Scale sample items

	Items	Strongly Agree	Agree	Not Sure	Disagree	Strongly Disagree
1	I can understand that a confusing problem whether a programming problem or not.					
18	I can use a loop instead of repetitive commands.					
31	I can explain my software project step by step.					

Students' Attitude Scale towards Technology (SAST)

The SAST with 5-point Likert type items developed by Yurdugül and Aşkar (2008) was was developed to measure students' general attitudes towards technology. In order to raise the validity of SAST, exploratory and confirmatory factor analyzes were performed. As a result, 24-items four-factor model was obtained from exploratory and confirmatory factor analyzes. The Cronbach's alpha value of the scale was found to be “0.82”. these results indicated that the SAST can be used as a valid and reliable instrument to measure secondary school students’ attitudes towards technology. Sample items related to the SAST scale are given in Table 2.

Table 2. Sample items fromthe SAST

	Items	Strongly Agree	Agree	Not Sure	Disagree	Strongly Disagree
1	I will probably choose a technology-related profession.					
14	We must use less technology since it causes pollution.					
24	Everyone can be good in the field of technology.					

Semi-Structured Interview Form

In the study, semi-structured interviews were conducted to reveal the views of secondary school students about coding education. In the research, the purpose of including the semi-structured interview form is to provide the opportunity to the researchers to hold the course of the interview in their own hands (Merriam, 2009). “The Interview Form towards Coding Education (IFCE)” which was used in the interview of secondary school students consists of 5 questions and was formed by the researchers. After the first form of the 5-question included form, the researchers took the opinion of two experts in coding education. As a result of the interviews, the interview form was finalized. After the interview form was developed, the pilot implementation was first administered to 5 students. During the implementations, the students were asked whether there were unclear places in the form. After the pilot application, the interview form was finalized. Semi-structured interviews were conducted for two weeks after coding education. The interviews lasted approximately in 7 minutes for each student. Sample questions regarding the semi-structured interview form are given in Figure 1.

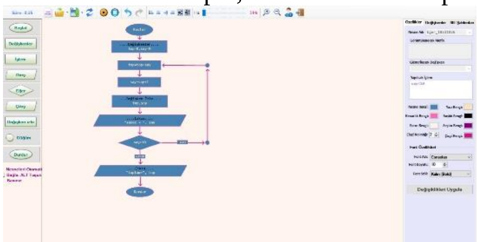
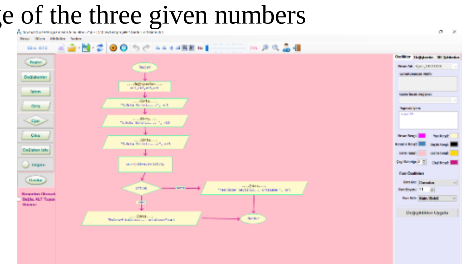
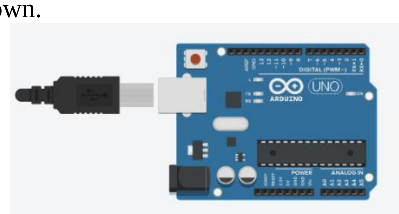
1. What do you think about coding education?
2. Which of the courses do you think that the coding education contribute?
3. What are the contributions of coding education to your courses?

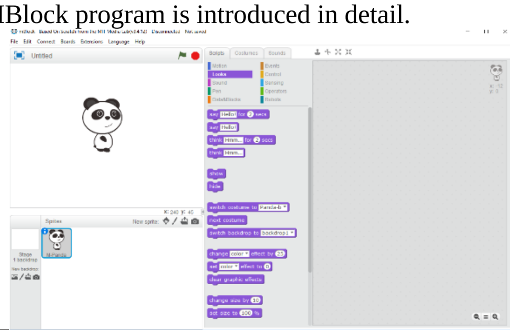
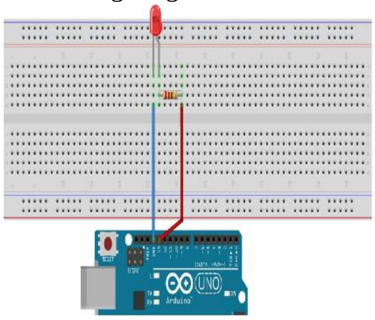
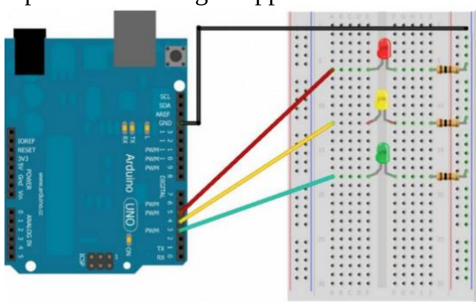
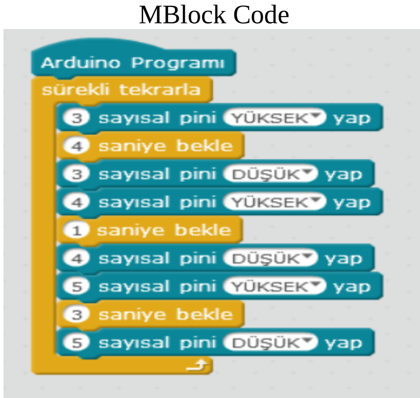
Figure 1. Sample Questions from Semi-Structured Interview Form

