Analysing K12 Students' Self-Efficacy Regarding Coding Education

Fatih Soykan [1], Sezer Kanbul [1]

[1]*Near East University, Nicosia, Cyprus*

Abstract **– This study aims to analyse a group of k12 students' self-efficacy regarding coding. The students are receiving "Coding Education" at a private school via Code.org as part of an elective course. The "Coding Self-Efficacy Scale" was used in the study. The scale which consists of 31 items is a 5-point Likert-type scale. The study is a descriptive survey model. The study's experimental group consisted of k12 students. The research sample is comprised of 193 students receiving coding education. Results will be derived from arithmetic means, standard variation, mode, median, frequency and t analyses of the collected data. At the end of the study, it was identified that the students taking coding education had a higher self-efficacy compared to the students who had not received coding education and that more than half of the students not taking coding education stated that they would like to take this course. It is believed that the study findings will provide an insight for other researchers working in the area of students' self-efficacy regarding coding.**

Keywords **– Algorithmic thinking, Coding education, Computer programming, Problem-solving skills.**

1. Introduction

In a rapidly changing and developing world, it is necessary to teach young generations how to create new programmes rather than exhausting the existing ones. To be a productive country in a technologically focused world individuals' coding skills become more significant [1,4]. The use of coding in education started in the 60s with a coding language called Logo [2]. And the programming languages have been revived in recent years with visual programmes like Scratch, Alice, Kodu, and code.org. These platforms enable students to code without learning complex traditional programming languages [13,16].

According to a study conducted by the "European School Network" with the participation of 21 separate countries in 2015; there are 18 European countries considering to include coding education into their curriculum [3]. The process students experience while learning coding also supports them in learning many other subjects and skills. Students do not merely learn how to code, they can also use coding to learn [14]. The current expectation from students regarding knowledge and behaviour is not what it used to be. As a result of a series of researches conducted by scientists, the skills expected of the students in the 21st century are qualities like critical thinking and problem-solving, creativity and renewal, communication and collaboration, flexibility and adaptation, knowledge, media and technology literacy, and it is argued that these skills need to be taught to students [8].

The most prominent organisation for teaching students these 21st-century skills is Code.org, which was established as a non-profit organisation aiming to promote computer sciences education (Code.Org, 2017). Code.org, which follows an open source policy, on their official webpage (http://code.org/) tries to teach students and whoever wants to learn the basic concepts of programming like until, if-else, while, functions via various game scenarios. Visitors to the page are asked to complete various tasks to pass levels using the drag-and-drop method. Students are given a certificate when they complete all stages. Throughout the education period, the participants are guided with videos presented by famous people and there are 53 language options including Turkish (Code.org, 2017). In order to join Code.Org one needs to register to the site, however, there are also courses available which can be joined without registering. The site which has teacher and student enrolment options enables teachers to enrol their students and monitor their progress. Code.Org uses programming tools like Alice, Scratch, Robomind,

DOI: 10.18421/TEM71-22
https://dx.doi.org/10.18421/TEM71-22

Corresponding author: Sezer Kanbul,
Near East University, Nicosia, Cyprus
Email: sezer.kanbul@neu.edu.tr

Received: 06 December 2017.
Accepted: 31 January 2018.
Published: 23 February 2018.

MIT App Inventor, Robomind, Kodu Game Lab and Light Bot. Individuals who want to support starting coding education at a young age can sign the statement available on the site.

Code.org has invited each K-12 level student in San Francisco to join a campaign called "Hour of Code" (Code.Org, 2014). The campaign's vision is to provide all students at all schools with the opportunity to learn computer programming (Code.org, 2017). Hour of Code, a set of one-hour computer lessons accessible to anyone wanting to learn the basics of programming, is a global movement which has reached 10 million students in over 180 countries (Hour of Code, 2017). The one-hour lessons are prepared by using the characters in the game Angry Birds. It is aimed for children to easily learn and recycle the basic structures and instructions like turn left-right while having fun. This campaign was launched to simplify programming from being a difficult and boring task and to provide anyone wanting to learn it with the basic skills (Hour of Code, 2017). Every student should have the opportunity to learn computer science and this helps students improve their problem-solving skills, their reasoning and their creativity. By starting coding education at an early age, students will establish a solid foundation for any 21st-century career.

