

Examination of Undergraduate and Associate Degree Students' Computer Programming Attitude and Self-Efficacy According to Thinking Style, Gender and Experience

Mustafa Serkan Gunbatar
Yuzuncu Yil University, Turkey

Submitted: 20.01.2018

Accepted: 20.09.2018

Published: 16.10.2018

Abstract

This study aimed to determine undergraduate and associate degree students' computer programming attitude and self-efficacy levels, and compare them according to thinking style, gender, department, weekly study time, and programming experience variables. The study employed the correlational research model. The researcher attempted to reach all associate and undergraduate students who had received the computer programming course at a state university. The computer programming self-efficacy scale, the computer programming attitude scale, and the holistic and analytic thinking in problem-solving scale were used to collect research data. Results suggested that the participants with different thinking styles showed significant differences regarding programming attitude and programming self-efficacy. Programming attitude and thinking style were significant predictors of programming self-efficacy. No difference was observed between genders in terms of the common effect and the partial effect of programming attitude and programming self-efficacy. However; differences were observed between participants from different departments and with different weekly study time. There was also a significant difference between the participants with different programming experience levels in terms of the common effect of programming attitude and self-efficacy, whereas no difference was found in terms of attitude alone.

Keywords: *Computer education; Programming self-efficacy; Programming attitude; Thinking style; University students*

Introduction

Access to information has become quite easy nowadays. However, it is still the individual's responsibility to turn information into knowledge and use it to solve problems. To come up with effective solutions for a given problem is about how the individual processes the information that is easily accessible. Critical, logical, reflective, metacognitive and creative thinking skills are collectively referred to as higher-order thinking skills (King, Goodson, & Rohani, 1998), and these skills play an important role in the individual's ability to solve problems. Thus, today's students should not only be evaluated based on their answers to questions asked, but also their knowledge level and ability to use 21st century skills (e.g. problem-solving, entrepreneurship, and creativity) (Soh, Arsad & Osman, 2010). After university education, graduates start their working life and are expected to come up with solutions to various problems on a daily basis. For this reason, graduates must place more importance on improving skills such as critical

thinking, creative thinking, communication skills, and entrepreneurship (Fong, Sidhu, & Fook, 2014).

Today, individuals tend to look for solutions to problems that they encounter by using ICT (Information and Communication Technology) tools, in other words, using their digital skills. 21st century skills may be classified as seven core skills (i.e. technical, information management, communication, collaboration, creativity, critical thinking, and problem solving) and five contextual skills (i.e. ethical awareness, cultural awareness, flexibility, self-direction, and lifelong learning) (van Laar, van Deursen, van Dijk, & de Haan, 2017). Also, 21st century learning skills include collaboration, communication, information literacy, media literacy, and ICT literacy (Pheeraphan, 2013). In addition to those skills, computational thinking is a frequently mentioned skill recently. Computational thinking skill is a 21st century skill that involves making use of computing and computers to solve complex problems. Individuals are expected to possess this skill in the digital age (Korkmaz, Cakir, & Ozden, 2017). It is believed that computational thinking will not be expected only from those work with computers, but it will be considered one of the basic skills (e.g. reading, writing, and arithmetic) that everyone must have (Korkmaz, Cakir, Ozden, Oluk, & Sarioglu, 2015). It can be described as “to possess knowledge, skills, and attitudes necessary to use computers to solve everyday problems” (Ozden, 2015). To put computational skill in a framework, it can be said that it is associated with creativity, algorithmic thinking, cooperativity, critical thinking, problem solving, and communication skills (Korkmaz, Cakir & Ozden, 2017; Korkmaz, Cakir, Ozden, Oluk, & Sarioglu, 2015).

Design-based learning activities such as computer programming training contribute to development of computational thinking skill which shows parallelisms with 21st century skills (Alsancak-Sarikaya, 2017; Barut, Tugtekin & Kuzu, 2016). Literature has shown that platforms such as SCRATCH and ALICE are useful for those who are new to programming in many aspects (i.e., computational thinking skill) (Sáez-López, Román-González, & Vázquez-Cano, 2016; Su, Yang, Hwang, Huang, & Tern, 2014). However, examination of current programming-related status of those who are required to have a programming basis for their profession according to different variables is a matter that needs to be studied constantly. Because they perceive that programming is relevant to their future career goals (Adamopoulos, 2017) and they will create computer programs with real programming editors in their professional lives. They should be examined in terms of other variables besides computational thinking. Programming self-efficacy, attitude, and thinking styles are important variables for programming performance.

Programming is known for its complexity and difficulty, and thus many programming students have difficulties with acquiring necessary programming competencies (Yukselturk & Altıok, 2017). Insomuch that, students with high, moderate, and low programming success tend to make similar mistakes (Rodrigo, Andallaza, Castro, Armenta, Dy, & Jadud, 2013) and students who are competent in other areas may be inadequate for success in programming (Byrne & Lyons, 2001). Because, computer programming requires abstract thinking, logical thinking, and program solving skills (Lin, 2016). Programming attitudes and self-efficacies are also important variables for programming performance (Yagci, 2016). Additionally, students primarily focus on “analytical thinking” and “how to turn the solution of the problem into code” when solving problems in computer programming (Hawi, 2014). Therefore, thinking style variable may also be a significant variable for programming.

One of the most significant determining variables distinguishing students with high and low programming performance is self-efficacy (Altun, & Mazman, 2015; Mazman, 2013). Self-efficacy is related to the individual’s belief in his or her capacity to make the necessary effort to

achieve a certain goal (Bandura, 1997). It is an important motivational construct (Lishinski, Yadav, Good & Enbody, 2016). It reflects the perceived capacity of the individual to accomplish a task, affects initial behaviors, and improves personal motivation to make extra effort to accomplish the task (Lin, Hung, & Lee, 2015). Thus, individuals who believe that they can perform the task successfully may make more effort than they normally would (Lin, 2016). It involves cognitive activities such as evaluation of the learning process (Al-Harthy, 2011), and is of great importance for success (Lishinski, Yadav, Good, & Enbody, 2016). It is recommended that self-efficacy is considered in studies on computer programming as an inclusive variable for other possible variables (Bergin & Reilly, 2005; Lin, 2016; Mazman & Altun, 2013; Yagci, 2016).

One of the problems encountered in programming teaching is negative attitudes of students (Korkmaz & Altun, 2013). However, studies mostly deal with cognitive constructs such as alternative comprehension, mental models, and novice-expert comparisons. Therefore, the number of studies on student attitudes has been limited (Cetin & Ozden, 2015). Attitude, which covers a considerable portion of affective learning, has significant effects on our behavior and is an important element that shapes our cognitive structure.

