



International Journal of Educational Methodology

Volume 6, Issue 2, 259 - 270.

ISSN: 2469-9632
<http://www.ijem.com/>

Examination of the Computer Programming Self-Efficacy's Prediction towards the Computational Thinking Skills of the Gifted and Talented Students

Yunus Emre Avcu*

Sehit Prof. Dr. Ilhan Varank Art and Science Center, TURKEY

Leyla Ayverdi

Sehit Prof. Dr. Ilhan Varank Art and Science Center, TURKEY

Received: February 14, 2020 • Revised: April 19, 2020 • Accepted: April 27, 2020

Abstract: The study's goal was to examine the correlation between the computer programming self-efficacy and computational thinking skills of gifted and talented students. The capacity of the computer programming self-efficacy of gifted and talented students to predict their computational thinking skills were also examined. The relational screening model has been implemented in the research. The participants of the study were composed of 106 secondary school gifted and talented students studying the Individual Ability Recognition Program (IAR) at the Science and Art Center in the city center. Typical case sampling was applied for the student identification of the participants, 46 are female and 60 are male. Gifted and talented students' computational thinking skills were assessed using the "Computational Thinking Skills Scale" and the computer programming self-efficacy was measured by using the "Computer Programming Self-Efficacy Scale". Data were analysed by Pearson correlation analysis and simple linear regression analysis in statistical software SPSS 22. Research results found that there was a positive and medium correlation between the computer programming self-efficacy and computational thinking skills. The gifted and talented students' computer programming self-efficacy demonstrated 31.5% of the total variance in computational thinking skills. This finding supports the claim which is present in the literature that self-efficacy in computer programming is the affective aspect of computational thinking skills. To predict computational thinking skills, it may be recommended to build multiple models for affective skills of gifted and talented students.

Keywords: *Gifted and talented students, computational thinking, computer programming self-efficacy, simple linear regression analysis.*

To cite this article: Avcu, Y. E., & Ayverdi, L. (2020). Examination of the computer programming self-efficacy's prediction towards the computational thinking skills of the gifted and talented students. *International Journal of Educational Methodology*, 6(2), 259-270. <https://doi.org/10.12973/ijem.6.2.259>

Introduction

Among digital natives, there is a particularly gifted and talented group that has been in interaction with the technological world from the moment they were born and can use technology easily and quickly in the processes of acquiring and generating information (Koroglu, 2015). Clark (2015) identifies individuals as gifted and talented who have created a performance in the talent fields of general mental ability, special academic ability, leadership, creative and productive thinking, visual and performing arts, or are capable of processing and creating performance by using one or more of those talent fields. It can be stated that the educational interventions to be applied to gifted and talented students who need special education because of their differences and special needs should be carried out considering all these characteristics of individuals. It is an obligation rather than a responsibility to provide educational services to the gifted and talented students they need and to improve their talents (Citel & Ozdemir-Kilic, 2019; Sak, 2014).

In the context of digital economy strategies, developed countries concentrate on enhancing the digital skills of their citizens, increasing their welfare standards and economic development by considering the world's digital transition (Gulbahar, 2018). Therefore relations have begun to be formed between the countries' development plans and the education policies of gifted and talented students (Avcu & Er, 2017; Citil, 2018; Demirhan et al., 2017; Ministry of National Education, 2013; Sak et al., 2015). By developing technology, gifted and talented students are expected to contribute to their countries' digital economy strategies. It is thought that the contribution of gifted and talented students to their countries in the field of technology will be realized by discovering and developing special talents in

* **Corresponding author:**

Yunus Emre Avcu, Sehit Prof. Dr. Ilhan Varank Art and Science Center, Karesi, Balikesir, Turkey. ✉ yunus1099@hotmail.com



this field. Computer science education which is provided to gifted and talented students, and recognized as a field of talent, has great importance in this sense (Roman-Gonzalez et al., 2018a; Siegel, 2004). Computer science education and programming education which stands at the core of this process, have been provided to gifted and talented digital natives (Ongoz & Aksoy, 2018).

Programming education is usually regarded as a process of vocational education but it contributes to a discipline of teaching which involves pedagogical priorities and computational thinking skills (Kert, 2018). Through programming teaching processes, learners can build computational thinking skills (Burke, 2012; Lye & Koh, 2014; Lockwood & Mooney, 2017; Hsu et al., 2018; Waite, 2017). Computational thinking is one of the skills that form the future professions, which is considered as essential for professional life and predicted to be developed by learners of the 21st century (Freeman et al., 2017; International Society for Technology in Education, 2016; Phoenix Research Institution, 2011). Gifted and talented students use the requisite thinking patterns or techniques to address real-life problems using cognitive tools, thanks to computational thinking. They also expand the limits of thinking through the use of computer science principles and concepts. Therefore, they have the opportunity to be better prepared for the technological world and the corporate life, and thus, they can be lifelong learners without the impact of constantly changing practices and technologies (Gulbahar et al., 2019). What computational thinking is and what are the factors influencing this mechanism should be explored based on this significance (Ibili et al., 2020; Korkmaz et al., 2015; Roman-Gonzalez et al., 2018b).

Computational thinking is "the process of thinking which involves formulating problems and solutions linked to the issues to deliver the solutions in a way which can be successfully performed by an information processing unit (human, computer, robot, etc.)" (Wing, 2011, p.1). Formulating is not only utilized as a method for mathematics or physics, but also for perceiving the elements of a problem or its solution by understanding the associations between them and revealing certain components within a framework (Tekinarşlan & Cetin, 2018). Formulation involves the use of cognitive processes such as decomposing problems, abstraction and generalization, and solutions created in the process of computational thinking should be able to be efficiently and accurately implemented in a limited time using system resources, in other words, it should be effective (Angeli et al., 2016; Cetin & Toluk-Ucar, 2018; Csizmadia et al., 2015; Grover & Pea, 2013; Selby & Woollard, 2013). Garcia-Valcarcel et al. (2019) use the fundamental concepts of programming and computer science to describe computational thinking as the capacity and ability to solve problems. In the light of the aforementioned definitions, computational thinking has been defined by considering problem solving, formulating and understanding the problem, cognitive concepts and actions.