Within this context, the self-efficacy levels of students learning coding via code.org as an extracurricular activity, the self-efficacy levels of students not taking coding and, also, the self-efficacy levels of students who would like to take coding were analysed; all students were k12 students attending a private school. This study is important as it emphasises the significance of students receiving a coding education.

2. Method

2.1. Research Design

The study was designed using the screening method, which is one of the descriptive survey models. "The descriptive survey model is the screening of a group, specimen, or sample taken from a population consisting of a multitude of components or a part of it, to reach a general judgment about that population" [11].

2.2. Purpose

The aim of this study is to compare students' – who are receiving coding education – self-efficacy in programming to those who are not. In order to identify this comparison, the following sub-aims were comprised;

1. Is there a meaningful difference between participants in regard to;
   1.1. gender,
   1.2. age group,
   1.3. receiving coding education or not,
   1.4. the type of computer mainly used at home and
   1.5. the average time spent on a computer at home?
2. Is there a meaningful difference between the participants taking the coding course, in regard to;
   2.1. the medium,
   2.2. the type of computer used for coding and
   2.3. how much time is spent on coding out of class
3. Is there a meaningful difference between the descriptive statistics of scores obtained from the self-efficacy scale of students taking the coding course and of students who were not?
4. Is there a meaningful difference between participants' scores obtained from the coding self- efficacy scale, in regards to;
   4.1. gender,
   4.2. age and
   4.3. whether they are taking a coding course?

2.3. Population – Sample

The population and sample of the study consist of K12 (k6 and k7) sixth and seventh-grade students studying at a private k12 school during the 2016-2017 academic year. The study was conducted with 193 students selected using the "simple random sampling" method.

2.4. Data Collection Tools

The data collection tools of the study were a personal information survey created by the researcher and the "Coding Self-Efficacy Scale" developed by Kukul, Gökçearslan and Günbatar [12].

The personal information survey was administered to identify the students' descriptive features like their age, gender, the type of computer mainly used at home, the average time spent on a computer at home as well as whether taking a coding course or not, the medium used for the coding course, the type of computer used and how much time is spent on coding out of class.

The Coding Self-Efficacy Scale is a 5-point Likert-type scale consisting of 31 statements. The scoring for the responses to the scale is as "1 point for I totally disagree" and "5 points for I totally agree". The criterion specified to identify the students' coding self-efficacy levels according to scores is illustrated in Table 1.

Table 1. Coding self-efficacy scale levels

Level	Range
Low Self-Efficacy	1-2,49 points
Average Self-Efficacy	2,50-3,49 points
High Self-Efficacy	3,50 and more points

In their study regarding validity-reliability Kukul, Gökçearslan and Günbatar [12] state that the variance value of the scale is 41,15% and the Cronbach's alpha coefficient is 0,95. In this study, the Cronbach's alpha coefficient was calculated as 0,80. Furthermore, before administering, the scale was moderated by 4 academics who were experts in their field. According to these analyses and the feedback received from the academics, the scale was considered to be both valid and reliable.

As the first step of the data collection process, permission was obtained from the school administration, where the study was going to be conducted, to administer the data collection tools to their students. Later, the data collection tools were administered by eight (8) computer instructors and statements which the students had difficulty understanding were clarified by the instructors, helping students understand the statements correctly. The data collection tools were administered to a total of 218 students. While entering the data electronically not fully and/or randomly completed surveys (all same distractors or choosing distractors to form patterns) were not included in the analysis and thus the data analysis was carried out with 193 usable surveys.