There are different findings in the literature regarding the effects of personal, and other factors on programming performance. Since programming is a skill that requires quite complex cognitive processes for each sub-task, it is necessary to find out individuals' cognitive skills required for programming, and reveal common effects of these skills by addressing them together with other factors (Mazman, 2013; Longi, 2016). A crucial dimension of cognitive skills is thinking style.

One of the theories that we need to consider when it comes to thinking style is Sternberg's Theory of Mental Self-government. In this theory, Sternberg (2017) explains thinking styles in terms of constructs from our notions of government. According to this theory, individuals can be understood in terms of the functions, forms, levels, scope, and leanings of government. People do not exhibit just one style or another, rather they do have preferences across various kinds of tasks and situations. There are two levels of mental self-government, namely; local and global. The local individual focuses on the nitty-gritty (the real issue). This individual may lose the forest for the trees, tends to break the task into parts, and focuses on the concrete specifics of the situation. The global individual likes to deal with big ideas, but sometimes can lose touch with the details. This individual may see the forest but lose track of the trees. People employing this style enjoy tasks that encourage them to think about major ideas and don't have to worry about details (Sternberg, 2017). Some researchers prefer to use holistic thinking style and analytic thinking style when describing thinking styles. Those concepts align with Sternberg's classification of local and global to a large extent (Umay & Ariol, 2011). If an individual visualizes the image as a whole, it is the holistic strategy. If s/he tends to break the image into smaller pieces, it is the analytic strategy. Independent of the content, both strategies are useful in problem solving. Some individuals are able to use both strategies alternately when they realize one is not suitable for the task (Hammouri, 2003). In this study, holistic and analytical thinking styles were taken into consideration as thinking styles.

It would be worthwhile to conduct studies on students' analytic thinking skills and computer programming (Sebetci & Aksu, 2014). Relations between holistic and analytic strategies with respect to problem solving in different disciplines may be studied in future research (Hammouri, 2003). As in other disciplines, learning approaches are strong determining factors of success in courses related to computer programming (de Raadt et al., 2005) and there is a relationship between study processes and thinking styles of students (Zhang, 2000). Gender is also an

important individual variable for research (Yang, Chen, & Hwang, 2015). Considering the gender variable, the most basic individual difference, female students are able to use analytic learning strategies significantly better than male students (Kesici, Sahin, & Akturk, 2009). Also, individuals with high cognitive development level are able to use a wider thinking range compared to those with low cognitive development level (Zhang, 2002). To conclude, thinking style and gender variables are important independent variables for this study.

Depending on the department variable, the personal traits of programming (e.g., self-efficacy, attitude) can be differentiated (de Raadt et al., 2005; Askar & Davenport, 2009; Korkmaz & Altun, 2013; Sebetci & Aksu, 2014; Yagci, 2016). Similarly, the effort (Cetin, 2016; Hawi, 2010; Rodrigo, Andallaza, Castro, Armenta, Dy, & Jadud, 2013) and past experience for programming (Askar & Davenport, 2009; Bergin & Reilly, 2005; Lin, 2016; Sharma, & Shen, 2018; Wilson & Shrock, 2001) can vary in terms of personal traits. Thus, within the scope of this study, department, weekly study hour (time), and programming experience variables appear to be the other important independent variables.

Purpose of the Study

This study aimed to examine Turkish undergraduate and associate degree students' computer programming attitude and self-efficacy according to some individual variables. To this end, the following questions were attempted to be answered:

RQ1: How are computer programming attitude levels and computer programming self-efficacy levels of the participants distributed with respect to thinking styles?

RQ2: Are programming attitude and thinking style variables significant predictors of programming self-efficacy?

RQ3: According to the common effect of computer programming attitude and computer programming self-efficacy:

- a. Is there a difference between the students based on the thinking style variable?
- b. Is there a difference between female and male students?
- c. Is there a difference between Computer Education and Instructional Technology (CEIT) department students and Computer Programming students?
- d. Is there a difference between the students based on weekly time spent studying programming languages?
- e. Is there a difference between the students based on experience in programming?

Methodology

Research Design

This study aimed at identifying computer programming attitude and self-efficacy levels of undergraduate and associate degree students, examining the relationship between computing self-efficacy and thinking style, and exploring whether computer programming attitude and self-efficacy differs depending upon variables of gender, academic department, weekly time spent for studying programming languages, level of experience in programming, and thinking style.

Because possible relationships between variables are revealed with the purpose of making predictions based on identified relationships between variables (Fraenkel & Wallen, 2009), this study employed the correlational research model.

Participants

The participants of the study were undergraduate and associate degree students in the province of Van, Turkey. The selection criteria were being enrolled in a computer-related department and having received the computer programming course. It was tried to reach all students in Yuzuncu Yil University who met the criteria. Therefore, the data were obtained from 2nd, 3rd, and 4th year undergraduate Computer Education and Instructional Technologies (CEIT) students and 1st and 2nd year associate degree Computer Programming (CP) students in accordance with the criteria. A total of 306 students were reached. One of the students was excluded from the study due to incompliance with multi variable normality assumption during data analysis. The study was completed with the data obtained from 305 students in total.

Table 1. Participants' Demographic Profiles

Department	Graduate level	Gender				Total	
		Female		Male		N	%
		N	%	N	%		
Computer Programming (Vocational School of Van)	freshmen	15	65.2	8	34.8	23	100.0
	sophomore	15	36.6	26	63.4	41	100.0
	Total	30	46.9	34	53.1	64	100.0
Computer Programming (Vocational School of Baskale)	freshmen	15	44.1	19	55.9	34	100.0
	sophomore	11	40.7	16	59.3	27	100.0
	Total	26	42.6	35	57.4	61	100.0
Computer Programming (Vocational School of Gevas)	freshmen	26	47.3	29	52.7	55	100.0
	sophomore	5	35.7	9	64.3	14	100.0
	3 rd year	1	100.0	0	0.0	1	100.0
	Total	32	45.7	38	54.3	70	100.0
Computer Programming (Vocational School of Ozalp)	sophomore	13	65.0	7	35.0	20	100.0
	Total	13	65.0	7	35.0	20	100.0
Computer Education and Instructional Technology (Faculty of Education)	sophomore	7	41.2	10	58.8	17	100.0
	junior	13	33.3	26	66.7	39	100.0
	senior	13	52.0	12	48.0	25	100.0
	5 th year	0	0.0	1	100.0	1	100.0
	Total	33	40.2	49	59.8	82	100.0
Total	junior	56	50.0	56	50.0	112	100.0
	sophomore	51	42.9	68	57.1	119	100.0
	junior	14	35.0	26	65.0	40	100.0
	senior	13	52.0	12	48.0	25	100.0
	5 th year	0	0.0	1	100.0	1	100.0
	Total	134	45.1	163	54.9	297	100.0

Detailed profiles of the participants according to variables of gender and graduate level can be seen in Table 1. 64 students were from Van Vocational School (VS), 61 were from Baskale VS, 70

were from Gevas VS, 20 were from Ozalp VS, and 82 were from CEIT Department of the Faculty of Education. Of the total, 134 (45%) participants were female, 163 (55%) were male, and 8 participants did not specify their gender.