Apart from these definitions, computational thinking is expressed by International Society for Technology in Education (ISTE, 2016) as an umbrella skill, as a combination of creativity, algorithmic thinking, critical thinking, cooperation, and problem-solving. Determining what computational thinking does not say how it should be handled practically in schools. It is thought that operational definitions should be developed to acquire computational thinking skills in especially information technologies and software courses at the K-12 level. Brennan and Resnick (2012) have developed their operational definitions of computational thinking according to their own experiences in teaching programming based on Scratch blocks. Consequently, computational thinking has three fundamental elements: computational concepts, computational practices, and computational perspectives. Concepts used in programming are computational concepts (sequence, loops, events, parallelism, conditionals, operators, data: variable, array, and function). Computational practices (experimenting and iterating, testing and debugging, reusing and remixing, abstracting and modularizing) are linked to the practices students follow when creating something through programming. Computational perspectives (expressing, connecting and questioning) present the awareness that learners have developed about themselves, their relationships with others and the technological environment (Brennan & Resnick, 2012).

To measure the change in students' computational thinking skills, the computational concepts (Avcu & Er, 2020; Constantinou & Ioannou, 2018; Faber et al., 2017; Fronza et al., 2016; Rodriguez, 2017) and computational practices (Atmatzidou & Demetriadis, 2016; Avcu & Er, 2020; Garcia-Valcarcel et al., 2019; Repenning et al., 2015; Sullivan et al., 2017; Weese & Feldhausen, 2017) have tremendous importance. Student attitudes and tendencies are just as critical as computational concepts and practices for acquiring computational thinking (Kong, 2019). Throughout the programming process, the tendencies and attitudes of individuals support computational thinking (Gulbahar et al., 2019). Computational perspectives represent the affective side of computational thinking according to Brennan and Resnick (2012). Barr and Stephenson (2011) also discuss attitudes and tendencies which constitute the computational thinking's affective dimension. Those are the learner's ability to;

- Have confidence in self to achieve complicated tasks,
- Show tolerance to the uncertainties,
- Be persistent and patient in solving difficult and open-ended problems,
- Reach a solution via working with the group,
- Be aware of personal strengths and weaknesses while working with others.

Computational thinking's affective aspect is not only about the things that Brennan and Resnick (2012) and Barr and Stephenson (2011) are said. The motivating beliefs of the students also have an impact on computational thinking (Kong, 2019). Computer programming-self-efficacy, which is one of the learners' motivational beliefs, has a great effect on programming teaching processes based on computational-thinking skill improvement. The programming self-efficacy represents the computational perspectives of students (Roman-Gonzalez et al., 2019) and the understanding and evaluation of individuals in terms of their ability to solve computational problems by utilizing their programming expertise and skills (Kong, 2017).

Determining the degree of self-efficacy in the computer programming of the students enables individuals to reflect on their programming performances (Anastasiadou & Karakos, 2011; Askar & Davenport, 2009; Cigdem, 2015; Davidson et al., 2010; Kanaparan et al., 2017; Ramalingam et al., 1998). Low self-efficacy in computer programming negatively impacts the success of students in the teaching of programming processes (Hongwarittorn & Krairit, 2010). Computer programming self-efficacy is a vital element to succeed in programming activities, and students with a high degree of computer programming self-efficacy are more likely to use their knowledge and their skills to solve computational problems (Kong, 2017). From such a standpoint, the purpose of this research is to explore the relationship between computer programming self-efficacy and computational thinking skills, and the prediction of computational thinking skills through computer programming self-efficacy.

In the literature, screening studies are examining the level of gifted and talented students' computational thinking skills and its differentiation status according to gender (Donmez et al., 2018; Kirit et al., 2018, Uzumcu & Ucar, 2018), the self-efficacy perception for computational thinking skills and its differentiation status according to demographic variables (gender, grade level, school type, previous programming experience, etc.) (Cakir & Mutlu-Bayraktar, 2019). There is also an experimental design study researching the impact of programming teaching on gifted and talented students' computational thinking skills and computer programming self-efficacy (Avcu & Er, 2020). In summary, it can be said that in the sample of gifted and talented students, there are very few studies that examine the differentiation status of the computational thinking skills according to independent variables and the effect of programming education on it.

The relations between computer programming self-efficacy and computational thinking skills of students with normal abilities have also been examined through literature (Ciftci et al., 2017; Yildiz-Durak et al., 2019). Nonetheless, there is no research found in the literature that explores the relationship between the gifted and talented students' computer programming self-efficacy and computational thinking skills, and the predictive status of computer programming self-efficacy, and their computational thinking abilities. In this regard, the findings of this research are thought to shed some light on future studies to improve the programming success of gifted and talented students.

Methodology

This section includes details about the method used, the research model, the participants, the data collection procedure, the data collection tools and how the data were collected and interpreted to obtain the information needed to address the sub-problems of the research.

Problem

The questions relating to this study are as follows:

1. Is there a relationship between computer programming self-efficacy and the computational thinking skills of gifted and talented students?
2. To what extent do the gifted and talented students' computer programming self-efficacy predict their computational thinking skills?

Research Model

In the study, a relational screening model, which is a subtype of the general survey model, was applied in order to evaluate the relationship between gifted and talented students' computational thinking skills and computer programming self-efficacy and to demonstrate to what extent computer programming self-efficacy predicted their computational thinking skills. Relational screening is a form of research to define the relationship between two or more variables and to gain information on causality (Buyukozturk et al., 2014; Karasar, 2014).

