World Journal on Educational Technology: Current Issues

Volume 10, Issue 3, (2018) 153-164

www.wj-et.eu

Evaluating programming self-efficacy in the context of inquiry skills and problem-solving skills: A perspective from teacher education

Ahmet Naci Çoklar*, Necmettin Erbakan University, Ahmet Kelesoglu Faculty of Education, Computer Education and Instructional Technologies, Konya.

Arif Akçay, Kastamonu University, Faculty of Education, Computer Education and Instructional Technologies, Kastamonu.

Suggested Citation:

Çoklar, A., N. & Akçay, A. (2018). Evaluating programming self-efficacy in the context of inquiry skills and problem-solving skills: A perspective from teacher education. *World Journal on Educational Technology: Current Issues*. *10*(3), 153-164.

April 15, 2018; revised date May 01, 2018; accepted date June 16, 2018.

Selection and peer review under responsibility of Prof. Dr. Servet Bayram, Yeditepe University, Turkey.

This article is based on the master's thesis written by the second author under the supervision of the first author.

Abstract

It is stated that programming skills, which are ever-increasingly becoming crucial nowadays, have an impact on cognitive processes and promote individual's self-efficacy in different ways. It is noted that students with programming skills have the character of problem solving, inquiry and being able to make decisions, which are all requisites of today's education system.. The purpose of this research is to examine the programming self-efficacy of Information Technologies and Software teacher candidates, who are trained in the field of programming. In total, 707 students, who are studying in the Department of Computer and Instructional Technologies Education of the Faculty of Education of eight universities in Turkey, participated in this research. The results of the research demonstrate that students' programming self-efficacy is at medium level, whereas high level is observed for their inquiry and problem-solving skills. Also, inquiry skills have a significant impact on their perceived programming self-efficacy; whereas problem-solving skills has not any impact on it.

Keywords: Inquiry skills, problem-solving skills, programming, self-efficacy.

* ADDRESS FOR CORRESPONDENCE: **Ahmet Naci Çoklar*,** Necmettin Erbakan University, Ahmet Kelesoglu Faculty of Education, Computer Education and Instructional Technologies, Konya. *E-mail address*: acoklar@konya.edu.tr / Tel.: +0332 323 82 20

1. Introduction

Programming is defined as strategically using instructions in certain premises by teaching students and the structure of these instructions in a particular (programming) language (Palumbo, 1990). Another definition states that programming is the understanding of techniques related to problem-solving; and the process of compiling syntax and structural components related to this (Oddie, Hazlewood, Blakeway & Whitfield, 2010). It can be argued that the one, who create a program on the basis of this definition, can bring about algorithmic solutions and products by analysing the problem (Saeli et al., 2011). In this process, learning programming skills, knowing and using at least one programming language provide numerous benefits to the user.

When the advantages of programming skills for students are considered, there are more than one purpose of the programming trainings provided by K-12, higher education and various private or public organisations and institutions. Trainings provided by organisations aim to address the lack of qualified personnel in this area; whereas trainings provided by educational institutions target students to have acquired skills required in today's education system. In relation to that, learning programming skills brings about many career opportunities and educational attainments for that individual (Israel, Wherfel, Pearson, Shehab & Tapia, 2015). Thinking of the some of these attainments, programming makes people acquire organisation and planning skills (Siegle, 2009). When students use programming applications, their spatial awareness increases; their mathematical thinking skills develop, their ability to evaluate the status of the problem and logical reasoning increases (Calder, 2010). Researchers, who studied the impact of learning programming at early ages, found that reflectivity and creativity competencies are higher in students with programming skills compared to those with no programming skills (Clements & Gullo, 1984; Kim, Chung & Yu, 2013). Critical thinking skills of students improve with visual game development, which is one of the programming applications (Dekhane, Xu & Tsoi, 2013). Additionally, Akpinar and Altun (2014) argue that when students learn programming skills, most of the concepts in mathematics can be understood more easily; problem-solving and analytical thinking skills improve, cooperative working and learning skills develop and their digital literacy improve as they will constantly use information technologies. Students with programming skills being more successful in fields such as maths, physics, go and chess can be explained with them as of having more advanced knowledge (Adelson, 1981; Robins, Rountree & Rountree, 2003); and even with having programming knowledge developing these skills for them.

Although programming is defined as the process of producing programs in terms of learning outcomes of the programming education for students in its most important and simplest way, it is an aggregate of skills which are not as simple as its definition. Despite the fact that it is easy to use a program or application on technological devices, the development of these programs is not seen as an easy process (Oddie et al., 2010). Programming is an important competency to achieve the 21st century student standards (innovative designer, data-processing thinker, etc.) (International Society for Technology in Education (ISTE), 2016), which is yet hard to comprehend, improve and master (Yukselturk & Altiok, 2017). Courses to learn this competency are considered to be rather difficult particularly for beginner students (Askar & Davenport, 2009). Students think of programming as difficult and abstract skills (Oddie et al., 2010); and find it challenging to learn the logic of programming (de Raadt, 2007; Holvikivi, 2010). In this sense, the basic programming education generally fails to motivate students for software studies (Holvikivi, 2010; Vanicek, 2015). There are several factors for learning and applying programming skills to be such a difficulty.