Instrumentation

Computer Programming Self-Efficacy Scale

The Computer Programming Self-Efficacy Scale adapted to Turkish by Altun and Mazman (2012) was used to determine computer programming self-efficacy of the participants. The original scale was developed by Ramalingam and Wiedenbeck (1998). The Turkish form of the seven-point Likert scale consists of two factors and 9 items. For each item, the participant picks one of the seven options: "Absolutely confident (7)", "Mostly confident (6)", "Fairly confident (5)", "50/50 (4)", "Slightly confident (3)", "Mostly not confident (2)", "Not confident at all (1)". The first factor called "the ability to perform simple programming tasks" consists of 3 items, whereas the second factor called "the ability to perform complex programming tasks" consists of 6 items. Factor loading values of the first factor varies between 0.945 and 0.790. Factor loading values of the second factor varies between 0.886 and 0.786. The scale explains 80.814% of the total variance. The Cronbach's Alpha internal consistency coefficient is $\alpha = 0.907$ for the first factor and $\alpha = 0.943$ for the second factor. Goodness of fit index values are within the boundaries of excellent and acceptable fitness ($\chi^2/sd = 1.55$; RMSEA = 0.062; S-RMR = 0.034; NNFI=0.99; NFI=0.98; CFI=0.99; GFI =0.95; AGFI = 0.90; IFI = 0.99).

Computer Programming Attitude Scale

The Computer Programming Attitude Scale developed by Baser (2013) was administrated to determine attitudes of the students toward computer programming. The five-point Likert-type scale consists of four factors and 38 items. For each item, the participant picks one of the following options: "Strongly Agree (5)", "Agree (4)", "Neutral (3)", "Disagree (2)", "Strongly Disagree (1)". The first factor, confidence and motivation in learning programming, consists of 17 items. Factor loading values of the first factor varies between 0.801 and 0.495. The second factor, usefulness of programming, consists of 10 items. Factor loading values of the second factor varies between 0.813 and 0.567. The third factor, attitude toward success in programming, consists of 8 items. Factor loading values of the third factor varies between 0.865 and 0.587. The fourth factor, social perception of success in programming, consists of 3 items. Factor loading values of the fourth factor varies between 0.599 and 0.547. The scale explains 59.83% of the variance. The Cronbach's Alpha internal consistency coefficient is $\alpha = 0.944$ for the first factor, $\alpha = 0.920$ for the second factor, $\alpha = 0.926$ for the third factor, and $\alpha = 0.618$ for the fourth one.

The Holistic and Analytic Thinking in Problem-solving Scale (Thinking Styles Scale)

The Holistic and Analytic Thinking in Problem-solving Scale developed by Umay and Ariol (2011) was used to determine thinking styles of the participants. The scale consists of 5 items. Each item involves two situations, one that fits the analytical thinking style and one that fits the holistic thinking style for problem-solving. Those who cannot pick one of these two situations can choose the option "I have no idea". In each item, the participant is given 1 point for the

analytical option, 2 points for the “no idea” option, and 3 points for the holistic option. The closer the score to 5, the more dominant the analytical thinking style is, the closer the score to 15, the more dominant the holistic thinking style is. The Cronbach’s Alpha internal consistency coefficient of the scale is $\alpha = 0.780$.

Personal Information Form

Variables which significantly contribute to programming attitude and self-efficacy were determined by the researcher in light of the literature review. The important variables were gender, department, weekly time (hours) spent studying programming languages, programming experience, and thinking style. The personal information form was developed to cover those variables. The data collected using the personal information form constituted the independent variables of the study.

Data Collection Procedure and Analysis

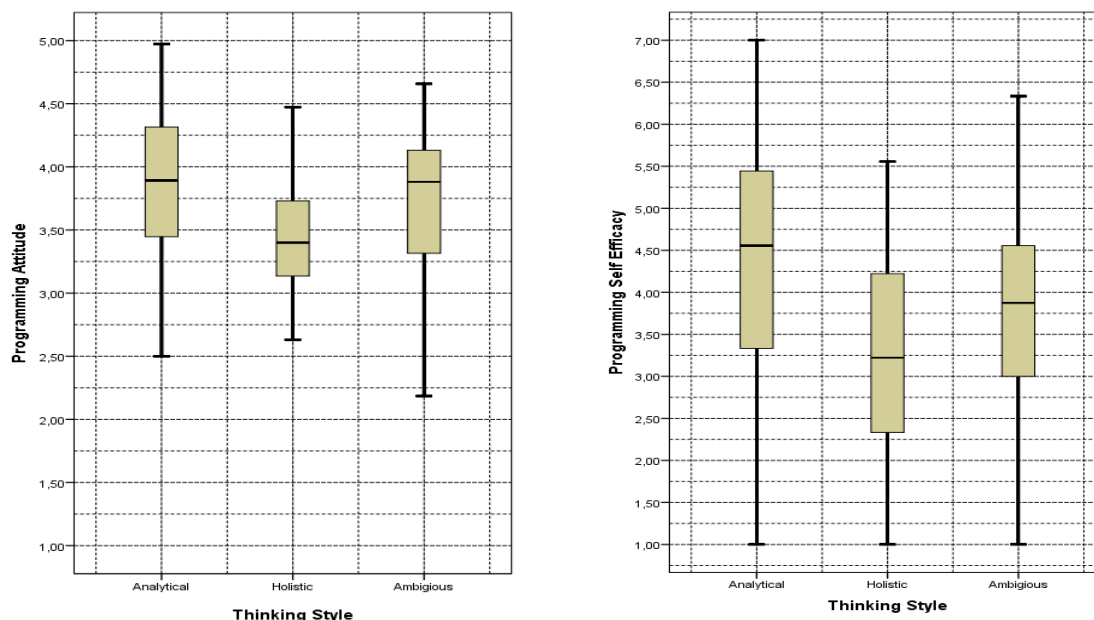
The data were collected using the instruments summarized above. The researcher administered the instruments himself after making necessary explanations in the spring semester of 2015-2016 academic year. The administration of the instruments took approximately 15-20 minutes.