Participants

Participants in this research were 106 gifted and talented secondary school students who study at the Balikesir Province's Sehit Prof.Dr. Ilhan Varank Science and Art Center, who are also a part of the Individual Ability Recognition Program (IAR). Balikesir is a province located in the west part of Turkey with a dense population. To identify the study group, typical case sampling from purposeful sampling methods has been applied. Purposeful sampling methods enable an in-depth analysis of cases that are assumed to include valuable details (Buyukozturk et al., 2014). For the typical

case sampling, an average, a typical sample is estimated among all the cases in the world, and information from this sample is gathered (Buyukozturk et al., 2014). This study is conducted in the Sehit Prof. Dr. Ilhan Varank Science and Art Center which is a typical sample among the 139 BILSEMs that are present in the 81 cities of Turkey. Of those students, 46 are female and 60 are male. 44 of the students are in the 5th grade, 25 are in the 6th grade, 19 are in the 7th grade and 18 are in the 8th grade. 85 of them were diagnosed within the general mental field, 5 in the visual arts field and 16 in more than one field. Furthermore, 100 students received programming education, and this education was with block-based programming.

Data Collection Procedure

The researchers implemented both scales to the students, and that process took almost a lesson hour. Each student was provided with a Personal Information Form and those two Scale Forms. While collecting information, students were informed about the purpose, the significance of the research and the measuring tools framework. Students were reminded that participating in the research was voluntary and that the results of this study could also only be used for scientific purposes, and their personal information will be protected and secured.

Data Collection Tools

Personal information form, computational thinking scale, and computer programming self-efficacy scale have been used as data collection tools in the spring term of the academic year 2018-2019. The researchers prepared the personal information form to learn about students' gender, grade level, BILSEM diagnostic field and status of programming education received.

Computational Thinking Scale (CTS): For the validity and reliability studies of the scale, the scale was applied to 726 university students firstly, and then to the 582 university students. As a result of the exploratory factor analysis of the scale consisting of 29 items, it is found out that is consisting of five factors. Confirmatory factor analysis was applied to confirm factor structures. The factors, item numbers and internal consistency coefficients included in the scale are as follows: creativity dimension (8 items, Cronbach's α value .843), algorithmic thinking dimension (6 items, Cronbach's α value .869), collaboration dimension (4 items, Cronbach's α value .865), critical thinking dimension (5 items, Cronbach's α value .784), problem-solving dimension (6 items, Cronbach's α value .727). The overall Cronbach's α value of the scale which is consisting of 29 items was calculated as .822. The draft scale form was applied to 241 secondary school students and the scale factor structure, structure validity, reliability of scale scores, and classification of items were defined. The software of AMOS and SPSS were used to test validity and reliability. Confirmatory factor analysis used for the examination of the validity of the unique factor structure of the scale for secondary school students, and it was reported that the requirements for model-data compliance had been fulfilled for secondary school level. Reliability was calculated as an internal consistency value of .809 for Cronbach's α . Seven items have been excluded from the original scale according to the findings and it has been decided that the final scale consisting of 22 items can be used to assess the level of computational thinking skills of secondary school students in Turkey (Korkmaz et al., 2015). The Cronbach's Alpha (α) coefficient calculated to determine the reliability of CTS was .857. Tests with a reliability coefficient of .70 and above: are considered to be sufficient for the reliability of the test scores (Buyukozturk, 2014).

Computer Programming Self-Efficacy Scale (CPSS): "Computer Programming Self-Efficacy Scale (CPSS)" was developed by Kukul et al. (2017) to measure the computer programming self-efficacy levels of secondary school students. A total of 233 secondary school students were examined for the study of validity and reliability, including 45 from 5th grade, 138 from 6th grade and 50 from 7th grade, who were enrolled in the 2014-2015 academic year at a state school in Ankara. Two items from the scale which consisted of 33 items were excluded as a result of an exploratory factor analysis. The final scale form is composed of 31 items. The percentage of the scale of variance explanation having a single factor structure is 41.15. The value corresponds to an acceptable rate for research in social sciences. After the exploratory factor analysis was conducted, confirmatory factor analysis assessed the appropriateness of the model for the accepting structure received, and it was recognized that certain values accepted were not appropriate to the results of the study. $\chi^2 / df = 1.47$; RMSEA value .045; NFI value .96; NNFI value .99; RMR value .061; CFI value .99; IFI value .99; GFI value .85 and AGFI value .83. The Cronbach's Alpha reliability coefficient was estimated as .95 as a result of reliability analysis. Besides, by applying the split-half method, the Spearman-Brown reliability coefficient was determined as $r = .96$. It was found, as a result of the tests, that the scale which was recognized to have a single factor structure, may be used to assess the computer programming self-efficacy of secondary school students (Kukul et al., 2017). The Cronbach's Alpha (α) coefficient calculated to determine the reliability of CPSS was .954 and it is considered to be sufficient for the reliability of the test scores (Buyukozturk, 2014).

When two or more different scales applied together, there was the potential for common variance bias problem. To examine Common Method Variance (CMV) in this study, A Harman's one-factor analysis has been carried out in statistical software SPSS 22. In Harman's one factor approach all items measuring constructs are loaded into one factor and run the Exploratory factor analysis (EFA). If total variance for one factor is more than 50% then it poses a threat of CMV (Tehseen et al., 2017). EFA with unrotated principal components analysis was run where all the variables were

entered and constrained the number of factors to one. The single factor revealed 31.8%. This shows that CMV might not be a serious problem for this study.

Data Analysis

To demonstrate to what extent the computer programming self-efficacy predictor (independent) variable predicts the variable of computational thinking (dependent), a simple linear regression analysis is decided upon. To carry out this study, firstly checked was if the conditions required have been satisfied. To give accurate results in a simple linear regression analysis, the following conditions must be satisfied (Can, 2014):

- A normal distribution should be represented by the predictor (independent) and predicted (dependent) variables, which are constantly variable at the least interval scale.
- There should be a linear relationship between the predictor (independent) and predicted (dependent) variables.

The skewness and kurtosis coefficients were examined and the Kolmogorov-Smirnov test was performed to verify if the computer programming self-efficacy and computational thinking variables showed a normal distribution. The skewness value of the scores obtained from the computer programming self-efficacy scale is -0.784 , and the value of kurtosis is $.650$. For the values obtained from the computational thinking scale, the skewness value was measured as -0.277 , and the kurtosis value as -0.087 . It is an indicator of normality (Garson, 2012; Tabachnick & Fidell, 2001) if the skewness and kurtosis values listed within the limits of -1 and $+1$. Following Kolmogorov-Smirnov tests on the scores of programming self-efficacy ($p = .115$) and computational thinking ($p = .062$), it was found that the results related to data showed a normal distribution. When the p (Sig.) value is insignificant, in other words above 0.05 , the data is normally distributed (Secer, 2015).