Since each sub-function of programming skills requires cognitive processes representing a different field of knowledge, they are complex skills (Ambrosio, Costa, Almeida, Franco & Macedo, 2011; Holvikivi, 2010; Yurdugul & Askar, 2013). Consequently, it is observed that there are several variables affecting the programming skills. In a study examining difficulties in learning programming skills conducted by Ozmen and Altun (2014), there are three main themes named as personal problems (biases, obliviousness, etc.), problems with in-class activities (limited time, insufficient practice, etc.)

and problems within programming process (lack of information, not being able to write algorithms, etc.). Saeli et al. (2011) state that programming skills of creative use, being able to read the program and finding effective solutions for problems have an impact on the success of the programming skills. In order for a student to be successful in programming, there is a need for systematic work patterns, abstract thinking, sufficient cognitive capacity, analytical intelligence and establishing cognitive diagrams regarding procedural thinking and implementation (Holvikivi, 2010; Israel et al., 2015). While visualisation in writing programs is important for students to be successful in programming skills, various cognitive models are effective in reading the program (Kumar, 2003). Moreover, it can be argued that determination and self-discipline are necessary as well when processes in the structure of programming applications such as debugging and deciphering are considered (Siegle, 2009).

Another skill exhibited by students in the programming process is the inquiry skills. Inquiry skills and cause and effect analysis are essential parts of the debugging operations in computer programming processes. The student is encouraged to make an inquiry regarding why the program is not working during the debugging process (Sullivan, 2008). Students produce new forms based on the information they acquired from a failed testing (Altin, Pedaste & Aabloo, 2011). In programming, students use skills related to inquiry skills such as controlling for variables, improving and testing hypotheses, generalisation and testing, finding solutions in the debugging process during syntax and use of language (Pedaste & Sarapuu, 2006; Sullivan, 2008). Apart from using inquiry skills in the programming process, programming is the development of programs intended to find a solution to a particular problem according to its definition and implementation (Oddie et al., 2010; Saeli et al., 2011). Hence, programming is generally based on the solution of a problem (Govender, 2007). Thus, in this study, an analysis of programming self-efficacy perceptions in the context of inquiry and problem-solving skills is found to be worth researching and aimed.

In Turkey, individuals, who are trained in programming skills and have the proficiency in providing programming trainings, are Information Technology and Software teachers. Information Technology and Software teachers are educated in the Department of Computer Education and Instructional Technologies (CEIT) of Faculties of Education in Turkey. Students graduate from CEIT, which is established in 1998, by completing 49 courses with 142 credits. Two of these courses are Programming Languages I and II. These courses are opened in fall and spring semesters as 3 hours of theoretical knowledge and 2 hours of practicing. In addition, students also have many courses such as web-based Programming, database management and web design (Turkish Council of Higher Education, 2007). While stating the competencies required for graduates of CEIT, ISTE, which is an international community setting standards regarding skills for students and teachers in the field of education, underlines that they can use modern and advanced level programming language; develop programs incorporating simple and structured data types, compound Boolean statements, ordinal, conditional and repetitive control structures; solve problems with algorithms and programming language using advanced data structures; and denotes that two or more developing environments can be used actively (ISTE, 2011). At the same time, Turkish Council of Higher Education (2007) aims that graduates of CEIT have the capacity to teach programming skills. Fessakis, Gouli and Mavroudi (2013) note that the teacher facilitates the learning of students and has a role to encourage and support students to overcome difficulties in programming. In this regard, it is important to analyse programming self-efficacy perceptions of Information Technologies and Software teacher candidates in the context of inquiry skills and problem-solving skills.

1.1. Research problems

- What is the opinion of Information Technologies and Software teacher candidates on their level of computer programming self-efficacy perceptions, inquiry skills and problem-solving skills regarding programming?
- Do Information Technologies and Software teacher candidates' computer programming self-efficacy perceptions differ on the basis of their inquiry skills?
- Do Information Technologies and Software teacher candidates' computer programming self-efficacy perceptions differ on the basis of their problem-solving skills?

2. Method

2.1. Research model

This research applies a quantitative method. The research design is a causal-competitive model. This research model intends to identify the causes of an existing situation, variables affecting these causes or consequences of an effect (Fraenkel, Wallen & Hyun, 2012).

2.2. Population and sample

The population of this research consisted of students from the Department of CEIT in Faculties of Education in Turkey between 2014 and 2015. Due to the size of the population, the sampling method is used; and Information Technology teacher candidates studying in Faculties of Education in eight different universities of three geographically different regions are included in the research by using cluster sampling. It is not the single unit but the group that the unit is connected in the sample, which is selected through cluster sampling, is selected; and it is effective in studies with many groups like CEIT (Fraenkel et al., 2012). There are 707 students in total who participated in this research. In terms of gender, 49.2% of the students that participated in the research are males; whereas 50.8% of them are female.

2.3. Data collection tools and collecting data

In the research, a survey form with four sections, titled as demographic information form, computer programming self-efficacy scale, inquiry skills scale and problem-solving skills inventory, is used.