Before analyzing the data, assumptions of the statistical techniques to be applied were tested. According to the results of the Kolmogorov-Smirnov test, which was performed for the single variable normality assumption, both the computer programming attitude and computer programming self-efficacy measurements showed normal distribution ($p > 0.05$). Mahalanobis distance coefficients were calculated for the multi variable normality assumption. The data with the ID number 166 was excluded since its coefficient was 15.278, while was supposed to be below 13.82 according to the chi-square table. For the linearity assumption, distribution graphs for each of the independent variables of programming attitude and programming self-efficacy measurements (gender, department, weekly time spent studying programming languages, experience in programming, and thinking style) were examined. The scattering was linear for all independent variables. Correlation levels between the variables were examined to test the data for multicollinearity and singularity. A moderate correlation was found between the dependent variables (i.e., programming attitude and programming self-efficacy) ($r = 0.376$, $p < 0.05$). Considering that multicollinearity may occur when the correlation value is above 0.8 and around 0.9 (Pallant, 2007), it was concluded that there was no multicollinearity in the dataset in question. The homogeneity of variance-covariance matrices was examined using the Box test. According to test results, the observed covariance matrices of the dependent variables are equal across groups ($p > 0.05$).

After concluding that the dataset met the necessary assumptions, the multivariable regression analysis and MANOVA were performed. Analysis of variance was performed to determine whether there was a difference based on programming attitude and programming self-efficacy measurements. The Bonferroni multiple comparison test was used to determine the source of the difference between independent variables with more than two categories.

Findings

RQ1: How are computer programming attitude levels and computer programming self-efficacy levels of the participants distributed with respect to thinking styles?



	Programming Attitude					Programming Self-Efficacy			
	N	Min	Max	\bar{X}	Ss	Min	Max	\bar{X}	Ss
Analytical	222	1.63	4.97	3.84	0.59	1.00	7.00	4.41	1.41
Holistic	42	2.63	4.47	3.46	0.42	1.00	5.56	3.25	1.10
Ambiguous	38	1.87	4.66	3.73	0.64	1.00	6.33	3.88	1.18
Total	302	1.63	4.97	3.77	0.59	1.00	7.00	4.16	1.41

Figure 1. Computer programming attitude levels and computer programming self-efficacy levels of the participants distributed with respect to thinking styles.

Figure 1 shows the descriptive data and the Box plot graphs where the participants are separately evaluated according to their thinking styles based on programming attitude and programming self-efficacy. Programming attitudes of the students with analytic thinking style showed more scattering compared to the students with holistic thinking style. In terms of self-efficacy, the students with analytic thinking style showed more scattering compared to both the students with holistic thinking style and the ambiguous students. The mean of programming attitude score of all students ($\bar{X}=3.77$) was at a high level, while the mean programming self-efficacy score of all students ($\bar{X}=4.16$) was at a moderate level.

RQ2: Are variables of programming attitude and thinking style significant predictors of programming self-efficacy?

The dependent variable is programming self-efficacy, while the independent variables are programming attitude and thinking style. Thinking style is on a classification scale.

Table 2. Results of Regression Analysis Regarding to Programming Self-Efficacy Prediction

Variable	B	Std Error	β	t	p	Zero-order r	Partial r
Constant	0.759	0.476		1.594	0.112	-	-
Programming Attitude	0.779	0.127	0.329	6.148	0.000	0.369	0.335
Thinking Style	0.654	0.170	0.206	3.846	0.000	0.270	0.217
R=0.420	R ² =0.177						
F _(2,299) =32.096	p=0.000						

Table 2 shows that programming attitude and thinking style were significant predictors of programming self-efficacy (R=0.420, R²= 0.177, F_(2,299)=32.096, p<0.05). It can be stated that programming attitude and thinking style explained approximately 18% of the total variance related to programming self-efficacy.

Programming self-efficacy increased as programming attitude increased. There was a positive correlation between programming self-efficacy and analytic thinking style, which was used as a dummy variable (r=0.27). Therefore, the participants with analytic thinking style had higher programming self-efficacy scores compared to the participants with holistic thinking style and ambiguous participants (i.e., those without a dominant style).

RQ3-a: According to the common effect of computer programming attitude and computer programming self-efficacy, is there a difference between the students based on the thinking style variable?

The data were divided into categories according to different thinking styles and MANOVA analysis was performed. As a result of the analysis, a significant difference was found between students with different thinking styles according to the common effect of dependent variables (F_(4,596) =8.293, p=0.000; Wilks' Lambda (Λ)= 0.897; Partial eta squared=0.053).

Table 3. Mean, Standard Deviation Values and ANOVA Results of Programming Attitude and Programming Self-Efficacy Measurements According to Thinking Style

	Thinking Style	N	\bar{X}	Ss	Sd	F	p	Partial Eta Squared
Programming Attitude	Analytic	222	3.84	0.59				
	Holistic	42	3.46	0.42	2-299	8.074	0.000*	0.051
	Ambiguous	38	3.73	0.64				
Programming Self-Efficacy	Analytic	222	4.41	1.41				
	Holistic	42	3.25	1.10	2-299	14.081	0.000*	0.086
	Ambiguous	38	3.88	1.18				

p<0.05

Table 3 shows the ANOVA result, where dependent variables are addressed separately with regard to thinking style. The participants with different thinking styles showed significant differences in terms of both programming attitude and programming self-efficacy (p<0.05). Bonferroni test was used to determine the source of the difference between the groups. According to the results of the Bonferroni test, there was a statistically significant difference between the participants with analytic thinking style and the participants with holistic thinking style in terms of programming attitude (p<0.05). Similarly, there was a statistically significant

difference between the participants with analytic thinking style and the participants with holistic thinking style in terms of programming self-efficacy ($p < 0.05$).

RQ3-b: According to the common effect of computer programming attitude and computer programming self-efficacy, is there a difference between female and male students?

According to the results of the MANOVA analysis, there was no significant difference between genders according to the common effect of dependent variables ($F_{(2,294)} = 1.245$, $p = 0.29$; Wilks' $\Lambda = 0.992$; Partial eta squared = 0.008).

Table 4. Mean, Standard Deviation Values and ANOVA Results of Programming Attitude and Programming Self-Efficacy Measurements According to Gender

	Gender	N	\bar{X}	Ss	Sd	F	p	Partial Eta Squared
Programming Attitude	Female	134	3.79	0.57	1-295	0.323	0.570	0.001
	Male	163	3.75	0.62				
Programming Self-Efficacy	Female	134	4.05	1.27	1-295	1.310	0.253	0.004
	Male	163	4.24	1.50				

* $p < 0.05$

Table 4 shows the ANOVA result, where dependent variables are separately addressed with respect to the gender variable. There was no significant difference between genders in terms of both programming attitude and programming self-efficacy ($p > 0.05$).