After the data were found to be normally distributed, it was examined whether there were outliers in the data set. A valid approach to detecting outliers is the outlier labeling rule, which is based on multiplying the Interquartile Range (IQR) by a factor of 1.5 (Hoaglin et al., 1986). The outlier labeling rule is more valid when using 2.2 as a multiplier, rather than 1.5 according to Hoaglin and Iglewicz (1987). The application of the rule was briefly provided as follows ($Q1$ = first quartile, $Q3$ = third quartile, IQR = Interquartile range): 1) Low outliers is calculated with the formula as $Low\ outliers = Q1 - 2.2 \cdot IQR$ ($Q3 - Q1$) and 2) High outliers is calculated with the formula as $High\ outliers = Q1 + 2.2 \cdot IQR$. Observation values less than low outliers and greater than high outliers are defined as possible outliers. For the data obtained from CTS and CPSS, it was determined that there are no outliers by using the outlier labeling rule.

A Scatter plot searched for the existence of a linear relationship between the data. The scatter plot is shown in Figure 1. When examining Figure 1 it can be seen that there is a linear relationship between the results about computer programming self-efficacy and computational thinking scales.

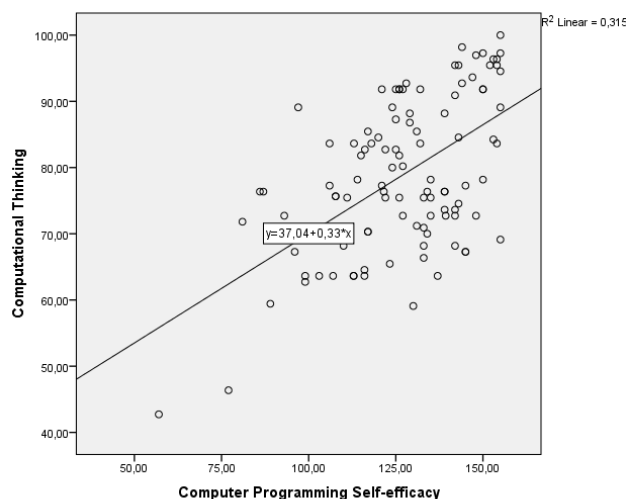


Figure 1. Scatter Plot

Pearson correlation analysis was conducted in order to assess the correlation between the computer programming self-efficacy and the scores of computational thinking. Simple linear regression analysis was performed to assess to what extent the computer programming self-efficacy of the gifted and talented students predicted their computational thinking skills. The "Enter" method was utilized when analyzing the regression. The degree of significance used in hypothesis testing was accepted in the study as $.01$ in the estimation of correlation and as $.05$ in the analysis of simple linear regression. The analysis was performed using statistical software of SPSS 22. The correlation coefficient (r) is defined according to the following expressions when evaluating the correlations between variables, "a low correlation

between .30 and .00, a medium correlation between .70 and .30, high correlation between 1.00 and .70" (Buyukozturk, 2014, p.32).

Findings

Analyzes were performed and summarized in the tables in this section to show the association between variables and to what extent the independent variable predicted the dependent variable. Table 1 presents the results of the Pearson correlation study conducted to assess the correlation between computer programming self-efficacy and computational thinking scores.

Table 1. Findings of the Correlation Analysis between Variables of Computer Programming Self-Efficacy and Computational Thinking Skills

	Computer Programming Self -Efficacy	Computational Thinking
Computer Programming Self -Efficacy	1	.561**
Computational Thinking	.561**	1

**p<.01, n=106

When analyzing Table 1 it demonstrates the simple linear correlation method conducted to decide whether there is a significant relationship between computer programming self-efficacy and computational thinking skills of gifted and talented students, it reveals a positive, medium and significant correlation between computer programming self-efficacy and computational thinking skills (r = .561, p< .01)

Simple linear regression analysis was performed to assess the what extent to the gifted and talented students' computer programming self-efficacy predicted their computational thinking skills, and the results relevant to the study are summarized in Table 2.

Table 2. Findings of Simple Regression Analysis That Determine to What Extent the Computer Programming Self-Efficacy Predict the Computational Thinking Skills

Criterion Variable: Computational Thinking Skills						
Prediction Variable	R	ΔR2	B	Beta (β)	t	p
Computer Programming Self -Efficacy	.561	.315	.330	.561	6.91	.00

* p<.05

While the results relevant to the simple linear regression analysis presented in Table 2 are reviewed, it has been found that the gifted and talented students' computer programming self-efficacy is a significant predictor of computational thinking skills (F₍₁₋₁₀₄₎= 47.76, p< .05). Computer programming self-efficacy explains 31.5% of computational thinking skills. Furthermore, the significance test of the predictive variable coefficient (B = .330) based on the regression equation indicates that computer programming self-efficacy is a significant predictor (p < .05).

The regression equation which predicts the computational thinking skills is as follows, as per the outcome of the regression analysis: Computational thinking skills = (0.330 X Computer programming self-efficacy)+37.03.

The findings of the study indicated that gifted and talented students' computer programming self-efficacy is positively correlated with their computational thinking skills. In other words, the more students have computer programming self-efficacy, the more they indicate computational thinking skills. Programming self-efficacy significantly predicts computational thinking skills of gifted and talented students. Namely, programming self-efficacy is a strong variable that predict the computational thinking skills of gifted and talented students.

Discussion

In this research, the correlation between computer programming self-efficacy and computational thinking skills of gifted and talented students and the prediction status of computer programming self-efficacy concerning computing thinking skills were examined. Research findings illustrated that the computer programming self-efficacy and computational thinking skills of gifted and talented students have a significant, positive, and medium relationship in between. Furthermore, computer programming self-efficacy is a significant predictor of computational thinking skills and explains 31.5 % of computational thinking skills. The regression equation which predicts computational thinking skills is "computational thinking skills = (0.330 X Computer programming self-efficacy) + 37.03". So, the average rate of change in the computational thinking skills caused by a unit change results in a change of programming self-efficacy as .330.