Computer programming self-efficacy scale is developed by Ramalingam and Wiedenback (1999); and adapted to Turkish by Altun and Mazman (2012). Turkish version of the scale, which is used to measure the perception of self-efficacy in programming, consisted of nine items and two factors (Performing simple programming tasks and performing complex programming tasks). For the scale as a whole, Cronbach's alpha coefficient is calculated to be 0.928, and McDonald's ω (omega) coefficient is found to be 0.956.

Inquiry skills scale is developed by Aldan Karademir and Saracaloglu (2013) and used for identifying inquiry skills of teacher candidates. The reliability coefficient of the scale, which consisted of 14 items and three factors (Acquiring knowledge, controlling knowledge and self-confidence) is, respectively, 0.76 for 'Knowledge Acquisition, 0.66 for 'Controlling Knowledge', and 0.82 for 'Self-Confidence'. The overall Cronbach's alpha coefficient for the scale as a whole is 0.82.

Problem-solving inventory is developed by Heppner and Petersen (1982) in order to measure problem-solving skills of teacher candidates; and adapted to Turkish by Sahin, Sahin and Heppner (1993). The inventory is composed of 35 items and prepared as six-point likert scale. In the factor analysis, it is found that there are six factors titled as 'Impetuous Approach', 'Deliberative Approach', 'Avoidant Approach', 'Evaluator Approach', 'Self-Confident Approch' and 'Planned Approach' (Savasir & Sahin, 1997). The Cronbach's alpha reliability coefficient of the scale is calculated as 0.88.

In the context of the research, these measuring instruments are collected in one form and reproduced in sufficient numbers. Permissions that are required for data collection are obtained from CEIT in universities; and data collection is completed by the researcher by visiting these departments.

2.4. Data collection and interpretation

Arithmetic mean and standard deviation (SD) from descriptive statistics are employed for the analysis of data collected with these data collection tools. The unpaired *t*-test is applied to examine the impact of inquiry skills and problem-solving skills on the level of perceived programming self-efficacy. SPSS 22.0 (Statistical Package for the Social Sciences) is used for statistical analysis of the data, and the level of significance is identified as 0.05.

The data collected from students are examined in order to interpret the results of the analysis; the following two evaluation criteria are applied to determine and to provide convenience for interpretation: low, medium and high level for programming self-efficacy perceptions and low and high level for inquiry skills and problem-solving skills. Based on these, the below mentioned evaluation criteria is specified and applied (Table 1).

Table 1. Evaluation criteria

	Range of evaluation	Evaluation criteria
Programming self-efficacy	1.00–3.00	Low level
	3.01–5.00	Medium level
	5.01–7.00	High level
Inquiry skills	1.00–3.00	Low level
	3.01–5.00	High level
Problem-solving skills	1.00–3.50	Low level
	3.51–6.00	High level

3. Findings

The purpose of this research is to analyse the programming self-efficacy perceptions of Information Technologies and Software teacher candidates in the context of inquiry skills and problem-solving skills. Findings related to the data obtained from 707 CEIT students that participated in the research are given in sections.

3.1. Students' opinion on their level of perceived programming self-efficacy references

In order to discover students' programming self-efficacy perceptions, the data are collected through a survey consisted of dimensions of performing simple programming skills and performing complex programming skills. The results of these data are provided below (Table 2).

Table 2. General and sub dimensions of the state of students' opinion on their level of perceived programming self-efficacy

Dimensions of computer programming self-efficacy	$\bar{X}$	SD
Performing simple programming tasks	5.52	1.92
Performing complex programming tasks	3.90	1.72
General	4.44	1.61

In Table 2, students' computer programming self-efficacy perceptions are analysed in terms of general and sub dimensions. While students had a high level ($\bar{X}$ = 5.52) of self-efficacy in performing simple programming tasks; they have a medium level ($\bar{X}$ = 3.90) of self-efficacy in performing complex programming tasks sub-dimension. In other words, students stated that they would not have any problems in performing simple programming tasks; however, they had lower self-efficacy when they were performing complex programming tasks. In general, it is found that they have a medium level self-efficacy considering their programming self-efficacy perceptions.

3.2. Students' opinion on their level of inquiry skills

In order to discover students' inquiry skills, the data are collected through a survey consisted of dimensions of acquiring knowledge, controlling knowledge and self-confidence. The results of these data are provided below (Table 3).

Table 3. General and sub dimensions of the state of students' opinion on their inquiry skills

Dimensions of inquiry skills	$\bar{X}$	SD
Acquiring knowledge	4.19	0.58
Controlling knowledge	3.69	0.69
Self-confidence	3.56	1.04
General	3.88	0.56

In Table 3, it is seen that students' level of acquiring knowledge is very high ($\bar{X}$ = 4.19). In other words, students stated that they did not have difficulties in acquiring knowledge. They also underlined that they were proficient enough in controlling knowledge ($\bar{X}$ = 3.69) and had self-confidence in inquiry skills ($\bar{X}$ = 3.56). Students have high averages for dimensions of acquiring knowledge, controlling knowledge and self-confidence. However, it can be argued that they are particularly good at acquiring knowledge. In general, it is found that they have high level ($\bar{X}$ = 3.88) inquiry skills.