Teaching Model Application Process for Coding Education

The instructional model for coding education, which was established within the scope of the study, was designed and implemented based on the information technologies gains of Information Technology and Software course and the idea of Informatics Workshops mentioned in the 2023 Education vision report. The application process is detailed in below.

Table 3. Practical Procedures about Implementation

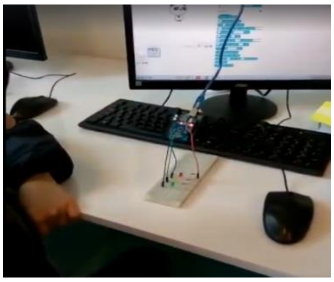
Practical Procedures		
Duration	Implementation of Developed Teaching Model for Coding Education	Pictures Related to Applications/Descriptions
Week 1	Implementation of pre-test	
Week 2	Explanation of problem concept	In order to improve our problem-solving ability, first of all, the definition of the problem concept has been made. A good understanding of the problem is very important as it will determine the steps of the created algorithm to solve the problem. While solving the problem; problem solving strategies are taught such as making inferences, tool-purpose analysis, disassembling, moving from part to whole and bringing solutions together.
Week 3	Explanation of Algorithms and Flowcharts	The benefits of algorithmic thinking are underlined by mentioning that we use algorithmic processes in many activities in our daily lives. In order to express the algorithms visually, flowcharts and their representations are taught.
Week 4	Algorithm and Flow Chart sample application	Algorithm and flow chart drawing for the sum of the given numbers. For example, sum of numbers up to 10. 
Week 5	Algorithm and Flow Chart sample application	Algorithm and flow chart drawing to calculate the average of the three given numbers 
Week 6	Sample application	Traffic crossing algorithm and flow chart drawing
Week 7 & 8	1. Programming languages, Block-based information is given, Code.org structure is mentioned.	code.org 
Week 9 & 10	Arduino trainings are provided to learn basic electronic knowledge.	In the content of Arduino training, the operating mechanisms of the sensors and their intended use are shown. 

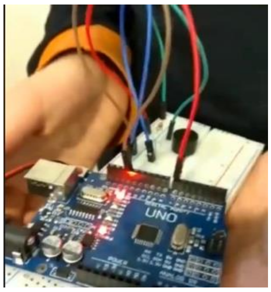
	MBlock Program	MBlock program is introduced in detail. 
Week 11-15	1. Led Lighting 2. Running Light Application 3. Traffic Light Application 4. Sound production with Buzzer 5. Musical Notes 6. Piano Making 7. Buzzer Application with Button 8. Police Siren and Strobe Light Application 9. Parking Sensor applications were made.	Example 1: Led Lighting  MBlock Code  Example 2. Traffic Light Application  MBlock Code 

Week 16	4. In our Robotics and Coding course to commemorate of acceptance of National Anthem at March 12, the National Anthem playing application was developed of waving flag with the servo motor that we can move at the desired angle with the help of the buzzer and the potentiometer.	
Week 17 & 18	Implementation of post-tests	1. Application of post-tests 2. Conducting semi-structured interviews

The implementation of the model developed for coding education has progressed as (1) preparation stage, (2) pre-tests implementation, (3) realization of the implementation process and (4) post-tests implementation. Before the codification education, the informatics classroom of the school was arranged, the tools and equipment required for the applications were provided and made ready for use. After the preparations were completed, pre-test applications were carried out in the first week, applications were carried out for the next 15 weeks and the last two weeks were finalized through making post-test implementations. The application photos are given in Figure 2.


Intelligence class


Running light application



National Anthem application

Waving flag application with servo motor

Figure 2. The application photos

Data Analysis

In the analysis of the quantitative data obtained within the scope of the research, SPSS was used (Version 22.0). Two steps were followed for analysis. Firstly, Kolmogorov-Smirnov test was used to determine whether the data sets showed homogeneous distribution. According to the results of Kolmogorov-Smirnov test, it was found that the difference scores obtained by subtracting the pre-test from the PSS and SAST post-test showed normal distribution. Moreover, the kurtosis and skewness coefficients of the difference scores are at the normal distribution level. Because of the homogeneous distribution of the obtained data, dependent groups t-test was used for dependent groups and One-Way ANOVA test was used to analyze whether the mean differences between the groups were statistically significant or not.