Research in the literature has shown that there is a significant, positive and medium correlation between computer programming self-efficacy and computational thinking skills of students with normal abilities (Ciftci et al., 2018; Yildiz-Durak et al., 2019). Unsal-Serim (2019) has also stated a positive and high correlation between students' computational thinking skills and self-efficacy perceptions regarding block-based programming. The results of previous studies focused on students with normal abilities and the results of the current study support each other. However, Ciftci et al. (2018) addressed computer programming self-efficacy as a dependent (predicted) variable in the multiple regression model they have developed and studied the contribution of independent (predictive) variables to the model, including computational thinking, reflective thought on problem-solving, and computer-related developments. They determined that the most important contribution to the formation of the model was the computational thinking, then reflective thinking related to problem-solving and following computer-related developments. This result attained by Ciftci et al. (2018) revealed that computational thinking skills would explain programming self-efficacy. The findings obtained in our research indicates that 31.5 % of computational thinking skills were explained by programming self-efficacy. The conclusion drawn in the current research coincides with the literature's views that programming self-efficacy is the affective aspect of computational thinking skills and is a reflection of computational perspectives (Kong, 2017; Kong, 2019; Roman-Gonzalez et al., 2018b; Roman-Gonzalez et al., 2019). In another research with students of normal ability level, it has been reported a positive and medium-level relationship between self-efficacy perception relative to computational thinking performance and computational thinking skill (Roman-Gonzalez et al., 2018). In the very same study, a positive and weak correlation was identified between general ICT self-efficacy and the general use of information and communication technologies in computational thinking. Even though Roman-Gonzalez et al. (2018b)'s work does not concentrate on the relationship between computer programming self-efficacy and computational thinking skills, it corresponds with the findings derived in the current study in terms of showing the relations between variables linked to computer programming self-efficacy and computational thinking skills.

Several literature studies demonstrate the role of affective factors in computational thinking education and computational thinking skills of the students with normal abilities. Yildiz-Durak and Saritepeci (2018) found the attitude towards mathematics significantly predicted the computational thinking skills. Roman-Gonzalez et al. (2016) found significant relationships between computational thinking and the factors of the Big Five Questionnaire-Children version (BFQ-C) which are openness, conscientiousness, and extraversion. Roman-Gonzalez et al. (2018b) claimed that while cognitive factors explain 27% of computational thinking skills, non-cognitive factors explain 24% of computational thinking skills. Kong et al. (2018) illustrate the relationships between the learner's programming empowerment, interest in programming, and attitude towards programming collaboration in computational thinking education through the development of a structural equation model. Results from studies conducted in Kong et al. (2018) indicate that students with a high interest in programming consider programming more meaningfully, had a greater impact, have greater creative self-efficacy and programming self-efficacy.

When the studies with gifted and talented students were put under the scope, Kirmit et al. (2018) investigated gifted and talented students' computational thinking skills in terms of girls and boys and found that there were significant differences in favor of male students in creative thinking, algorithmic thinking, and critical thinking sub-factors. Donmez et al. (2018), on the other hand, found that the computational thinking skills of the gifted and talented students do not differ by gender. In another study conducted with gifted and talented students (Uzumcu & Ucar, 2018), it was revealed that students received high scores in general computational thinking and decomposition questions, and they received lower scores in abstraction, algorithm, pattern-model, and evaluation-debugging questions. Cakir and Bayraktar (2019) revealed that programming education had an effect on gifted and talented students' attitudes towards the information technology course and their self-efficacy perceptions on computational thinking skills. Avcu and Er (2020) found that as a result of 74 hours of programming training, gifted and talented students improved their computational thinking skills, but there was no change in their computer programming self-efficacy.

Studies with gifted and talented students are mostly to reveal the average scores of students in the sub-dimensions of computational thinking skills and to examine this skill in terms of gender. It was also determined that there is a correlation between computational thinking and programming training. However, in our study, unlike these studies, the correlation between computational thinking and computer programming-self-efficacy and the predictive level of computer programming self-efficacy in computational thinking of gifted and talented students was examined. Our study differs from other studies in this aspect and offers a different perspective on the literature. Research indicates that gifted and talented students begin to lose their self-confidence while they are in primary school and this case can proceed until university (Merriman, 2012). This can lead gifted and talented students to grow up skeptical of their intellectual abilities, consider themselves as less talented than they actually are, and feel they have to work harder while others become successful by relying on their innate talents (Edins, 2010). Depending on this, it should not be overlooked that computer programming self-efficacy has a major impact on early age programming education for gaining computational thinking skills in young gifted and talented students.

Conclusion

The conclusion about the above researches and the conclusion we obtained in our study are mutually supportive in disclosing the affective aspects of computational thinking. The outcomes of the current study will guide future research for the extensive collaborative opportunities to examine the affective factors associated with computational thinking and programming, as well as the impact of the programming curriculum. Computer programming self-efficacy is highlighted as an important affective variable in this study that predicts the computational thinking abilities of gifted and talented secondary school students. The conclusion drawn in this research corroborates the presence of a non-cognitive side of computational thinking. Thus, it is presumed that priority should be given to computer programming self-efficacy to become thorough and successful in educational interventions for gifted and talented students aimed at enhancing computational thinking.