3.3. Students' opinion on their level of problem-solving skills

In order to discover students' problem-solving skills, the data are collected through a survey consisted of dimensions of impetuous approach, deliberative approach, avoidant approach, evaluative approach, self-confident approach and planned approach. The results of these data are provided in Table 4.

Table 4. General and sub dimensions of the state of students' opinion on their problem-solving skills

Dimensions of problem-solving skills	$\bar{X}$	SD
Impetuous approach	3.56	0.67
Deliberative approach	4.36	1.03
Avoidant approach	4.27	1.06
Evaluative approach	4.10	1.14
Self-confident approach	4.23	0.98
Planned approach	4.34	1.07
General	4.06	0.63

When students' problem-solving approaches are examined in Table 4, it is seen that they have a rather low level impetuous approach ($\bar{X}$ = 3.56). Students' levels of deliberative approach ($\bar{X}$ = 4.36), avoidant approach ($\bar{X}$ = 4.27), evaluative approach ($\bar{X}$ = 4.10), self-confident approach ($\bar{X}$ = 4.23) and planned approach ($\bar{X}$ = 4.34) to problems are high. It can also be argued that students have tendency towards thinking, planning, evaluating and having self-confidence and being rather impetuous. It can be commented that most of the students state having high levels of problem-solving skills ($\bar{X}$ = 4.06).

3.4. An analysis of students' opinions on their level of perceived programming self-efficacy on the basis of their inquiry skills and problem-solving skills

In order to determine the impact of students' inquiry skills on their opinions on programming self-efficacy, the average scores of students' programming self-efficacy with low and high level inquiry skills are compared (Table 5).

Table 5. *t*-test results of teacher candidates' dimensions of computer programming self-efficacy according to their level of inquiry skills

	Level of inquiry skills	*N*	$\bar{X}$	SD	*df*	*t*	*p*
Performing simple programming	Low	225	5.17	2.04	705	−3.29	0.001
tasks	High	482	5.68	1.85			
Performing complex	Low	225	3.39	1.61	705	−5.61	0.001
programming tasks	High	482	4.14	1.73			
General computer programming	Low	225	3.99	1.56	705	−5.22	0.001
self-efficacy	High	482	4.66	1.60			

In Table 5, it is seen that the level of students' inquiry skills have a significant difference on their perceived programming self-efficacy in general ($t = -5.22$; $p < 0.05$). In other words, it can be claimed that programming self-efficacy for students with low level inquiry skills ($\bar{X}$ = 3.99) is lower than the programming self-efficacy for students with high level inquiry skills ($\bar{X}$ = 4.66). Therefore, it can be argued that if the inquiry skills of teacher candidates improve, their self-efficacy in programming skills also improve.

When the impact of inquiry skills on performing simple programming tasks is analysed, it is found that there is a significant difference between groups ($t = -3.29$; $p < 0.05$). Self-efficacy in performing simple programming tasks for teacher candidates with high level inquiry skills ($\bar{X}$ = 5.68) is higher compared to those with low level inquiry skills ($\bar{X}$ = 5.17). Furthermore, it is seen that students' level of inquiry skills have a significant impact on their self-efficacy in complex programming skills ($t = -5.61$; $p < 0.05$). It is observed that students with high level inquiry skills ($\bar{X}$ = 4.14) have higher self-efficacy in performing complex programming tasks compared to those with low level inquiry skills ($\bar{X}$ = 3.39). According to these findings, it can be argued that students' self-efficacy in performing both simple and complex programming tasks would improve as their inquiry skills improve.

In order to determine the impact of students' problem-solving skills on their opinions on programming self-efficacy, the average scores of students' programming self-efficacy perceptions with low and high level problem-solving skills are compared (Table 6).

Table 6. *t*-test results of teacher candidates' dimensions of computer programming self-efficacy according to their level of problem-solving skills

	Level of problem-solving skills	*N*	$\bar{X}$	SD	*df*	*t*	*p*
Performing simple programming	Low	578	5.54	1.94	705	0.43	0.668
tasks	High	129	5.54	1.88			
Performing complex	Low	578	3.91	1.73	705	0.22	0.828
programming tasks	High	129	3.87	1.73			
General computer programming	Low	578	4.45	1.63	705	0.33	0.739
self-efficacy	High	129	4.40	1.60			

In Table 6, it is seen that students' problem-solving skills have no significant difference on their self-efficacy in programming in general ($t = 0.33$; $p > 0.05$). In other words, it can be argued that there is no statistically significant difference between perceptions of students with low ($\bar{X}$ = 4.45) and high ($\bar{X}$ = 4.40) problem solving skills regarding their computer programming self-efficacy. According to this finding, it can be commented that problem-solving skills have no impact on programming self-efficacy perceptions in general.