The data collected through the "Interview Form for Coding Education (IFCE)" developed by the researchers in the research was subjected to content analysis. As a result of the interviews, 448 minutes of audio recording was created. After the researchers transcribed the entire audio recordings, the data were coded separately by the researchers. In this coding process, parts of consensus and disagreement were determined. The coders were met again on the codes with disagreement and, the consensus was reached on the codes as a result of these meetings. In this study, coder reliability was calculated with the formula [(Consensus / "Consensus + Disagreement") * 100] (Miles & Huberman, 1994). The encoder reliability for this research was calculated as ((64/64 + 16) * 100) = 80%. As a result of the calculation, it can be stated that coding is reliable since it is over 80% compliance level stated by Miles and Huberma (1994).

Findings

One-Way ANOVA test was used to determine any significant difference between the pre-test and post-test mean scores of the PSS and SAST scale between 5th, 6th and 7th grades in which the developed model was applied for coding education. Descriptive statistics regarding the mean pre-test and post-test scores of the 5th, 6th, and 7th grades were given in Table 4.

Table 1. Descriptive Statistics of the 5th, 6th, and 7th Grade PSS Scale Pre-Test and Post-Test Scores

	Grade Level	N	M	SD
PSS pre-test	5	28	85.93	13.71
	6	20	92.75	23.33
	7	16	97.19	21.88
PSS post-test	5	28	132.46	14.02
	6	20	120.75	11.99
	7	16	108.31	20.51

According to the table 4, PSS pre-test mean score was the highest in 7th grade depending on grade level. On the other hand, the highest score was found to be in the 5th grade according to the mean scores of PSS post-test. One-Way ANOVA test was used to determine whether there was a statistically significant difference between grade levels. The results of one-Way ANOVA test for the PSS pre-test scores are presented in Table 5.

Table 2. One-way ANOVA test results related to the PSS Scale Pre-Test Scores of Grades

	Source of Variance	Sum of squares	df	Mean of squares	F	p	Significant difference
	Inter group	1392.955	2	696.478			
PSS pre-test	Inner group	22606.045	61	370.591	1.88	0.16	
	Total	23999.000	63				

When Table 5 was examined, it was found that there was no significant difference between the mean pre-test scores of the PSS according to grade level ($F(2,61) = 1.88$, $p > .05$). In other words, the programming self-efficacy of the classes was similar prior to implementation.

One-way ANOVA test was used to assess the significance of the difference among the mean scores of PSS post-test scores among the three classes in which the developed model was applied for coding education. The results of one-Way ANOVA for the PSS post-test scores are given in Table 6.

Table 3. One-way ANOVA test results related to the PSS scale post-test scores of the classes

	Source of Variance	Sum of squares	df	Mean of squares	F	p	Significant difference
	Inter group	6057.33	2	3028.66			5^h-6^h
PSS post-test	Inner group	14352.15	61	235.28	12.87	.000	5^h-7^h
	Total	20409.48	63				6^h-7^h

When Table 6 is examined, the PSS posttest scores differ significantly depending on the grade level ($F(2,61) = 12.873$, $p < .05$). In other words, PSS scale scores vary depending on grade level. Tukey test was used to find out which classes differ among the classes. According to Tukey test results, it was seen that the difference among 5^{th} grade (M = 132.46, Sd = 14.02) and 6^{th} grade (M = 120.75, Sd = 11.99) and 7^{th} grade (M = 108.31, Sd = 20.51). In addition, there is a differentiation between 6^{th} grade (M = 120.75, Sd = 11.99) and 7^{th} grade (M = 108.31, Sd = 20.51).

One-Way ANOVA test was used to determine any difference between pre-test and post-test mean scores of SAST among the 5^{th}, 6^{th} and 7^{th} grades in which the developed model was applied for coding education. Descriptive statistics regarding the pre-test and post-test mean scores of the 5^{th}, 6^{th}, and 7^{th} grades of the SAST scale are given in Table 7.

Table 4. Descriptive Statistics Regarding the Pre-test and Post-Test Scores of 5^{th}, 6^{th}, and 7^{th} SAST Scale

	Grade Level	**N**	**M**	**SD**
	5	28	92.61	11.22
SAST pre-test	6	20	91.10	10.47
	7	16	85.06	9.73
	5	28	100.21	12.01
SAST post-test	6	20	98.15	15.09
	7	16	85.13	14.61

When Table 7 is examined, it was seen that the pre-test and post-test mean scores of the SAST scale are highest in the 5th grade depending on the grade level.

One-way ANOVA test was used to determine the significance of the difference among the mean scores of pre-test scores of the SAST among the three classes in which the developed model was applied for coding education. One-Way ANOVA test results regarding the difference among the mean scores of SAST pre-tests are given in Table 8.