Suggestions

The suggestions that were established in light of the findings of the research are:

- To improve gifted and talented students' computational thinking skills, it could be proposed that resources enhancing computer programming self-efficacy should be put into action in programming teaching processes for gifted and talented students.
- Intermediary models can be established to influence the computational thinking skills from gifted and talented students' social/emotional characteristics (empathy and high levels of awareness of the expectations and feelings of others, emotional intensity, high expectations of others, etc.), intuitive characteristics (creative approaches and inventiveness, curiosity, ability to predict, interest in the future, etc.) and characteristics that could be an obstacle for success (extensive daydreaming, failure to complete work, perfectionism, a challenge to assignments that seem pointless to the student, etc.) through computer programming self-efficacy.
- By conducting longitudinal researches, it is possible to investigate whether the strengthening of computer programming self-efficacy of gifted and talented students actually improves their computational thinking skills.
- Multiple regression models can be developed by taking into consideration the cognitive and affective factors that predict gifted and talented students' computational thinking skills.
- Computer programming self-efficacy and computational thinking skills can be recommended as part of courses offered in education programs at different grades regarding the programming teaching processes for gifted and talented students.

Limitations

Working with gifted and talented secondary school students in only an institution that offers education to gifted and talented students and evaluation of computational thinking skills using only a measuring tool of the Likert type can be seen as limitations. Since the computational thinking skills need to be assessed by using several different evaluation methods together. In addition, it can be counted as a limitation that the generalizability of the research results is valid in the sample of gifted and talented students.

References

- Anastasiadou, S. D., & Karakos, A. S. (2011). The beliefs of electrical and computer engineering students regarding computer programming. *The International Journal of Technology, Knowledge and Society*, 7(1), 37-51.
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, 19(3), 47-57.
- Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java programming. *The Turkish Online Journal of Educational Technology – TOJET January*, 8(1), 1-7.
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: a study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75(2), 661-670.
- Avcu, Y. E., & Er, K. O. (2017). Education of gifted and talented individuals in Germany, the Netherlands, Sweden and Turkey: An investigation of education policies and its implementations. *International Online Journal of Educational Sciences*, 9(4), 1154-1170. <https://doi.org/10.15345/iojes.2017.04.018>
- Avcu, Y.E., & Er, K.O. (2020). Developing an instructional design for the field of ICT and software for gifted and talented students. *International Journal of Educational Methodology*, 6(1), 161-183. <https://doi.org/10.12973/ijem.6.1.161>
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *Acm Inroads*, 2(1), 48-54.

- Brennan, K., & Resnick, M. (2012, April 13-17). *Using artifact-based interviews to study the development of computational thinking in interactive media design* [Paper presentation]. Annual American Educational Research Association Meeting, Vancouver, BC, Canada.
- Burke, Q. (2012). The Markings of a New Pencil: Introducing programming-as-writing in the middle school classroom. *The National Association for Media Literacy Education's Journal of Media Literacy Education*, 4(2), 121-135.
- Buyukozturk, S. (2014). *Sosyal bilimler icin veri analizi el kitabi: Istatistik, arastirma deseni, SPSS uygulamalari ve yorum* [Data analysis handbook for social sciences: Statistics, research design, SPSS applications and interpretation]. Pegem Akademi.
- Buyukozturk, S., Kilic-Cakmak, E., Akgun, O. E., Karadeniz, S., & Demirel, F. (2014). *Bilimsel arastirma yontemleri* [Scientific research methods]. Pegem Akademi.
- Cakir, C., & Mutlu-Bayraktar, D. (2019). Programlama egitiminin ustun yetenekli ogrencilerin bilgi islemsel dusunme becerisine ve bilisim teknolojilerine yonelik oz yeterlik algilarina etkisi [The effect of programming training on gifted students' thinking skills and self-efficacy perceptions of information technologies]. In S. Emir (Ed.), *VI. National Gifted Education Congress* (pp. 63-77). Istanbul University.
- Can, A. (2014). *SPSS ile bilimsel arastirma surecinde nicel veri analizi* [Quantitative data analysis in scientific research process with SPSS]. Pegem Akademi.
- Cetin, I., & Toluk-Ucar Z. (2018). Bilgi islemsel dusunme tanimi ve kapsamı [Definition and scope of computational thinking]. In Y. Gulbahar (Ed.), *Bilgi islemsel dusunmeden programlamaya* [From computational thinking to programming] (pp.41-74). Pegem Akademi.
- Ciftci, S., Cengel, M., & Paf, M. (2018). Reflective thinking skills on computational thinking and problem solving as a predictor of self-efficacy of informatics teacher candidates on programming. *Ahi Evran University Journal of the Faculty of Education*, 19(1), 321-334. <https://doi.org/10.29299/kefad.2018.19.009>
- Cigdem, H. (2015). How does self-regulation affect computer-programming achievement in a blended context? *Contemporary Educational Technology*, 6(1), 19-37.
- Citil, M. (2018). Turkiye'de ustun yeteneklilerin egitimi politikalarinin degerlendirilmesi [Evaluation of gifted students' education policy in Turkey]. *Journal of National Education/Milli Egitim Dergisi*, 47(Special Issue 1), 143-172.
- Citil, M., & Ozdemir-Kilic, M. (2019). Ozel yetenekliler alanina giris. In O. Kilic & M. Citil (Eds.). *Ozel yetenekli ogrencim var* [I have a gifted and talented student] (pp.98-119). Gokce Ofset.
- Cetin, I., & Toluk-Ucar Z. (2018). Bilgi islemsel dusunme tanimi ve kapsamı [Definition and scope of computational thinking]. In Y. Gulbahar (Ed.), *Bilgi islemsel dusunmeden programlamaya* [From computational thinking to programming] (pp.41-74). Pegem Akademi.
- Clark, B. (2015). Ustun zekali ogrencileri anlamak [Understanding the gifted students]. In F. Kaya & U. Ogurlu (Eds.), *Ustun zekali olarak buyumek* [Growing Up Gifted: Developing the Potential of Children at School and at Home] (pp.1-34). Nobel Akademik.
- Constantinou, V., & Ioannou, A. (2018, September 3-6). *Development of computational thinking skills through educational robotics* [Paper presentation]. 13th European Conference On Technology Enhanced Learning, Leeds, UK.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking a guide for teachers*. Computing At School. <https://www.computingschool.org.uk/computationalthinking>
- Davidson, K., Larzon, L., & Ljunggren, K. (2010). *Self-Efficacy in Programming among STS Students*. Uppsala University. <https://www.it.uu.se/edu/course/homepage/datadidaktik/ht10/reports/Self-Efficacy.pdf>
- Demirhan, E., Elmali, S., & Besoluk, S. (2018). How to raise awareness in society on the importance of gifted students? *Journal of Multidisciplinary Studies in Education*, 1(1), 1-6.
- Donmez, I., Kirit, S., Gurbuz, S., & Birsen, O. (2018, May 4-6). *Ustun yetenekli ogrencilerin bilgisayarca dusunme becerilerinin incelenmesi* [Investigation of computational thinking skills of gifted students] [Paper presentation]. V. Gifted and Education Congress, Gaziantep, Turkey.
- Edins, C. A. (2010). *Self-efficacy and self-esteem in gifted and non-gifted students in the elementary school system* (Unpublished doctoral dissertation). University of Capella.
- Faber, H. H., Wierdsma, M. D., Doornbos, R. P., van der Ven, J. S., & de Vette, K. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network*, 12(5), 13-24.