When the impact of problem-solving skills on performing simple programming tasks is analysed, it is found that there is no significant difference between Information Technology and Software teacher candidates with low level problem-solving skills and those with high level problem-solving skills ($t = 0.43$; $p > 0.05$). The average score of teacher candidates with low and high level problem-solving skills for performing simple programming tasks is $\overline{X} = 5.54$. Similarly, with the case of performing complex programming tasks, it is found that there is no significant difference between students with low level and high level problem-solving skills ($t = 0.22$; $p > 0.05$). While the average score of students with low level problem-solving skills for performing complex programming tasks is $\overline{X} = 3.91$; the average score of students with high levels is $\overline{X} = 3.87$, and there is no statistically significant difference between them. Based on these findings, it can be argued that students' problem-solving skills have no impact on their self-efficacy in performing neither simple nor complex programming tasks.

4. Results and discussion

Nowadays, different countries started to provide programming courses in their curriculum with differentiating course levels. With these courses, it is aimed to improve cognitive skills rather than having all individuals as programmers working in this sector (Palumbo, 1990). In relation to this purpose, inquiry skills and problem-solving skills of programming are surveyed with the participation of university students, who are also prospective teachers. The data are collected from 707 teacher candidates who are students in CEIT, and results are presented below.

According to the research, it is observed that the perceptions of Information Technologies and Software teacher candidates on self-efficacy in programming in general are at a medium level. As they demonstrate high level performance to perform simple programming tasks, they only show medium level performance when performing complex programming tasks. As a finding supporting this research, Yurdugul and Askar (2013) stated that CEIT students have average level programming skills. Moreover, in their study, Yukselturk and Altiok (2017) also found that students have medium level programming self-efficacy perceptions; and their complex programming skills would improve once they are trained with Scratch programming tool. According to another research conducted by Mazman and Altun (2013), the level of self-efficacy in programming for CEIT students is at a medium level and this level can be improved with programming trainings. Consequently, the trainings provided aim to have CEIT graduates who are capable of teaching programming skills (Turkish Council of Higher Education, 2007). Since they have no intention of these students to find jobs like computer engineers do in the sector, having medium level skills can be considered normal. However, a research conducted by the Turkish Ministry of Education (2007) demonstrates that Information Technologies and Software teachers find their programming skills as insufficient. Alongside with these findings, it can be claimed that there is a parallelism in the literature regarding having Information Technologies and Software teacher candidates' self-efficacy as medium level, despite the fact that these teachers consider themselves as insufficient.

While CEIT students have inquiry skills in general, they also have no problems with acquiring knowledge skills and possess controlling knowledge skills and self-confidence. In a study conducted by Engin (2009), it is argued that teachers in Turkey are confident in terms of their inquiry skill. However, Tanisli (2013) noted that teacher candidates' inquiry skills, in general, could be qualified as a novice. As a result of studies, the reason for differences in inquiry levels of teacher candidates can be due to the qualifications of their departments.

Considering the approaches of Information Technology and Software teachers towards problem-solving skills, it is observed that their level of impetuous approach is rather low and other approaches (deliberative approach, avoidant approach, evaluative approach, self-confident approach and planned approach) are high. It can be argued that problem-solving skills are high. Ozkutuk, Silku, Orgun and Yalcinkaya (2003) showed that Information Technology and Software teacher candidates have low level impetuous approach in problem-solving approaches; and high level deliberative approach, avoidant approach, evaluative approach, self-confident approach and planned approach. Additionally, as a finding to support the results of this research, it is proved that their problem-solving skills in general is high (Ozkutuk et al., 2003). However, a research conducted by Akyuz (2012) conclude that Information Technologies and Software teacher candidates are at low level in all sub-dimensions on problem-solving inventory. Alongside with this, many researchers also stated in their studies that teacher candidates' problem solving skills are low (Genc & Tinmaz, 2010; Kazu & Ersozlu, 2008; Ocak & Egmir, 2014; Schreglmann & Dogruluk, 2012). The reason for having different results for different researchers can be having students studying in different universities.

Another finding of the research is that students' inquiry skills have an impact on their programming self-efficacy perceptions. In parallel with this finding, Zacharia and Louca (2011) argued that computer-based programming environments support inquiry skills; and inquiry skills are put in use in these environments. The research conducted by Vanicek (2015) on programming skills training on Scratch application with an evaluative approach can also be considered in parallel with this study. Moreover, Altin et al. (2011) claimed that problem-solving skills, inquiry skills and programming skills are effective in robotic programming, which is another field of application for programming skills. Findings in various studies claiming that trial-error and debugging processes in using programming skills urge students to question, inquiry skills are important, deeper and more meaningful learning can be achieved in environments based on inquiry are all in support of the result of this research (Coleman & Nichols, 2011; Maeots, Padeste & Sarapuu, 2008). Based on these findings, it can be argued that inquiry skills are put in use in programming processes; and as a result they have an impact on perceived programming self-efficacy.