RQ3-c: According to the common effect of computer programming attitude and computer programming self-efficacy, is there a difference between CEIT and Computer Programming students?

As a result of the MANOVA analysis, a significant difference was found between the CEIT students and the computer programming students according to the common effect of dependent variables ($F_{(2,301)} = 11.678$, $p = 0.000$; Wilks' $\Lambda = 0.928$; Partial eta squared = 0.072).

Table 5. Mean, Standard Deviation Values and ANOVA Results of Programming Attitude and Programming Self-Efficacy Measurements According to Department

	Department	N	\bar{X}	Ss	Sd	F	p	Partial Eta Squared
Programming Attitude	CEIT	83	3.90	0.57	1-302	4.962	0.027*	0.016
	Comp Prog	221	3.73	0.60				
Programming Self-Efficacy	CEIT	83	4.78	1.27	1-302	23.093	0.000*	0.071
	Comp Prog	221	3.94	1.39				

* $p < 0.05$

Table 5 shows the ANOVA result, where dependent variables are separately addressed with respect to the department variable. There was significant differences between the CEIT students and the Computer Programming students for both programming attitude and programming self-efficacy ($p < 0.05$).

RQ3-d: According to the common effect of computer programming attitude and computer programming self-efficacy, is there a difference between the students based on the weekly study time variable?

As a result of the MANOVA analysis, a significant difference was found between students with different weekly study times according to the common effect of dependent variables ($F_{(14,576)} = 3.961$, $p=0.000$; Wilks' $\Lambda = 0.832$; Partial eta squared=0.088).

Table 6. Mean, Standard Deviation Values and ANOVA Results of Programming Attitude and Programming Self-Efficacy Measurements According to Weekly Study Hour

	Weekly Study Hour	N	\bar{x}	Ss	Sd	F	p	Partial Eta Squared			
Programming Attitude	Less than 1	97	3.54	0.55							
	1 hour	61	3.84	0.50							
	2 hour	52	3.87	0.53							
	3 hour	29	3.93	0.67	7-289	4.108	0.000*	0.090			
	4 hour	30	3.98	0.56							
	5 hour	8	3.86	0.67							
	6 hour	8	3.88	0.66							
	7 hour	12	4.06	0.82							
Less than 1	97	3.59	1.45								
1 hour	61	4.14	1.40								
2 hour	52	4.27	1.13								
Programming Self-Efficacy	3 hour	29	4.49	1.32	7-289	6.071	0.000*	0.128			
	4 hour	30	4.79	1.01							
	5 hour	8	5.17	1.39							
	6 hour	8	5.02	1.50							
	7 hour	12	5.24	1.13							

* $p < 0.05$

Table 6 shows the differences between students with different weekly study times in terms of programming attitude and programming self-efficacy. As shown in the table, the participants with different weekly study times showed significant differences in terms of both programming attitude and programming self-efficacy. According to the results of the Bonferroni multiple comparison test performed to find the source of the differences, the students with a weekly study time less than 1 hour had significant differences in terms of programming attitude compared to the students with a weekly study time of 1 hour, 2 hours, 3 hours, and 4 hours ($p < 0.05$). In terms of programming self-efficacy, the students with a weekly study time less than 1 hour had significant differences compared to the students with a weekly study time of 3 hour, 4 hours, 5 hours, and 7 hours ($p < 0.05$).

RQ3-e: According to the common effect of computer programming attitude and computer programming self-efficacy, is there a difference between the students based on the programming experience variable?

As a result of the MANOVA analysis, a significant difference was found between students with different programming experiences according to the common effect of dependent variables ($F_{(18,584)} = 2.486$, $p=0.001$; Wilks' $\Lambda = 0.863$; Partial eta squared=0.071).

Table 7. Mean, Standard Deviation Values and ANOVA Results of Programming Attitude and Programming Self-Efficacy Measurements According to Programming Experience

	How many years	N	\bar{x}	Ss	Sd	F	p	Partial Eta Squared
Programming Attitude	Less than 1	52	3.64	0.64				
	1 year	23	3.85	0.52				
	2 year	59	3.63	0.60				
	3 year	38	3.86	0.54				
	4 year	52	3.92	0.53	9-293	1.688	0.091	0.049
	5 year	37	3.78	0.65				
	6 year	21	3.70	0.61				
	7 year	14	4.07	0.64				
	8 year	4	3.81	0.23				
	9 year	3	3.60	0.47				
Less than 1	52	3.62	1.43					
1 year	23	4.04	1.18					
2 year	59	3.88	1.23					
3 year	38	4.53	1.30					
Programming Self-Efficacy	4 year	52	4.27	1.43	9-293	4.007	0.000*	0.110
	5 year	37	4.54	1.39				
	6 year	21	3.83	1.45				
	7 year	14	5.25	1.28				
	8 year	4	5.72	1.30				
	9 year	3	5.52	1.23				

*p<0.05

ANOVA was used to determine possible differences between the participants with different programming experience in terms of programming attitude and programming self-efficacy. The results are shown in Table 7. There was no significant difference between the participants with different programming experiences in terms of programming attitude. However, there was a significant difference in terms of programming self-efficacy (p<0.05). The Bonferroni test performed to determine the source of the difference revealed that there was a significant difference between the participants with 7 years of experience and the participants with less than 1 year and 2 years of experience in terms of programming self-efficacy (p<0.05).

Discussion

Results revealed that, when computer programming attitude levels of the participants were examined based on the thinking style variable, it was seen that the participants with analytic thinking style had a wider range of programming attitudes compared to the participants with holistic thinking style. Considering the mean scores, individuals with all three thinking styles (i.e., analytic, holistic, and ambiguous) were observed to have a high level of programming attitude. In terms of programming self-efficacy levels, the participants with analytic thinking style had a wider range of self-efficacy levels compared to both the participants with holistic thinking style and the ambiguous participants (i.e., those without a dominant style). Considering the mean scores, the analytic and ambiguous thinking styles had a moderate self-efficacy level, whereas those with holistic styles were found to have low self-efficacy level. While these findings are

consistent with the literature with regard to the attitude variable, there are differences with regard to the self-efficacy variable. Korkmaz and Altun (2013) found that undergraduate students had sufficiently positive level of computer programming attitude, while Ozyurt (2015) reported that those who received distance vocational school education had usually positive computer programming attitudes and high level of self-efficacy. It is recommended that new studies should be conducted on programming self-efficacy. It is believed that such studies will contribute to the literature with respect to the self-efficacy variable.

Programming attitude and thinking style are significant predictors of programming self-efficacy. It can be stated that programming self-efficacy increases as programming attitude improves. Also, those with analytic thinking style had higher self-efficacy compared to those with holistic thinking style.