- Freeman, A., Adams Becker, S., Cummins, M., Davis, A., & Hall Giesinger, C. (2017). The Horizon Report. K-12 Edition Examines Emerging Technologies for Their Potential Impact On and Use in Teaching, Learning, And Creative Inquiry in Schools. Retrieved from Eric database.
- Fronza, I., Corral, L., & El Ioini, N. (2016, May 4-6). Computational thinking through mobile programming [Paper presentation]. 16th Annual Conference on Information Technology Education, Chicago, IL, USA.
- Garcia-Valcarcel Munoz-Repiso, A., & Caballero-Gonzalez, Y. A. (2019). Robotics to develop computational thinking in early childhood education. *Media Education Research Journal*, 59(17), 63-72.
- Garson, G.D. (2012). *Testing statistical assumptions*. Statistical Associates Publishing.
- Grover, S., & Pea, R. (2013, March 6-9). *Using a discourse- intensive pedagogy and android's app inventor for introducing computational concepts to middle school students* [Paper presentation]. SIGCSE'13, Colorado, USA.
- Gulbahar, Y. (2018). Bilgi islemsel dusunme ve programlama konusunda degisim ve donusumler [Changes and transformations in computational thinking and programming]. In Y. Gulbahar (Ed.), *Bilgi islemsel dusunmeden programlamaya* [From computational thinking to programming] (pp.395-411). Pegem Akademi.
- Gulbahar, Y. Kert, S. B., & Kalelioglu F. (2019). The self-efficacy perception scale for computational thinking skill: Validity and reliability study. *Turkish Journal of Computer and Mathematics Education*, 10(1), 1-29.
- Hoaglin, D. C., Iglewicz, B., & Tukey, J. W. (1986). Performance of some resistant rules for outlier labeling. *Journal of the American Statistical Association*, 81(396), 991-999.
- Hoaglin, D. C., & Iglewicz, B. (1987). Fine-tuning some resistant rules for outlier labeling. *Journal of the American Statistical Association*, 82(400), 1147-1149.
- Hongwarittorn, N., & Krairit, D. (2010, April 6-9). *Effects of program visualization on students' performance and attitudes towards java programming* [Paper presentation]. The International conference on Computing, Communication and Control Technologies, Orlando, Florida.
- Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126(11), 296-310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Ibili, E., Gunbatar, M. S., & Sirakaya, M. (2020). Bilgi-islemsel dusunme becerilerinin incelenmesi: meslek liseleri orneklemi [An examination of the computational thinking skills: Sample of vocational high schools]. *Kastamonu Education Journal/Kastamonu Egitim Dergisi*, 28(2), 1067-1078.
- International Society for Technology in Education (2016). *ISTE standarts for students*. ISTE. <https://www.iste.org/standards/for-students>
- Kanaparan, G., Cullen, R., & Mason, D. (2017, December 4-6). *Effect of self-efficacy and emotional engagement on introductory programming students* [Paper presentation]. Australasian Conference on Information Systems, Hobart, Australia.
- Karasar, N. (2014). *Bilimsel arastirma yontemi (kavramlar ilkeler teknikler)* [Scientific research method (concepts, principles, techniques)] (27th ed.). Nobel Akademik.
- Kert, S. B. (2018). Programlama ogretimi icin pedagojik yaklasimlar [Pedagogical approaches for programming teaching]. In Y. Gulbahar, & H. Karal (Eds.), *Kuramdan uygulamaya programlama ogretimi* [Theory-to-practice programming teaching] (pp.93-130). Pegem Akademi.
- Kirmit, S., Donmez, I., & Cataltas, H. E. (2018). Ustun yetenekli ogrencilerin bilgisayarca dusunme becerilerinin incelenmesi [The study of gifted students' computational thinking skills]. *Journal of STEAM Education*, 1(2), 17-26.
- Kong, S. C. (2017). Development and validation of a programming self-efficacy scale for senior primary school learners. In S. C. Kong, J. Sheldon, & K. Y. Li (Eds.), *Proceedings of the International Conference on Computational Thinking Education* (pp. 97-102). The Education University of Hong Kong.
- Kong, S. C. (2019). Components and methods of evaluating computational thinking for fostering creative problem-solvers in senior primary school education. In S. C. Kong, & H. Abelson (Eds.), *Computational thinking education* (pp.119-141). Springer Open Access.
- Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education*, 127(12), 178-189.
- Korkmaz, O., Cakir, R., & Ozden, M. Y. (2015). Bilgisayarca dusunme beceri duzeyleri olceginin ortaokul duzeyine uyarlanmasi [Computational thinking levels scale adaptation for secondary school level]. *Gazi Journal of Educational Science/ Gazi Egitim Bilimleri Dergisi*, 1(2), 67-86.