2.5. Data Analysis

After the data obtained from the data collection tools was entered into a computer, the Statistical Package for the Social Sciences (SPSS) 24.0 for Windows Evaluation version was used to statistically analyse the data. The data on personal information obtained as part of the study was submitted to a frequency analysis and frequency distribution tables were created.

In order to determine which tests to use in the statistical analysis, and therefore to examine whether the data set had a normal distribution the Kolmogrov-Smirnov test of normality was applied. As a result of the test, it was determined that the data set showed a normal distribution, and then, the parametric tests were applied. T-test, which is a type of parametric tests, was employed to compare students' personal information to their responses to the coding self-efficacy scale.

3. Results

This section has been designated to statistical analysis in order to investigate the answers to the study's aim and sub-aims.

Table 2. Characteristics according to random variables

	Number (n)	Percent (%)
Gender		
Female	102	52,8
Male	91	47,2
Age Group		
11	126	65,3
12	67	34,7
Coding Education		
Students taking a Coding Course	63	32,6
Students not taking a Coding Course	130	56,9
Students who want to take a Coding Course	78	70,9
Type of Computer Mainly Used at Home		
Desktop	11	5,7
Tablet	97	50,3
Smartphone	80	41,5
Not Applicable	5	2,6
Average Time Spent on a Computer at Home (per day)		
Between 0-1 hours	23	11,9
Between 1-2 hours	58	30,1
More than 2 hours	107	55,4
Not Applicable	5	2,6
Total	193	100

It can be seen from Table 2 that there is not a major difference between the number of female and male students and that the majority of students are 11 years of age. Moreover, students taking a coding course were less than the students who were not taking a coding course, yet the majority of them wanted to take such a course. The reason behind these figures is that coding is not a compulsory course, and thus, is treated as an in-service training activity. Besides, it can be determined that at home students mainly prefer mobile computers (tablets and smartphones) and that most of the students spend more than 2 hours on these devices per day.

Table 3. Various characteristics of students taking a coding course

	Students Taking a Coding Course	
	N	%
Medium Used for Coding Education		
Code.org	63	100
Scratch		
Other		
Not Applicable		
Type of Computer Used for Coding Education		
Desktop	63	100
Tablet		
Smartphone		

Not Applicable		
Time Spent on Coding Out of Class		
Between 0-1 hours	35	55,56
Between 1-2 hours	8	12,70
More than 2 hours	5	7,94
Not Applicable	15	23,81
Total	63	100

According to Table 3, all the students taking a coding course used the same internet site, code.org. Furthermore, it can be seen that all students used a desktop computer for coding. Besides, a majority of the students spent between 0-1 hour on coding out of the class. According to these findings, it can be said that students learnt coding mainly in the class.

The descriptive statistics of the scores students obtained from the coding self-efficacy scale are presented in Table 4.

Table 4. Descriptive statistics of student scores obtained from the coding self-efficacy scale

	N	$\bar{x}$	**s**	**Min**	**Max**
Students Taking a Coding Course	63	3,13	0,42	2,42	4,13
Students not Taking a Coding Course	130	2,21	0,30	1,58	2,84
Overall	193	2,51	0,55	1,58	4,13

**p<0,05*

The students' overall average score from the scale is 2,51 ± 0,55. This range signifies an average self-efficacy level. Thus, it can be said that generally, the students have an average self-efficacy coding level. When students' taking a coding course and students who were not, are anaylsed, the average score for the students taking a coding course is 3,13±0,42. This range signifies an average self-efficacy level. Thus, the students who take a coding course have an average self-efficacy coding level. The scores of the students not taking a coding course is 2,12±0,24. As this range signifies a low self-efficacy coding level, it can be said that the students who do not take a coding course have a low self-efficacy coding level.

Table 5 illustrates the descriptive statistics and the comparison of the independent t-test student overall scores obtained from the coding self-efficacy scale, according to gender and whether taking a coding course or not.