Table 5. One-way ANOVA test results related to SAST pre-test scores of the classes

	Source of Variance	Sum of squares	df	Mean of squares	F	p	Significant difference
SAST pre-test	Inter group	600.584	2	300.292	2.652	.079	
	Inner group	6907.416	61	113.236			
	Total	7508.000	63				

Table 8 shows that there is no significant difference between the pre-test scores of the SAST scale depending on the grade levels (F (2-61) = 2,652 p> .05). In other words, there was no significant difference between the attitudes of the classes towards technology prior to application.

One-way ANOVA test was used to determine the significance of the difference among the mean scores of post-test scores of the SAST scale among the three classes in which the developed model was applied for coding education. One-Way ANOVA test results related to the difference among the SAST post-test mean scores are given in Table 9.

Table 6. One-way ANOVA test results related to SAST post-test scores of the classes

	Source of Variance	Sum of squares	df	Mean of squares	F	p	Significant difference
SAST post-test	Inter group	2479.34	2	1239.67	6.61	.003	1-3 2-3
	Inner group	11427.01	61	187.32			
	Total	13906.35	63				

When Table 9 is examined, the SAST scale post-test mean scores differ significantly depending on the grade level (F (2,61) = 6,618, p <.05). In other words, attitude scores towards technology vary according to class level. According to the results of the Tukey test, which was done to find out the differences among the classes, there is a difference between the 5th grade (M = 100,21 S = 12,010) and the 7th grade (M = 85,13, S = 14,610) in addition to 6th grade (M = 98,15, S = 15,097) and 7th grade (M = 85,13, S = 14,610).

T-test was used for dependent groups in the comparison of the pre-test and post-test results of the PSS and SAST of 5th, 6th and 7th grades. The obtained results are given in Table 10.

Table 7. Indpendent sample t-test results of PSS and SAST scales

Grade	Scale	Test	N	M	SD	t	p
5th	PSS	Pre test	28	85.93	13.717	12.207	.000**
		Post test	28	132.46	14.022		
	SAST	Pre test	28	92.61	11.226	2.850	.008**
		Post test	28	100.21	12.010		
6th	PSS	Pre test	20	92.75	23.333	5.775	.000**
		Post test	20	120.75	11.991		
	SAST	Pre test	20	91.10	10.473	1.607	.125
		Post test	20	98.15	15.097		
7th	PSS	Pre test	16	97.19	21.882	1.721	.106
		Post test	16	108.31	20.512		
	SAST	Pre test	16	85.06	9.733	.015	.988
		Post test	16	85.13	14.610		

** Indicates that there is a significant difference.

Table 10 which summarized whether there are statistically significant differences between pre-test and post-test scores of two different scales (PSS and SAST) in terms of grade-level indicated that in both PSS and SAST, there were significant differences between 5th grade students' pre-test and post-test scores. However, a statistically significant difference between 6th grade students' pre-test and post-test scores was found for PSS but not found for SAST. The differences between 7th grade students' pre-test and post-test scores were not found to be statistically significant for both PSS and SAST.

Students' opinions about the model developed for coding education

The obtained data sets from the semi-structured interview form were analyzed in order to support the findings obtained from the analysis of the quantitative data. Based on the qualitative data obtained as a result of the analysis, separate codes and themes were created for 5th, 6th and 7th grades. In this context, first of all, the opinions of the students were asked about coding education. Information about this question is presented in Table 11.

Table 8. Student views on coding education

Theme	5th grade		6th grade		7th grade	
	Codes	**f**	**Codes**	**f**	**Codes**	**f**
Thoughts and Emotions	Teaching to everyone	9	Teaching to everyone	5	Teaching to everyone	4
	Funny	4	Language of the future	4	Compulsory	4
	Instructive	3	Compulsory	4	Instructive	2
	Imagination	2	Facilitating life	4	Contribution to the profession	1
	Problem solving	1	Problem solving	2	Funny	1
	Perspective	1	Instructive	1	Problem solving	1
	Sequential thinking	1			**Not tutorial*	1
	Language of the future	1				
	Making life easier	1				

* Contains negative code.

When Table 11 is examined, 5th grade students think that coding education should be taught to everyone, funny, instructive and develop imagination, while 6th grade students think that coding education should be taught to everyone, it is the language of the future, it makes life easier, it is problem solving and instructive. 7th grade students think that coding education should be taught to everyone, it must be compulsory and it is instructive. In addition, one student stated that coding training is not instructive. Some of the students' views on this question are given separately for 5th, 6th and 7th grades:

Some of the statements of 5th grade students:

S10: "Coding education is a lesson that should be given in all schools and I think that it contributes to other courses as coding is learned".

S14: "In the future, coding is the field of technology since everything will be controlled by technology and coding education must be given anywhere in Turkey".

S20: "Coding education allows us to look at life from a different perspective".

Some of the statements of 6th grade students:

S3: "It makes our lives easier and helps to teach courses such as mathematics".