Programming self-efficacy is one of the variables that affect the programming success (Bergin & Reilly, 2005; Mazman, 2013; Lishinski, Yadav, Good, & Enbody, 2016). Although attitude toward computer programming has not been a comprehensively studied subject (Cetin & Ozden, 2015), Ozyurt (2015) reported a moderate, positive, and significant correlation between programming attitude and self-efficacy. The fact that attitude and self-efficacy increased and decreased together in this study as well indicates the relationship between them.

The relationship between perceived self-efficacy and performance is reciprocal (Lishinski, Yadav, Good & Enbody, 2016). Self-efficacy regarding programming is one of the variables that improves student success in the introduction to programming course (Wilson & Shrock, 2001). However, individuals with similar programming self-efficacies may show different levels of learning insistence (Lin, 2016). The situation is a little different in terms of thinking style. Programming success has a high and significant correlation with analytic thinking, and a moderate and significant correlation with logical thinking skill. Compared to holistic thinking skill, the effect of analytic thinking style on programming success is more significant (Sebetci & Aksu, 2014). The results about the thinking style on the self-efficacy variable is consistent with the studies cited. Programming attitude and thinking style were shown to predict programming self-efficacy in this study. In future studies, different variables may take into account such as computational thinking that is likely to predict self-efficacy.

Regarding the common effect of attitude and self-efficacy, there were significant differences between analytic, holistic, and ambiguous (i.e., without a dominant thinking style) participants. When examined separately in terms of the attitude variable and the self-efficacy variable, there were differences between participants with different thinking styles. The students with analytic thinking style had significantly higher attitude and self-efficacy compared to those with holistic thinking style. The ambiguous students, on the other hand, had no difference with other thinking styles in term of both attitude and self-efficacy. Shaw (2012) carried out an online programming course with forum support and found that the learning style variable affected learning performance. Considering the direct effect of perceived programming self-efficacy on programming success (Mazman, 2013), findings obtained in relation to the self-efficacy variable in our study and findings obtained by Shaw in the online environment where the interaction took place are consistent. The number of studies which compare programming self-efficacy and attitude with respect to the thinking style variable is limited in the literature. According to the findings obtained in our study, the thinking style variable is significant for self-efficacy and attitude.

According to this study, analytical thinking style is an important element for programming self-efficacy and programming attitude. In the future studies, it is recommended that the thinking style should be considered as an independent variable and comparisons should be made in terms of variables that can be important for computer programming (e.g., programming attitude, self-efficacy, computational thinking).

Regarding the gender, there was no significant difference between female and male students according to the common effect of attitude and self-efficacy variables. In other words, high or low scores in both of these variables did not produce different results for males and females. Similar results were found when attitude and self-efficacy were examined separately. There was no significant difference between genders in terms of mean attitude scores and mean self-efficacy scores.

It seems that the gender variable is one of the most frequently studied variables in the literature when it comes to programming attitude and self-efficacy. While programming attitude is not a predictor of programming performance on its own, there is a significant and positive correlation between programming attitude and programming success (Baser, 2013). It is reported that gender has a positive correlation with programming success (Owolabi, Olanipekun, & Iwerima, 2014). That is a variable predicting programming performance (Lau & Yuen, 2011), and increases success in the introduction to programming course (Wilson & Shrock, 2001). Findings in the literature show that beginning male students find programming easier than females, have more enthusiasm for programming later on, and have higher learning outcomes than females (Rubio, Romero-Zaliz, Mañoso, & Angel, 2015). On the other hand, it is stated that female students may internalize the initial feedback and lose their enthusiasm for programming due to negative effects of these feedbacks on their self-efficacy (Lishinski, Yadav, Good & Enbody, 2016). Similarly, various studies show that male students have better attitude toward programming compared to female students (Baser, 2013; Korkmaz & Altun, 2013; Ozyurt, 2015) and self-efficacy that may be considered as an indicator for attitude (Korkmaz & Altun, 2013), consistently differ in terms of the gender variable (Lin, 2016) in favor of male students (Askar, & Davenport, 2009; Ozyurt, 2015). However, some studies suggest that female students present similar characteristics with male students in relation to programming (Byrne & Lyons, 2001; Malik & Coldwell-Neilson, 2018; Sharma & Shen, 2018; Ventura, 2005), gender does not contribute to prediction of programming performance significantly (Mazman, 2013), and programming attitude does not vary depending on gender (Erol & Kurt, 2017; Yagci, 2016). There are conflicting results in terms of gender variable in the literature (Adamopoulos, 2017; Tsai, Wang & Hsu, 2018). Thus, findings obtained in this study conflict with studies in the literature which suggests that gender affects programming attitude and self-efficacy. One of the most basic personal differences, the gender variable should be considered in future studies. There is no consensus on gender variable in the studies related to programming attitude and self-efficacy. It is recommended that these variables should be considered again in new studies.

Regarding the department, there was a significant difference between departments according to the common effect of attitude and self-efficacy. The CEIT students had higher levels compared to computer programming students in terms of the common effect of attitude and self-efficacy. When examined separately in terms of the attitude variable and the self-efficacy variable, a similar result was found. The CEIT students had higher scores compared to computer programming students in both attitude and self-efficacy.

Erol and Kurt (2017) revealed the reasons behind positive and negative computer programming attitudes in their study and found CEIT students to have positive programming attitude in

general. The possible reason behind the difference between students from different departments in terms of programming attitudes might be the difference in importance placed to programming (Korkmaz & Altun, 2013). However, Yagci (2016) reported no difference between CEIT students, who receive 14 credits of computer programming until graduation, and computer programming students, who receive 31 credits of computer programming until graduation, in terms of programming attitude and self-efficacy. The results of the current study conflict with the study mentioned. It is recommended that associate and undergraduate students enrolled in different departments other than CEIT should be compared in future studies. Also, possible reasons behind differences, if any, should be examined.

Regarding the weekly study time variable, there were differences between participants with different weekly study times according to the common effect of attitude and self-efficacy. A significant difference was found when the participants were examined separately in terms of attitude and self-efficacy as well. The students with a weekly study time less than 1 hour had significantly lower attitude scores than the students with a weekly study time of 1 hour, 2 hours, 3 hours, and 4 hours. The students with a weekly study time less than 1 hour had significantly lower self-efficacy scores than the students with a weekly study time of 3 hour, 4 hours, 5 hours, and 7 hours.

Variables such as lack of studying, lack of effort, and lack of practice have the potential influence on computer programming (Hawi, 2010). The primary predictor variable for programming success is effort (Ventura, 2005). Findings obtained in our study show that the students with a weekly programming study time less than 1 hour had lower attitude, self-efficacy, and combined effect compared to those with a weekly programming study time more than 1 hour. Thus, it can be stated that a study time less than one hour is insufficient in terms of studying, effort, and practice. In this study the attitude and self-efficacy of individuals who showed inadequate individual learning efforts were significantly lower than those who tried harder. Hence, it is recommended to carry out new research which can give cause and effect relation between weekly study time (i.e., individual learning effort) and those variables (i.e., attitude and self-efficacy).