According to findings of the study, Information Technologies and Software teacher candidates' problem-solving skills have no impact on their programming self-efficacy. In parallel with this result, based on the findings of studies conducted with various methods and groups, problem-solving skills have no significant impact on programming skills (Kalelioglu & Gulbahar, 2014). However, in a study conducted by Akcaoglu (2014), it is argued game developing activities that present programming skills create appropriate environments for using problem-solving skills. Holvikivi (2010) argued that a failed programming experience may result from low problem-solving skills. Moreover, numerous studies in the literature found that problem-solving skills have an impact on programming skills (Adelson, 1981; Ismail, Ngah & Umar, 2010; Kim et al., 2013; Liao & Bright, 1991; Robins et al., 2003; Yurdugul & Askar, 2013). It is thought that differences in findings of this research may result from the differences in structure of programming languages or difference in methods being used in the working group.

This research is limited to only CEIT students. A similar research can be conducted with students from the Department of Computer studying in different faculties (Faculties of Engineering, Programming and Education). On the other hand, considering the problem-solving-age relation, it can be suggested to have research with students from different grades such as primary and secondary school whom received programming education at earlier ages.

References

Adelson, B. (1981). Problem solving and the development of abstract categories in programming languages. *Memory & Cognition, 9*(4), 422–433. doi: 10.3758/BF03197568

Akcaoglu, M. (2014). Learning problem-solving through making games at the game design and learning summer program. *Educational Technology Research and Development, 62*(5), 583–600. https://doi.org/10.1007/s11423-014-9347-4

Akpinar, Y. & Altun, A. (2014). Bilgi toplumu okullarinda programlama egitimi gereksinimi [The requirements of programming education in the information society schools]. *Elementary Education Online, 13*(1), 1–4. Retrieved from http://ilkogretim-online.org.tr/index.php/io/article/view/2099

Akyuz, H. I. (2012). *Cevrimici gorev temelli ogrenme ortaminda egitsel ajanin rolunun ve bicim ozelliklerinin ogrencilerin motivasyonuna, bilissel yuklenmesine ve problem cozme becerisi algisina etkisi* [*The effects of pedagogical agents' role and property of modality on students' motivation, cognitive load and problem solving perception in online task based learning environment*] (Ph.D. thesis). Ankara University, Institute of Educational Sciences, Ankara, Turkey. Retrieved from http://acikarsiv.ankara.edu.tr/browse/24783/Halil%20IbrahimAKYUZ.pdf

Aldan Karademir, C. & Saracaloglu, A. S. (2013). Sorgulama becerileri olcegi'nin gelistirilmesi: Gecerlik ve guvenirlik calismasi [The development of inquiry scale: reliability and validity study]. *Asian Journal of Instruction, 1*(2), 56–65. Retrieved from http://dergipark.gov.tr/download/article-file/17636

Altin, H., Pedaste, M. & Aabloo, A. (2011). Educational robotics and inquiry learning: a pilot study in a web-based learning environment. In *Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies* (pp. 224–226). doi: 10.1109/ICALT.2011.72

Altun, A. & Mazman, S. G. (2012). Programlamaya iliskin oz yeterlilik algisi olceginin Turkce formumun gecerlilik ve guvenirlik calismasi [Reliability and validity study on Turkish version of perceived self-efficacy scale for programming]. *Egitimde ve Psikolojide Olcme ve Degerlendirme Dergisi, 3*(2), 297–308. Retrieved from http://dergipark.ulakbim.gov.tr/epod/article/view/5000045510

Ambrosio, A. P., Costa, F. M., Almeida, L., Franco, A. & Macedo, J. (2011). Identifying cognitive abilities to improve CS1 outcome. In *Proceedings—Frontiers in Education Conference—FIE 2011* (pp. F3G-1-F3G-7). doi: 10.1109/FIE.2011.6142824

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. Retrieved from https://files.eric.ed.gov/fulltext/ED503900.pdf

Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom, 15*(4), 9–14. Retrieved from https://files.eric.ed.gov/fulltext/EJ906680.pdf

Clements, D. H. & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology, 76*(6), 1051–1058. doi: 10.1037/0022-0663.76.6.1051

Coleman, S. A. & Nichols, E. (2011). Embeding inquiry based learning into programming via pair assessment. *Innovation in Teaching and Learning in Information and Computer Sciences, 10*(1), 72–77. doi: 10.11120/ital.2011.10010072

de Raadt, M. (2007). A review of Australasian investigations into problem solving and the novice programmer. *Computer Science Education, 17*(3), 201–203. doi: 10.1080/08993400701538104

Dekhane, S., Xu, X. & Tsoi, M. Y. (2013). Mobile app development to increase student engagement and problem solving skills. *Journal of Information Systems Education, 24*(4), 299–308. Retrieved from http://jise.org/Volume24/n4/JISEv24n4p299.pdf

Engin, G. (2009). *Sinif ogretmeni adaylarinin sorgulama yaklasimini algilama ve ogretim becerilerinin arastirilmasi: Turkiye-Hollanda karsilastirma calismasi* [*The investigation of pre-service elementary teachers' perception of inquiry approach and teaching skills: the case of the Netherland and Turkey*] (Master thesis). Ege University, Institute of Social Sciences, Izmir, Turkey. Retrieved from http://www.acikerisim.ege.edu.tr:8081/jspui/handle/11454/3926

Fessakis, G., Gouli, E. & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: a case study. *Computers & Education, 63*, 87–97. doi: 10.1016/j.compedu.2012.11.016

Fraenkel, J. R., Wallen, N. E. & Hyun, H. H. (2012). *How to design and evaluate research in education*. New York, NY: McGraw-Hill.