Table 5. Independent group t-test results to determine whether coding self-efficacy scale scores differed according to the gender variable

	Groups	***N***	$\bar{x}$	**s**	***t* -Test**		
Students Taking a Coding Course	Female	34	3,17	0,44	*t*	*Sd*	*p*
	Male	29	3,09	0,40	0,75	61	0,45
Students not Taking a Coding Course	Female	88	2,21	0,04			
	Male	62	2,21	0,02	0,06	128	0,95
Overall	Female	102	2,53	0,57			
	Male	91	2,49	0,53	0,47	191	0,63

**p<0,05*

The study findings suggest that there was neither a meaningful statistical difference (p<0,05) in the scores obtained from the self-efficacy scale between students taking a coding course and those who were not, nor was there a meaningful statistical difference in their overall scores. Hence, it can be said that students' coding self-efficacy level is not gender dependent.

Table 6. Independent group t-test results to determine whether coding self-efficacy scale scores differed according to the age variable

	Groups	***N***	$\bar{x}$	**s**	***t* -Test**		
					t	*Sd*	*p*
Overall	11 year old	126	2,57	0,62	2,17	186,20	0,03
	12 year old	67	2,41	0,38			

**p<0,05*

It was identified that there was a meaningful statistical difference (p<0,05) in the scores obtained from the self-efficacy scale according to students' age. The 11-year-old group students obtained a significantly higher overall grade from the scale compared to the students in the 12-year-old group.

Table 7. The comparison of scores obtained from the coding self-efficacy scale according to whether students were taking a coding course or not

	Groups	***N***	$\bar{x}$	**s**	***t* -Test**		
					t	***Sd***	***p***
Overall	**Students Taking a Coding Course**	63	3,13	0,42	15,53	93,09	0,0
	Students not Taking a Coding Course	130	2,21	0,26			

According to Table 7, it was identified that there was a meaningful statistical difference ($p<0,05$) in the scores obtained from the self-efficacy scale according to whether students were taking a coding course or not. Students taking a coding course obtained a significantly higher overall grade from the scale compared to the students not taking a coding course.

4.Discussion and Conclusion

In this study, which aimed to determine students' coding self-efficacy levels according to whether taking a coding course or not, the findings are presented and discussed.

Students aged 11-12 (%70.9) stated that they wanted to take a coding course. This finding proves the significance of directing students to take a coding course and that coding education must be a compulsory course in the curriculum.

Coding education is highly significant for North Cyprus and not only including it into the curriculum at the private schools but in the state schools as well will support the vision of being an informatics island [5,10].

In many countries all over the world (England, USA, Estonia, South Korea... etc.) coding education starts at pre-school. In Turkey, there is an ongoing process to include computational thinking as a compulsory course into the 5th and the 6th grades syllabi, and as an elective course into the 7th and the 8th-grade syllabi.

50,3% of the students stated that they used a tablet, 41,5% indicated that they used a smartphone for more than 2 hours a day. This finding reveals that this generation is growing up using mobile devices such as tablets, smartphones ... etc. Instead of ignoring this, we can come to the conclusion that these mediums as well as being used for entertainment purposes like playing games and watching videos, can be transformed into platforms where coding applications help develop information processing skills. People all over the world are aware of this situation and researchers are conducting studies in this field [15].

The FATİH Project, which is designed to provide each student in Turkey with a good education, access to high quality learning content and an equal opportunity of education, is the largest and most comprehensive educational movement to be implemented in the world in relation to the use of technology in education [6]. Using technology is an important project which will stop students being passive learners and will enable equal education opportunities by improving their effective communication, critical thinking, problem-solving, collaboration and cooperation skills.

Another finding is that students taking the coding course only used desktop PCs to access the code.org internet site mainly in the classroom environment. The reason behind this finding is that the private schools conducted coding education in this way. Kalelioğlu [9] in his study conducted with 32 primary school students, in which he aimed to analyse the effect of the code.org site on reflective thinking skills in problem-solving, revealed that there was a slight increase in the female students' reflective thinking skills for problem-solving, within the male students' problem-solving regarding reflective thinking skills and that students developed a positive attitude towards coding.