S9: "Coding education is an important lesson just like Mathematics. Because it improves our problem-solving skills".

S12: "Coding education is necessary. In my opinion, all students are required to receive coding education. Because in the future we need a world where robots don't rule us, but we program them".

Some of the statements of 7th grade students:

S1: "I think we don't learn that much from coding education".

S4: "I think this education will be very useful if we choose this profession in the future".

S16: "It teaches us to put our lives into algorithms and do everything in order. Therefore, It should be compulsory in all schools".

After revealing the opinions of the students about coding education, the students were asked about the contribution of coding education on any courses. Information about the answers of the students is presented in Table 12.

Table 9. Student opinions about contribution of coding education on any curses

	5th grade		6th grade		7th grade	
Theme	Codes	f	Codes	f	Codes	f
Courses	Mathematics	23	Mathematics	14	Mathematics	14
	Turkish	4	Turkish	4	Science	13
	Science	3	Science	4	Turkish	7
	Social Science	2	English	2	Social Science	1
	English	1	Social Science	1	**No contribution*	1

* Contains negative code.

When Table 12 is examined, the 5th grade students think that the coding education contributes to Mathematics, Turkish, Science, Social Studies and English courses, the 6th grade students think that coding education contributes to Mathematics, Turkish, Science, English and Social Studies courses and 7th grade students think that it contributes to Mathematics, Science, Turkish and Social Studies courses. However, one student stated that coding education did not contribute to any course.

After explaining the courses that they think the coding education contributed to, the students were asked to explain the contribution of coding education to the courses. The answers given by the students to this question are given in Table 13.

Table 10. Information about the contribution of coding education to the courses

	5th grade		**6th grade**		**7th grade**	
Theme	**Codes**	**f**	**Codes**	**f**	**Codes**	**f**
Contribution to Courses	Problem solving	7	Subject teaching	8	Subject teaching	11
	Subject teaching	5	Problem solving	7	Sequential thinking	5
	Different perspectives	3	Easy comprehension	3	Rapid thinking	2
	Sequential thinking	3	Sequential thinking	2	Focusing	2
	Rapid thinking	2	Focusing	1	Question solving	1
	Easy comprehension	2	Question solving	1	Problem solving	1
	Focusing	1			**No contribution*	1
	Technology overview	1				

* Contains negative code.

When Table 13 is examined, the 5th grade students stated the contribution of coding education to the courses as the problem solving, subject teaching, different perspectives, sequential thinking; the 6th grade students stated the contribution of coding education to the courses as subject teaching, problem solving, easy understanding of the courses and sequential thinking; the 7th grade students stated the contribution of coding education to the courses as subject teaching, focusing on sequential and rapid thinking. On the other hand, a student thinks that coding education has no contribution. Some of the student statements regarding this question were:

Some of the statements of 5th grade students:

S4: "It improves the problem-solving skills in mathematics and closes our attention deficit in Turkish lessons at the same time".

S10: "I can think fast, find the right answer and solve the problem in different steps".

S22: "I learned technology through coding education and my perspective on technology has changed".

Some of the statements of 6th grade students:

S5: "It enables to teach issues about social science course. Because it teaches us directions, latitude and longitude".

S6: "It provides us with teaching the angles in mathematics and helps us to solve paragraph sentences in Turkish".

S12: "Improves our problem-solving skills. Explains how to sort events, explain, and teaches shapes".

S13: "It improves attention in problem solving and finding angle in lessons".

Some of the statements of 7th grade students:

S1: "I think it doesn't contribute".

S11: "It is very useful for teaching the subject of electricity within the scope of science lesson".

S13: "We can do the questions faster in math class".

After the opinions of the students about coding education and the contribution of coding education to the courses, they were asked what they could do with the information they learned about coding education. The answers given by students to this question are given in Table 14.

Table 11. Information about the things that can be done with coding education

	5th grade		6th grade		7th grade	
Theme	Codes	f	Codes	f	Codes	f
Use of Information	Writing code/program	11	Writing code/program	8	Writing code/program	9
	Tool design	11	Robot design	6	Robot design	4
	Robot design	8	Tool design	3	Game creating	2
	Problem solving	2	Game creating	3	Tool design	1
	Electrical circuit setting up	1	Making a traffic light	1	Sensor design	1
			Making job selection	1		

* Contains negative code.

When Table 13 is examined, 5th grade students think that they can write code/programs, design tools and robots, solve problems and set-up electrical circuit with coding trainings; 6th grade students think that they can make code/program writing, robot and tool design, game creation, traffic lamp making and profession selection with coding education; 7th grade students think that they can make code/program writing, robot, tool and sensor design with coding education. Some of the student statements regarding this question are given separately for the 5th, 6th and 7th grades:

Some of the statements of 5th grade students:

S13: "With coding, I can create a loop, solve the given problems and find the errors in the coding".