Genc, Z. & Tinmaz, H. (2010). *Programlama dilleri dersi icerisinde ogrencilerin problem cozme becerileri ve bilgisayar tutumlari degerlendirmesi: Bir Firat Universitesi ornegi* [*Evaluation of students' problem solving skills and attitude toward computers in a programming course: a case study from Firat University*]. 2010

International Education Technology Conference, Istanbul, Turkey. Retrieved from https://www.academia.edu/240634/Programlama_Dilleri_Dersi_%C4%B0%C3%A7erisinde_%C3%96%C4%9Frencilerin_Problem_%C3%87%C3%B6zme_Becerileri_ve_Bilgisayar_Tutumlar%C4%B1_De%C4%9Ferlendirmesi_Bir_F%C4%B1rat_%C3%9Cniversitesi_%C3%96rne%C4%9Fi

Govender, I. (2007). Experiences of learning and teaching: problem solving in computer programming. *African Journal of Research in Mathematics, Science and Technology Education, 11*(2), 39–50. doi: 10.1080/10288457.2007.10740620

Heppner, P. P. & Petersen, C. H. (1982). The development and implications of a personal problem-solving inventory. *Journal of Counseling Psychology, 29*(1), 66–75. doi: 10.1037/0022-0167.29.1.66

Holvikivi, J. (2010). Conditions for successful learning of programming skills. In N. Reynolds & M. Turcsanyi-Szabo (Eds.), *Key competencies in the knowledge society* (pp. 155–164). Berlin, Heidelberg, Germany: Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-15378-5_15

International Society for Technology in Education. (2011). *ISTE standards for computer science educators*. Retrieved from https://www.iste.org/standards/for-computer-science-educators.

International Society for Technology in Education. (2016). *ISTE standards for students*. Retrieved from https://www.iste.org/standards/for-students.

Ismail, M. N., Ngah, N. A. & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: a need assessment analyses. *Turkish Online Journal of Educational Technology, 9*(2), 125–131. Retrieved from http://www.tojet.net/articles/v9i2/9214.pdf

Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S. & Tapia, T. (2015). Empowering K-12 students with disabilities to learn computational thinking and computer programming. *Teaching Exceptional Children, 48*(1), 45–53. doi: 10.1177/0040059915594790

Kalelioglu, F. & Gulbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: a discussion from learners' perspective. *Informatics in Education, 13*(1), 33–50. Retrieved from https://eric.ed.gov/?id=EJ1064285

Kazu, H., & Ersozlu, Z. N. (2008). *Ogretmen adaylarinin problem cozme becerilerinin cinsiyet, bolum ve OSS puan turune gore incelenmesi* [A study on the problem solving skills of candidate teachers according to gender, department and university entrance examination score types]. *Abant Izzet Baysal Universitesi Egitim Fakultesi Dergisi, 8*(1), 161–172. Retrieved from http://dergipark.ulakbim.gov.tr/aibuefd/article/view/5000091363

Kim, S., Chung, K. & Yu, H. (2013). Enhancing digital fluency through a training program for creative problem solving using computer programming. *The Journal of Creative Behavior, 47*(3), 171–199. doi: 10.1002/jocb.30

Kumar, A. N. (2003). Learning programming by solving problems. In L. Cassel & R. Reis (Eds.), *Informatics curricula and teaching methods* (pp. 29–39). Boston, MA: Springer. doi: 10.1007/978-0-387-35619-8_4

Liao, Y.-K. C. & Bright, G. W. (1991). Effects of computer programming on cognitive outcomes: a meta-analysis. *Journal of Educational Computing Research, 7*(3), 251–268. doi: 10.2190/E53G-HH8K-AJRR-K69M

Mazman, S. G. & Altun, A. (2013). Programlama—I dersinin BOTE bolumu ogrencilerinin programlamaya iliskin oz yeterlilik algilari uzerine etkisi [The effect of introductory to programming course on programming self efficacy of CEIT students]. *Journal of Instructional Technologies & Teacher Education, 2*(3), 24–29. Retrieved from http://dergipark.ulakbim.gov.tr/jitte/article/view/1092000029/0

Maeots, M., Pedaste, M. & Sarapuu, T. (2008). Transforming students' inquiry skills with computer based simulations. In *8th IEEE International Conference on Adavanced Learning Technologies* (pp. 938–942). Santander, Spain: IEEE Computer Society Press. doi: 10.1109/ICALT.2008.239

Ocak, G. & Egmir, E. (2014). Ogretmen adaylarinin problem cozme becerilerinin cesitli degiskenler acisindan incelenmesi [Investigation of pre-service teachers' problem solving skills in terms of different variables]. *Asian Journal of Instruction, 2*(1), 27–45. Retrieved from http://dergipark.ulakbim.gov.tr/aji/article/view/1064000033