Finally, regarding the experience, there were significant differences between students with different programming experiences according to the common effect of attitude and self-efficacy. When examined separately in terms of attitude and self-efficacy, there were significant differences between students with different experience levels in terms of self-efficacy, while no difference was found in terms of attitude. In other words, while experience did not cause a difference in terms of attitude, a significant difference occurred when the self-efficacy variable was introduced. The participants with 7 years of programming experience had significantly higher self-efficacy scores compared to those with less than 1 year and 2 years of experience.

Prior knowledge influences programming performance (Veerasamy, D'Souza, Lindén, & Laakso, 2018). There is a relationship between programming experience and programming performance (Adamopoulos, 2017; Bergin & Reilly, 2005; Kock, Moqbel, Jung & Syn, 2018). Increased programming experience leads to improved programming performance (Byrne & Lyons, 2001). Additionally, experience enhances success in individuals who receive the programming course for the first time (Wilson & Shrock, 2001). One of the two most important elements of success in computer science, programming experience may be predictive of success, albeit not on its own, but through the common effect of different variables (Wilson & Shrock, 2001). Previous experiences may be predictive of learning self-efficacy as well (Lin, 2016). Therefore, it can be said that programming self-efficacy perception is related with previous experiences and

preliminary knowledge (Mazman & Altun, 2013). The programming self-efficacy of those who have more programming experience is also significantly higher (Tsai, Wang & Hsu, 2018). It is quite reasonable that there was a difference between the students with a high level of experience in programming (i.e., 7 years) and the students who had met with programming only a short time ago (less than 1 year and 2 years) in this study. The fact that attitude did not change with the programming experience variable, but a significant difference occurred when the self-efficacy variable was added to the picture is an important finding. Based on this finding, it can be said that considering self-efficacy and attitude variables together when it comes to programming experience may produce healthier results.

According to the programming experience variable, participants' self-efficacy differs whereas attitude do not differ. In the literature, studies have been mostly carried out with participants who are new in programming. Therefore, in the future studies, it is recommended that attitudes of individuals with more programming experience should be examined.

References

- Adamopoulos, F. (2017). *An influence model of the experience of learning programming*. (Unpublished doctoral dissertation). RMIT, Melbourne, Australia.
- Alsancak-Sarikaya, D. (2017, May). The effect of teaching programming on computational thinking skills. *11th International Computer & Instructional Technologies Symposium*, Malatya, Turkey, 24-26 May 2017.
- Altun, A. & Mazman, S. G. (2012). Programlamaya ilişkin öz yeterlilik algisi ölçeğinin Türkçe formunun geçerlilik ve güvenilirlik çalışması. *Journal of Measurement and Evaluation in Education and Psychology*, 3(2), 297-308.
- Altun, A. & Mazman, S. G. (2015). Identifying latent patterns in undergraduate Students' programming profiles. *Smart Learning Environments*, 2(1), 1.
- Askar, P. & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java programming among engineering students. *The Turkish Online Journal of Educational Technology*, 8(1), 26-32.
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: W. H. Freeman.
- Barut, E., Tugtekin, U. & Kuzu, A. (2016, October). Programlama eğitiminin bilgi işlemsel düşünme becerileri bağlamında incelenmesi. *4th International Instructional Technologies & Teacher Education Symposium*. Elazığ, Turkey, 6-8 October 2016.
- Baser, M. (2013). Developing attitude scale toward computer programming. *The Journal of Academic Social Science Studies*, 6(6), 199-215.
- Bergin, S. & Reilly, R. (2005, February). Programming: factors that influence success. In *ACM SIGCSE Bulletin*, 37(1), 411-415.
- Byrne, P. & Lyons, G. (2001, June). The effect of student attributes on success in programming. *ACM SIGCSE Bulletin*, 33(3), 49-52.
- Cetin, I. & Ozden, M. Y. (2015). Development of computer programming attitude scale for university students. *Computer Applications in Engineering Education*, 23(5), 667-672.

- de Raadt, M., Hamilton, M., Lister, R., Tutty, J., Baker, B., Box, I., ... & Petre, M. (2005). Approaches to learning in computer programming students and their effect on success. In *Proceedings of the 28th HERDSA Annual Conference: Higher education in a changing world (HERDSA 2005)* (pp. 407-414). Higher Education Research and Development Society of Australasia.
- Erol, O. & Kurt, A. A. (2017). Investigation of CEIT students' attitudes towards programming. *Mehmet Akif Ersoy University Journal of Faculty of Education*, 1(41), 314-325.
- Fong, L. L., Sidhu, G. K., & Fook, C. Y. (2014). Exploring 21st century skills among postgraduates in Malaysia. *Procedia-Social and Behavioral Sciences*, 123, 130-138.
- Freankel, J.R. & Wallen, N. E. (2009). *How to design and evaluate research in education* (7th ed.). New York: McGraw-Hill.
- Hammouri, H. A. (2003). An investigation of undergraduates' transformational problem solving strategies: Cognitive/metacognitive processes as predictors of holistic/analytic strategies. *Assessment & Evaluation in Higher Education*, 28(6), 571-586.
- Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education*, 54(4), 1127-1136.
- Hawi, N. (2014). Learning programming: a model emerging from data. *International Journal of Computer Applications*, 100(4), 24-34.
- Kesici, S., Sahin, I., & Akturk, A. O. (2009). Analysis of cognitive learning strategies and computer attitudes, according to college students' gender and locus of control. *Computers in Human Behavior*, 25(2), 529-534.
- King, F. J., Goodson, L., & Rohani, F. (1998). Higher order thinking skills: Definition, teaching strategies, assessment. *Publication of the Educational Services Program (now known as the Center for Advancement of Learning and Assessment)*. Retrieved on 20 January 2018 from www.cala.fsu.edu.
- Kock, N., Moqbel, M., Jung, Y., & Syn, T. (2018). Do older programmers perform as well as young ones? Exploring the intermediate effects of stress and programming experience. *Cognition, Technology & Work*, 20, 489-504.
- Korkmaz, O. & Altun, H. (2013). Engineering and CEIT students' attitude towards learning computer programming. *The Journal of Academic Social Science Studies*, 6(2), 1169-1185.
- Korkmaz, O., Cakir, R., Ozden, M. Y., Oluk, A., & Sarioglu, S. (2015). Investigation of Individuals' Computational Thinking Skills in terms of Different Variables. *Ondokuz Mayıs University Journal of Faculty of Education*, 34(2), 68-87.
- Korkmaz, O., Cakir, R., & Ozden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558-569.
- Lau, W. W. & Yuen, A. H. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, 57(1), 1202-1213.
- Lin, G. Y. (2016). Self-efficacy beliefs and their sources in undergraduate computing disciplines: An examination of gender and persistence. *Journal of Educational Computing Research*, 53(4), 540-561.