S16: "We can do many things with coding. For example; robot, computer".

S17: "I can make an electrical circuit with the help of coding".

Some of the statements of 6th grade students:

S2: In the future, I can write a code and make a robot for myself. I can make my life easier with coding".

S12: "I can advance in my classes and have a good profession in the future".

S15: "We can invent intelligent robots for service purposes".

Some of the statements of 7th grade students:

S8: "We can do things that make our daily life easier".

S14: "We can write many codes that will save lives".

S16: "I can design new games and do most of the things in coding field".

Discussion, Conclusion and Implementation

The study aimed to determine the effect of the teaching model developed for coding education on 5th, 6th and 7th grade students' programming self-efficacy and attitudes towards technology. It was found that there was a significant difference in favor of 5th grade among the programming self-efficacy scale scores of 5th, 6th and 7th grades in which the developed teaching model was applied for coding education. Moreover, there was a significant difference between 6th and 7th grade programming self-efficacy scale scores in favor of 6th grade. Through considering the pre-test and post-test programming self-efficacy scale scores of the 5th, 6th, and 7th grades, there was a significant difference in favor of the post-test between the pre-test and post-test programming self-efficacy mean scores of the 5th and 6th grades. When the post-test self-efficacy scale mean scores were examined in terms of the grades, it was seen that the mean scores of 5th grade students were higher than the scores of both 6th grade and 7th grade students (See Table 3). These results can be interpreted as providing coding education starting from early ages make positive effect on students' programming self-efficacy, but the positive effect of coding education on programming self-efficacy decreases with increasing grade level.

When the literature was examined, it was found that coding education had a positive effect on programming self-efficacy in a similar way with this study (Kasalak, 2017; Ramalingam & Wiedenbeck, 1998; Şahutoğlu, 2018; Yukselturk & Altiok, 2017). Mazman and Altun (2013) conducted a research examining the effect of the students who took and did not take a programming course on self-efficacy perceptions of programming. According to the results of the research, it was found that the students who had pre-experience in programming had higher levels of programming self-efficacy than those who had no pre-experience. As a matter of fact, in many studies, it has been found that students with prior programming experience have more programming self-efficacy similar to this study (Jegede, 2009; Ramalingam & Wiedenbeck, 1998; Resnick et al., 2009; Wiedenbeck, La Belle & Kain, 2004). However, as a result of this study, it was concluded that the average of the 5th grade students without programming background was higher than the 7th grade students with programming experience. Holden and Weeden (2003) explain this difference between students with prior programming experience and those with no prior experience. As the students come from different backgrounds, it has been shown that the difference in these backgrounds of students is reflected especially in the first programming education and this difference decreases in later programming education. The results obtained in this study support the results of Holden and Weeden (2003).

It was found that there was a significant difference in favor of Grade 5 and 6 among the mean scores of attitude scale towards technology of 5^{th}, 6^{th} and 7^{th} grades in which the developed teaching model was applied for coding education. However, it was found that there was no significant difference between attitudes towards technology when the 5^{th} and 6^{th} grades were compared. Pre-test and post-test SAST scale scores of 5^{th}, 6^{th} and 7^{th} grades were compared within groups. As a result of the comparison, it was found that there was a significant difference between the pre-test and post-test mean scores of the 5^{th} grade SAST scores in favor of the post-test. According to these results, it can be interpreted that having coding education from an early age had a positive effect on students' attitudes towards technology, but coding education did not have a positive effect on attitudes towards technology with the increase in grade level. As a matter of fact, Güden (2015) examined the attitudes of secondary school students towards technology and found that 5^{th}, 6^{th} and 7^{th} grade students had a more positive attitude towards technology than 8^{th} grade students. Furthermore, Gülden (2015) found that 5th grade students had higher attitudes towards technology than other grades.

As a result of the analysis of the qualitative data obtained within the scope of the study, 74 codes were obtained under 4 themes which includes 34 positive, 3 negative, and 37 various codes. First of all, thoughts and emotions of students about coding education were presented in this context. The students emphasized that coding education should be taught to everyone as a compulsory subject. Moreover, students think that coding education is instructive and entertaining, makes life easier, develops problem-solving skills and improves imagination. When the literature is examined, it is seen that coding education has a positive effect on the opinions of the students which consistent with this study (Çetin, 2012; Çoşar, 2013; Şahin & Namlı, 2017; Vatansever & Baltacı Göktalay, 2018). Sarıkaya (2018) examined the views of students about coding. At the end of the investigation, students think that coding education is fun, interesting, develops creativity and contributes to problem solving skills. Some students reported negative opinions about coding education. Similarly, Kaučič and Asič (2011), in their study with primary school students, stated that Scratch visual programming is instructive and engaging. These results are consistent with the results of this study. In another study, Cetin (2012) believes that coding education improve students' problem-solving skills. In fact, in many studies, it is stated that coding education enables students to improve their problem-solving skills (Calder, 2010; Shin & Park, 2014). However, there are some studies states that coding education has no effect on problem solving skills. As a matter of fact, Kalelioğlu and Gülbahar (2014) found that coding education did not make a statistically significant difference on the problem-solving skills of secondary school students.