- Koroglu, I. S. (2015). Ustun yetenekli dijital yerlilerin sosyal medya kullanimlari uzerine nicel bir calisma [Social media usage of gifted digital natives: a quantitative study]. *Journal of Communication Theory & Research/ Iletisim Kuram ve Arastirma Dergisi*, 40(2), 266-290.
- Kukul, V., Gokcearslan, S., & Gunbatar, M. S. (2017). Computer programming self-efficacy scale for secondary school students: Development, validation and reliability. *Educational Technology Theory and Practice*, 7(1), 158-179.
- Lockwood, J., & Mooney, A. (2017, March 22). *Computational thinking in education: Where does it fit? A systematic literary review*. Cornell University Archive. <https://arxiv.org/abs/1703.07659>
- Lye, S. Z., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41(12), 51-61.
- Merriman, L. (2012). *Developing academic self-efficacy: Strategies to support gifted elementary school students* (Unpublished master's thesis). University of California.
- Ministry of National Education. (2013). *Ozel yetenekli bireyler strateji ve uygulama plani 2013-2017* [Strategy and implementation plan for gifted and talented individuals 2013-2017]. Ministry of National Education Publishing Service.
- Ongoz, S., & Sozel, H. K. (2018). Ustun yeteneklilerin egitiminde teknoloji kullanimi [Use of technology in gifted education]. In H. F. Odabasi (Ed.), *Ozel egitim ve egitim teknolojisi* [Special education and educational technology] (pp.91-114). Pegem Akademi.
- Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer selfefficacy. *Journal of Educational Computing Research*, 19(4), 367-381.
- Repenning, R., Webb, D.C., Koh, K.H., Nickerson, N., Miller, B., Brand, S.B., & Repenning, N. (2015). Scalable game design: a strategy to bring systemic computer science education to schools through game design and simulation creation. *Trantions on Computing Education*, 15(2), 1-31.
- Rodriguez, B. R. (2017). *Assessing computational thinking in computer science unplugged activities* (Unpublished master's thesis). Colorado School of Mines.
- Roman-Gonzalez, M., Perez-Gonzalez, J. C., Moreno-Leon, J., & Robles, G. (2016). Does computational thinking correlate with personality? The non-cognitive side of computational thinking. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 51-58). Association for Computing Machinery.
- Roman-Gonzalez, M., Perez-Gonzalez, J. C., Moreno-Leon, J., & Robles, G. (2018a). Can computational talent be detected? Predictive validity of the computational thinking test. *International Journal of Child-Computer Interaction*, 18(11), 47-58.
- Roman-Gonzalez, M., Perez-Gonzalez, J. C., Moreno-Leon, J., & Robles, G. (2018b). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, 80(3), 441-459.
- Roman-Gonzalez, M., Moreno-Leon, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In S. C. Kong, & H. Abelson (Eds.), *Computational thinking education* (pp.79-98). Springer Open Access.
- Sak, U. (2014). *Ustun zekalilar ozellikleri tanilanmalari egitimleri* [Gifted and talented students, their characteristics identification education]. Vize Yayıncılık.
- Sak, U., Ayas, M. B., Sezerel, B. B., Opengin, E., Ozdemir, N. N., & Gurbuz, S. D. (2015). Turkiye'de ustun yeteneklilerin egitiminin elestirel bir degerlendirmesi [Gifted and talented education in Turkey: critics and prospects]. *Turkish Journal of Giftedness and Education/Turk Ustun Zeka ve Egitim Dergisi*, 5(2), 110-132.
- Secer, I. (2015). *SPSS ve LISREL ile pratik veri analizi* [Practical data analysis with SPSS and LISREL]. (2nd Edition). Ani Yayıncılık.
- Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition. In J. Carter, I. Utting, & A. Clear (Eds.), *Proceedings of 18th Annual Conference on Innovation and Technology in Computer Science Education* (p.6). University of Southampton.
- Siegle, D. (2004). Identifying students with gifts and talents in technology. *Gifted Child Today*, 27(4), 30-34.
- Sullivan, A., Bers, M., & Mihm, C. (2017). Playing, and coding with KIBO: Using robotics to foster computational thinking in young children. In *Proc. CTE 2017 (International Conf. on Computational Thinking Education)* (pp. 110-115). The Education University of Hong Kong.

- University of Phoenix Research Institute. (2011). *Future work skills 2020*. http://www.iftf.org/uploads/media/SR-1382A_UPRI_future_work_skills_sm.pdf
- Unsal-Serim, E. (2019). *Investigation of computational thinking skills and selfefficacy perceptions of coding with coding education designed by gamification method* (Unpublished master's thesis). University of Balikesir.
- Tabachnick, B. G., & Fidell, L. S. (2013). *Using multivariate statistics* (6th ed.). Pearson Education.
- Tehseen, S., Ramayah, T., & Sajilan, S. (2017). Testing and controlling for common method variance: A review of available methods. *Journal of Management Sciences*, 4(2), 142-168.
- Tekinarslan, E., & Cetin, I. (2018). Bilissel, duyussal ve sosyal acidan programlama ogretimi [Programming teaching in terms of cognitive, affective and social issues]. In Y. Gulbahar, & H. Karal (Eds.), *Kuramdan uygulamaya programlama ogretimi* [Theory-to-practice programming teaching] (pp.68-89). Pegem Akademi. <https://doi.org/10.14527/9786052415092>
- Uzumcu, O., & Ucar, M. (2018, May 4-6). *Ustun yetenekli ogrencilerin temel duzeyde bilgi islemsel dusunme becerilerinin belirlenmesi* [Determination of gifted students' basic level of computational thinking skills] [Paper presentation].V. Gifted and Education Congress, Gaziantep, Turkey.
- Waite, J. (2017, May 13). Pedagogy in teaching computer science in schools: A Literature Review. Queen Mary University of London and King's College London. <https://royalsociety.org/-/media/policy/projects/computing-education/literature-review-pedagogy-in-teaching.pdf>
- Weese, J. L., & Feldhausen, R. (2017, June 25-28). *STEM outreach: assessing computational thinking and problem solving*. [Paper presentation]. ASEE Annual Conference & Exposition, Columbus, Ohio.
- Wing, J. M. (2011, November 17). Computational thinking: What and why? <http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- Yildiz-Durak, Karaoglan-Yilmaz, F. G., & Yilmaz, R. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Educational Technology*, 10(2), 173-197.
- Yildiz-Durak, H., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, 116(1), 191-202.