Oddie, A., Hazlewood, P., Blakeway, S. & Whitfield, A. (2010). Introductory problem solving and programming: robotics versus traditional approaches. *Innovations in Teaching & Learning in Information & Computer Sciences, 9*(2), 1–11. doi: 10.11120/ital.2010.09020011

Ozkutuk, N., Silku, H. A., Orgun, F. & Yalcinkaya, M. (2003). Ogretmen adaylarinin problem cozme becerileri [The problem-solving skills of teacher candidates]. *Ege Egitim Dergisi, 2*(3), 1–9. Retrieved from http://dergipark.gov.tr/download/article-file/17639

Ozmen, B. & Altun, A. (2014). Undergraduate student's experiences in programming: difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry, 5*(3), 9–27. doi: 10.17569/tojqi.20328

Palumbo, D. B. (1990). Programming language/problem solving research: a review of relevant issues. *Review of Educational Research, 60*(1), 65–89. doi: 10.3102/00346543060001065

Pedaste, M. & Sarapuu, T. (2006). Developing an effective support system for inquiry learning in a web-based environment. *Journal of Computer Assisted Learning, 22*, 47–62. doi: 10.1111/j.1365-2729.2006.00159.x

Ramalingam, V. & Wiedenbeck, S. (1999). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research, 19*(4), 367–381. doi: 10.2190/C670-Y3C8-LTJ1-CT3P

Robins, A., Rountree, J. & Rountree, N. (2003). Learning and teaching programming: a review and discussion. *Computer Science Education, 13*(2), 137–172. doi: 10.1076/csed.13.2.137.14200

Saeli, M., Perrenet, J., Jochems, W. M. G., Zwaneveld, B., Nederland, O. U. & Centrum, R. D. M. (2011). Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education, 10*(1), 73–88. doi: 10.1145/2016911.2016943

Sahin, N., Sahin, N. H. & Heppner, P. P. (1993). Psychometric properties of the problem solving inventory in a group Turkish university students. *Cognitive Therapy and Research, 17*(4), 379–396. doi: 10.1007/BF01177661

Savasir, I. & Sahin, N. H. (1997). *Bilissel-davranisci terapilerde degerlendirme: Sik kullanilan olcekler [Assessment in cognitive-behavioral therapies: Frequently used scales].* Ankara, Turkey: Turk Psikologlar Dernegi.

Schreglmann, S. & Dogruluk, S. (2012). Bilisim teknolojileri ogretmen adaylarinin problem cozme becerilerinin cesitli degiskenler acisindan incelenmesi [Investigation of problem solving skills of prospective Information Technologies teachers in terms of different variables]. *Amasya Educational Journal, 1*(2), 143–150. Retrieved from http://dergipark.gov.tr/download/article-file/19595

Siegle, D. (2009). Developing student programming and problem-solving skills with visual basic. *Gifted Child Today, 32*(4), 24–29. doi: 10.1177/107621750903200408

Sullivan, F. R. (2008). Robotics and science literacy: thinking skills, science process skills and understanding. *Journal of Research in Science Teaching, 45*(3), 373–394. doi: 10.1002/tea.20238

Tanisli, D. (2013). Preservice primary school mathematics teachers' questioning skills and knowledge of students in terms of pedagogical content knowledge. *Education and Science, 38*(169), 80–95. Retrieved from http://egitimvebilim.ted.org.tr/index.php/EB/article/view/1736

Turkish Council of Higher Education. (2007). *Egitim fakultesi ogretmen yetistirme lisans programlari* [*Teacher education degree programs in Education faculty*]. Ankara, Turkey. Retrieved from http://www.yok.gov.tr/web/guest/icerik/-/journal_content/56_INSTANCE_rEHF8BIsfYRx/10279/49875

Turkish Ministry of Education. (2007). *BT entegrasyonu temel arastirmasi raporu* [*Basic research report on IT integration*]. Ankara, Turkey: Bilgitek Education and Consultancy. Retrieved from http://ocw.metu.edu.tr/pluginfile.php/3298/course/section/1180/BT%20Entegrasyonu.pdf

Vanicek, J. (2015). Programming in Stracth using inquiry-based approach. In A. Brodnik & J. Vahrenhold (Eds.), *Informatics in schools. Curricula, competences, and xompetitions* (pp. 82–93). New York, NY: Springer. doi: 10.1007/978-3-319-25396-1_8

Yukselturk, E. & Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology, 48*(3), 789-801. doi:10.1111/bjet.12453

Yurdugul, H. & Askar, P. (2013). Learning programming, problem solving and gender: a longitudinal study. *Procedia—Social and Behavioral Sciences, 83*, 605–610. doi: 10.1016/j.sbspro.2013.06.115

Zacharia, Z. & Louca, L. (2011). *Investigating how graphical and textual computer-based programming environments support student inquiry in science during modeling*. In The 15th World Multi-Conference on Systemics, Cybernetics and Informatics (pp. 156–159), Orlando, Florida. Retrieved from http://www.iiis.org/CDs2011/CD2011SCI/MMMSE_2011/PapersPdf/NB133FB.pdf