Another result obtained within the scope of the study is the opinions of students about the courses contributing to the coding education. The students think that Mathematics, Science, Turkish, Social Sciences and English courses contributes to coding education. However, one student stated that coding education did not contribute to any course. The students were then asked about the contribution of coding education to the courses. The students expressed the contribution of coding education to the courses as problem solving, subject teaching, gaining different perspectives, sequential thinking, easy comprehension, focus, technology perspective and question solving. In addition, a student stated that coding education had no contribution to the courses. When the literature is examined, there are studies that are consistent with the results of this study and that coding education has a positive effect on the teaching of other subjects such as mathematics (Akpınar & Altun, 2014; Resnick & Ocko, 1990). Fessakis, Gouli and Mavroudi (2013) concluded that programming education was effective in acquiring mathematical concepts and developing problem solving skills in the study with pre-school students.

Finally, the students were asked about their thought on the things that they could do with the learned coding information. Students stated that they could write code/program, design robots, tools and sensors with coding trainings. Moreover, the students think that with coding education they can solve problems, make profession choices, make electrical circuits and traffic lights. When the literature is examined, it was found that there are results consistent with the results of this study (Kasalak, 2017; Türker and Pala, 2018). For example; Türker and Pala (2018) examined students' views on coding education. As a result of the study, students stated that they could write programs with coding education and could make games and robots. This result shows that students can use the learned coding information. In other words, it can be said that the students with increased coding knowledge has improved positively their programming self-efficacy.

In this context, the study was conducted with 64 students who were at 5th, 6th and 7th grades. Within the scope of the study, one class was taken from each grade level and pre-test and post-test single group experimental design was used. further studies can be conducted at the same grade levels and the effects of coding education can be revealed by comparative analysis through adding a control group. This study was carried out during 18 weeks in the fall semester of 2018-2019 academic year. After that, the studies can be carried out in long-term and focus on variables such as attitudes which changes in a long time. Moreover, in this study, the effect of coding education on different variables can be examined except the various dependent variables included in the study. According to the results obtained in this study, it can be said that the teaching model prepared in accordance with coding education will be a guide for teachers. At this point, it is important that teachers and educators who will include coding education in their courses develop content for coding education and apply it for long periods.

References

Akpınar, Y., & Altun, A. (2014). The need for programming training in information society schools. *Elementary Education Online, 13*(1), 1–4.

Ala-Mutka, K. (2004). *Problems in learning and teaching programming*. Retrieved from: https://www.cs.tut.fi/~edge/literature_study.pdf

Arabacıoğlu, C., Bülbül, H., & Filiz, A. (2007). *A new approach to teaching computer programming.* Paper presented at the Academic Computing Conference, the Kutahya, Turkey.

Balanskat, A., & Engelhardt, K. (2014). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe.* European Schoolnet. Retrieved from: http://www.eun.org/resources/detail?publicationID=481

Balcı, A. (2016). *Research methods techniques and principles in social sciences.* Ankara: Pegem Academy Publishing.

Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. *ACM SIGCSE Bulletin, 33*(3), 49-52. https://doi.org/10.1145/377435.377467

Calder, N. (2010). Using scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom, 15*(4), 9–14.

Choi, J. Lee, Y., & Lee, E. (2016). Puzzle based algorithm learning for cultivating computational thinking. *Wireless Personal Communications, 93*(1), 131-145. https://doi.org/10.1007/s11277-016-3679-9

Code.org (2015a). *The hour of code is here.* Retrieved from: https://code.org/

Cohen, L. & Manion, L. (1997). *Research methods in education (4th ed.).* Routledge: London and New York.

Creswell, J.W. (2008). *Educational research planning, conducting and evaluating quantitative and qualitative research*. Upper Saddle River, N.J., Pearson/Merrill Prentice Hall.

Çetin, E. (2012). *The effect of computer programming education on children's problem solving skills.* (Unpublished master's thesis). Gazi University: Ankara.

Coşar, M. (2013). *Effects of computer programming studies on academic success, critical thinking skills and programming-based attitudes in problem-based learning environment.* (Unpublished PhD dissertation). Gazi University: Ankara.

European Commission (2014). *Coding-the 21st century skill.* European Commission. Retrieved from: https://ec.europa.eu/digital-single-market/coding-21st-century-skill

Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. R. T. Mittermeir (Ed.), *Informatics Education – The Bridge between Using and Understanding Computers* (C. 4226, s. 159-168). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11915355_15

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

Güden, C. (2015). *Examining secondary school students' cognitive process skills and attitudes towards science and technology course (Çanakkale sample).* (Unpublished master's thesis). Canakkale Onsekiz Mart University, Canakkale.