- Lin, S., Hung, T. C., & Lee, C. T. (2015). Revalidate forms of presence in training effectiveness mediating effect of self-efficacy. *Journal of Educational Computing Research*, 53(1), 32-54.
- Lishinski, A., Yadav, A., Good, J., & Enbody, R. (2016, August). Learning to program: Gender differences and interactive effects of students' motivation, goals, and self-efficacy on performance. *Paper presented at the 12th Annual International ACM Conference on International Computing Education Research (ICER'16)*. Tacoma, WA. August 18-20, 2016.
- Longi, K. (2016). *Exploring factors that affect performance on introductory programming courses* (Unpublished master's thesis). University of Helsinki, Finland.
- Malik, S. I. & Coldwell-Neilson, J. (2018). Gender differences in an introductory programming course: New teaching approach, students' learning outcomes, and perceptions. *Education and Information Technologies*, 23(5), 2453-2475.
- Mazman, S. G. & Altun, A. (2013). The effect of introductory to programming course on programming self-efficacy of CEIT students. *Journal of Instructional Technologies & Teacher Education*, 2(3). 24-29.
- Mazman, S. G. (2013). *Modeling The Influence Of Cognitive Based Individual Differences On Programming Performance* (Unpublished doctoral dissertation). Hacettepe University, Ankara, Turkey.
- Owolabi, J., Olanipekun, P., & Iwerima, J. (2014). Mathematics ability and anxiety, computer and programming anxieties, age and gender as determinants of achievement in basic programming. *GSTF Journal on Computing (JoC)*, 3(4), 109.
- Ozden, M. Y. (2015). *Computational thinking*. Retrieved on 03 May 2017 from <http://myozden.blogspot.com.tr/2015/06/computational-thinking-bilgisayarca.html>.
- Ozyurt, O. (2015). An analysis on distance education computer programming students' attitudes regarding programming and their self-efficacy for programming. *Turkish Online Journal of Distance Education*, 16(2), 111-121.
- Pallant, J. F. (2007). *SPSS survival manual: A step-by-step guide to data analysis with SPSS* (3rd edition). New York: Open University Press.
- Pheeraphan, N. (2013). Enhancement of the 21st century skills for Thai higher education by integration of ICT in classroom. *Procedia-Social and Behavioral Sciences*, 103, 365-373.
- Ramalingam V. & Wiedenbeck, S. (1998). Development and validation of scores on computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, 19(4) 365-379.
- Rodrigo, M. M. T., Andallaza, T. C. S., Castro, F. E. V. G., Armenta, M. L. V., Dy, T. T., & Jadud, M. C. (2013). An analysis of java programming behaviors, affect, perceptions, and syntax errors among low-achieving, average, and high-achieving novice programmers. *Journal of Educational Computing Research*, 49(3), 293-325.
- Rubio, M. A., Romero-Zaliz, R., Mañoso, C., & Angel, P. (2015). Closing the gender gap in an introductory programming course. *Computers & Education*, 82, 409-420.
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools. *Computers & Education*, 97, 129-141.

- Sebetci, O., & Aksu, G. (2014). The effect of logical and analytical thinking skills on computer programming languages. *Journal of Educational Sciences & Practices, 13*(25), 65-83.
- Sharma, R. & Shen, H. (2018). Does education culture influence factors in learning programming: A comparative study between two universities across continents. *International Journal of Learning, Teaching and Educational Research, 17*(2), 1-24.
- Shaw, R. S. (2012). A study of the relationships among learning styles, participation types, and performance in programming language learning supported by online forums. *Computers & Education, 58*(1), 111-120.
- Soh, T. M. T., Arsad, N. M., & Osman, K. (2010). The relationship of 21st century skills on students' attitude and perception towards physics. *Procedia-Social and Behavioral Sciences, 7*, 546-554.
- Sternberg, R.J. (2017). *Theory of mental self-government: Thinking styles*. Retrieved on 01 June 2017 from <http://www.robertjsternberg.com/thinking-styles>
- Su, A. Y. S., Yang, S. J. H., Hwang, W. Y., Huang, C. S. J., & Tern, M. Y. (2014). Investigating the role of computer-supported annotation in problem-solving-based teaching: An empirical study of a Scratch programming pedagogy. *British Journal of Educational Technology, 45*(4), 647-665.
- Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2018). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research, 56*(6), 1-16.
- Umay, A. & Ariol, S. (2011). A comparison of problem solving skills in terms of holistic and analytical thinking styles. *Pamukkale University Journal of Faculty of Education, 30*, 27-37.
- van Laar, E., van Deursen, A. J., van Dijk, J. A., & de Haan, J. (2017). The relation between 21st-century skills and digital skills: A systematic literature review. *Computers in Human Behavior, 72*, 577-588.
- Veerasamy, A. K., D'Souza, D., Lindén, R., & Laakso, M. J. (2018). The impact of prior programming knowledge on lecture attendance and final exam. *Journal of Educational Computing Research, 56*(2), 226-253.
- Ventura Jr, P. R. (2005). Identifying predictors of success for an objects-first CS1. *Computer Science Education, 15*(3), 223-243.
- Wilson, B. C. & Shrock, S. (2001, February). Contributing to success in an introductory computer science course: a study of twelve factors. *ACM SIGCSE Bulletin, 33*(1), 184-188).
- Yagci, M. (2016). Effect of attitudes of information technologies (IT) preservice teachers and computer programming (CP) students toward programming on their perception regarding their self-sufficiency for programming. *Journal of Human Sciences, 13*(1), 1418-1432.
- Yang, T. C., Chen, S. Y., & Hwang, G. J. (2015). The influences of a two-tier test strategy on student learning: A lag sequential analysis approach. *Computers & Education, 82*, 366-377.

- Yukselturk, E. & Altioek, 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
- Zhang, L. F. (2000). Relationship between thinking styles inventory and study process questionnaire. *Personality and Individual Differences*, 29(5), 841-856.
- Zhang, L. F. (2002). Thinking styles and cognitive development. *The Journal of Genetic Psychology*, 163(2), 179-195.

Correspondence: Mustafa Serkan Gunbatar, Assistant Professor, Department of Computer Education and Instructional Technologies, Faculty of Education, Yuzuncu Yil University, Van, Turkey.