Furthermore, a private school is preparing to offer code.org additionally to WeDo 2.0 Logo Robotic education. 20 WeDo 2.0 education sets have been bought and the necessary teaching programmes have been launched.

Students reached the conclusion that generally the students were at an average coding self-efficacy level, and the students not taking a coding course were at a low coding self-efficacy level.

It was observed that there was not a statistically meaningful difference between the overall scores obtained from the coding self-efficacy scale according to gender with neither the students taking a coding course or between the students who were not taking a coding course.

While there was no difference between the overall scores obtained from the coding self-efficacy scale dependent on gender, the 11-year-old group of students have a higher sense of self-efficacy than the 12-year-old group students.

Similar findings were reported by Kukul, Gökçearslan and Günbatar [12] where they used a scale they had developed. This shows that similar solutions need to be brought to not only secondary schools but also to k12 schools [7].

This study is significant as it reveals that k12 students also want to receive coding education and that coding education has a significant impact on students' sense of self-efficacy. Within this context, it is suggested that similar studies are conducted not only in private schools but also in public schools and that steps are taken students with algorithmic thinking, computational thinking and problem-solving skills which they are expected to learn as 21st-century students.

References

[1]. Babori, A., Fihri Fassi, H., Hariri, A., & Bideq, M. (2016). An e-Learning environment for algorithmic: toward an active construction of skills. *World Journal of Educational Technology*, *8*(2), 82-90. Doi:http://dx.doi.org/10.18844/wjet.v8i2.819

[2]. Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing Mathematical Thinking with Scratch: an Experiment with 6th-grade students. Design for Teaching and Learning in a Networked World (pp. 17-27). Springer International Publishing.

[3]. Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe", European Schoolnet, [Doi: http://goo.gl/QIOkGw], Retrieved on: 11.11.2015, (2014).

[4]. Demirer & Sak (2016). Programming education and new approaches around the world and in Turkey. *Journal of Theory and Practice in Education. 12*(3), 521-546.

[5]. Duque, R., Bollen, L., Anjewierden, A., & Bravo, C. (2012). Automating the Analysis of Problem-solving Activities in Learning Environments: the Co-Lab Case Study. *J. UCS*, *18*(10), 1279-1307.

[6]. FATİH Project in Education, 2017. *About FATİH Project in Education* .http://fatihprojesi.meb.gov.tr/proje-hakkinda/ [accessed 18 September 2017].

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

[8]. Framework For 21st Century Learning 2017. http://www.p21.org/about-us/p21-framework [accessed 25 September 2017].

[9]. Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior*, *52*, 200-210.

[10]. Kanbul, S., & Uzunboylu, H. (2017). The Importance of Coding Education and Robotic Applications For Achieving 21st-Century Skills in North Cyprus. *International Journal Of Emerging Technologies In Learning (IJET), 12*(01), pp. 130-140. doi:http://dx.doi.org/10.3991/ijet.v12i01.6097

[11]. Karasar, N. (2017). Scientific Research Method: Concepts Principles Techniques. Nobel Academic Publishing, Ankara.

[12]. Kukul, V., Gökçearslan, Ş., & Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Eğitim Teknolojisi Kuram ve Uygulama*, *7*(1), 158-179.

[13]. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67.

[14]. Sayın, Z., & Seferoğlu, S., S. (2016). The Impact of Coding Education and Coding as a New 21st Century Skill on Education Policies. Academic Information 2016, 3-5 February 2016, Adnan Menderes University, Aydın.

[15]. Soykan, E., & Uzunboylu, H. (2015). New trends on mobile learning area: The review of published articles on mobile learning in science direct database. *World Journal of Educational Technology*, *7*(1), 31-41.

[16]. 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.