Holden, E., & Weeden., E. (2003). *The impact of prior experience inan information technology programming course sequence*. In Proceeding of the 4th conference on Information technology curriculum, pages 41–46. Indiana: USA.

Jegede, P.O. (2009). Predictors of java programming self-efficacy among engineering students in a Nigerian university. *International Journal of Computer Science and Information Security, 4*(1 & 2). https://arxiv.org/abs/0909.0074v1

Johnson, R. B., Onwuegbuzie, A. J. & Turner, L. A. (2007) Toward a definition of mixed methods research. *Journal of Mixed Methods Research, 1,* 112-133.

Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education, 13*(1), 33–50.

Kasalak, İ. (2017). *Effects of robotic coding activities on the effectiveness of secondary school students 'self-efficacy and student experience about activities. (*Unpublished master thesis). Hacettepe University, Ankara.

Kaučič, B., & Asič, T. (2011). *Improving introductory programming with Stratch?*. Paper presented at 34th International Convention Conferance, Opatija, Croatia.

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

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

Merriam, S., B. (2009). *Qualitative research. A guide to design and implementation, (2nd e.d.).* San Francisco, CA: Jossey-Bass.

Miles, M.B., & Huberman, A.M. (1994). *Qualitative data analysis (2nd edition).* Thousand Oaks, CA: Sage.

Monroy-Hernández, A., & Resnick, M. (2008). Empowering kids to create and share programmable media. *Interactions, 15*(2), 50-53.

Namlı, N. A., & Şahin, M. C. (2017). The Effect of algorithm training on problem solving skills. *Recep Tayyip Erdogan Universtiy Journal of Social Sciences, 3*(5), 135-153.

Oluk, A., & Saltan, F. (2015). Effects of using the scratch program in 6th grade information technologies courses on algorithm development and problemsolving skills [Special issue]. *Participatory Educational Research*, 10-20.

Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry, 5*(3), 1-27.

Patton, M. (2002). *Qualitative evaluation and research methods.* Beverly Hills, CA: SAGE.

Penmetcha, M. R. (2012). *Exploring the effectiveness of robotics as a vehicle for computational thinking*. (Unpublished doctoral dissertation). Purdue University: Purdue.

Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research, 19*(4), 367–381. https://doi.org/10.2190/C670-Y3C8-LTJ1-CT3P

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11).

Resnick, M., & Ocko, S. (1990). *LEGO/Logo: learning though and about design*. Retrieved from: https://llk.media.mit.edu/papers/ll.html

Sarıkaya, M. (2018). Student views on coding training. *Ondokuz Mayis University Journal of Education Faculty, 37*(2), 79-90.

Shin, S., & Park, P. (2014). A study on the effect affecting problem solving ability of primary students through the Scratch programming. *Advanced Science and Technology Letters*, *59*, 117–120.

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.

SPSS Statistics (Version 22) [Computer software]. Chicago, IL: SPSS.

Şahutoğlu, N. G. (2018). *The effects of using EBA coding module on middle school learners beliefs about their self-efficacy on programming and the views of learners about the module.* (Unpublished master's thesis). University of Gaziantep, Gaziantep.

Taylor, M., Harlow, A., & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia - Social and Behavioral Sciences, 8*, 561- 570.

Türker, P. M., & Pala, F. K. (2018). Ortaokul Öğrencilerinin, Öğretmenlerin ve Öğrenci Velilerinin Kodlamaya Yönelik Görüşleri. *İlköğretim Online,* 17(4), 2013- 2029.

Wiedenbeck, S., LaBelle, D., & Kain, V. (2004). Factors affecting course outcomes in introductory programming. In Proceedings of the 16th Workshop of the Psychology of Programming Interest Group (pp. 97–110). Carlow, Ireland. http://www.ppig.org/papers/16th-wiedenbeck.pdf

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

Vatansever, Ö., & Baltacı Göktalay, Ş. (2018). How does teaching programming through scratch affect problem-solving skills of 5th and 6th grade middle school students?, *International Journal of Eurasia Social Sciences*, *9*(33), 1778-1801.

Yıldırım, A., & Şimşek, H. (2011). *Qualitative research methods in the social sciences.* Ankara: Seckin Publishing.

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

Yurdugül, H., & Aşkar, P. (2008). *An investigation of the factorial structures of pupils' attitude towards technology (patt): A Turkish sample*. *Elementary Education Online, 7*(2), 288-309. http://dergipark.org.tr/ilkonline/issue/8601/107109

Ziatdinov, R., & Musa, S. (2012). Rapid mental computation system as a tool for algorithmic thinking of elementary school student's development. *European researcher, Series A,* (7), 1105-1110.