

Research Article

**Cognitive Problems in the Process of Programming Teaching in Higher Education:
Learner-Instructor Experiences¹**

Mithat Elçiçek², Hasan Karal³

Abstract

Programming teaching plays an important role in developing cognitive thinking skills that are considered to be among 21st century skills, such as problem solving, critical thinking, reflective and analytical thinking. However, it is considered to be a difficult and complex process due to its abstract structure. In this context, it is important to identify the factors that affect the process of the programming teaching and to develop solutions for these factors. Cognitive factors are the leading factors among them. In this context, the aim of the research is to examine the experiences of students and instructors regarding the cognitive problems they face during the process of programming teaching. Content analysis method was used to examine the experiences of students and instructors. The participants of the research consisted of 14 undergraduate students and 4 instructors selected through purposive sampling method. Interviews, reflective student diaries, observer reports and semi-structured focus group interviews were used as data collection tools. It is thought that the cognitive problems determined regarding the process of programming teaching will contribute to the studies aimed at developing programming skills.

Keywords: *Higher education, programming teaching, cognitive problems.*

¹ This study was produced from the doctoral thesis prepared by the first author under the supervision of the second author.

² Dr., Siirt University, Faculty of Education, Department of Computer Education and Instructional Technology, mithatelcicek@gmail.com, <https://orcid.org/0000-0003-1845-7271>

³ Prof.Dr., Trabzon University, Faculty of Education, Department of Computer Education and Instructional Technology, karalhasan@gmail.com, <https://orcid.org/0000-0002-3555-050X>

Yükseköğretimde Programlama Öğretimi Sürecinde Yaşanan Bilişsel Problemler: Öğrenen-Öğreten Deneyimleri

Öz

Programlama öğretimi, 21.yy becerileri arasında gösterilen problem çözme, eleştirel düşünme, yansıtıcı ve analitik düşünme gibi bilişsel düşünme becerilerin geliştirilmesinde önemli bir rol oynamaktadır. Ancak, soyut yapısından dolayı zor ve karmaşık bir süreç olarak değerlendirilmektedir. Bu bağlamda, programlama öğretimi sürecine etki eden faktörlerin belirlenmesi ve bu faktörlere yönelik çözüm önerilerinin geliştirilmesi önemlidir. Bu faktörlerin başında bilişsel faktörler öne çıkmaktadır. Bu kapsamda araştırmanın amacı, öğrenci ve öğretim elemanlarının programlama öğretimi sürecinde karşılaştıkları bilişsel problemlere ilişkin deneyimlerinin incelenmesidir. Öğrenci ve öğretim elemanlarının deneyimlerinin incelenmesinde içerik analizi yöntemi kullanılmıştır. Araştırmanın katılımcıları amaca uygun örnekleme yöntemiyle seçilen 14 lisans öğrencisi ve dört öğretim elemanından oluşmaktadır. Veri toplama aracı olarak mülakat, yansıtıcı öğrenci günlükleri, gözlemci raporları ve yarı yapılandırılmış odak grup görüşmeleri kullanılmıştır. Programlama öğretimi sürecine yönelik belirlenen bilişsel problemlerin, programlama becerisinin geliştirilmesine yönelik çalışmalara katkı yapacağı düşünülmektedir.

Anahtar Sözcükler: *Yükseköğretim, programlama öğretimi, bilişsel problemler.*

Introduction

Recently, studies on the programming teaching have gained a momentum in almost all teaching levels, especially with early age groups (Tsai, Wang, & Hsu, 2018; Popescu, 2018). Programming is the process of modelling the required script sequence to real life, necessary for computer systems to operate (Prensky, 2008; Thomas & Greene, 2011; Vee, 2013). Researches have shown that programming teaching is effective in developing important skills of individuals such as reflective, critical and analytical thinking (Lye & Koh, 2014; Gezgin & Adnan, 2016; Smith & Burrow, 2016). Programming teaching has important functions such as developing the imagination of individuals, helping them think in a process-oriented way, internalizing information, preparing product-oriented projects and gaining the habit of collaboration (Cosar, 2013; Gülbahar & Kalelioglu, 2014; Demirer & Nurcan, 2016). In recent years, as in other countries, studies on programming teaching have accelerated in our country (Akpınar & Altun, 2014). With the decision made in 2012 to upskill basic programming skills from an early age in our country, the content of “Information Technologies Course” was updated and software and programming subjects were added to the curriculum. In the "Computer Science Course" program for secondary education, which was published by the Board of Education and Discipline in 2016, new gains were added for the development of "critical thinking", "algorithmic thinking", “creative thinking” and "analytical thinking" skills. In 2018, the Computer Education and Instructional Technologies Undergraduate Program was updated within the scope of "New Teacher Training Undergraduate Programs" in order to ensure that “Information Technology Teacher” candidates have these qualifications. Similarly, in the United States, the United Kingdom, South Korea, and many European Union countries, the importance of programming teaching for the future of the country was realized and regulations were made in relation to programming teaching (Demirer & Nurcan, 2016). In the UK, programming topics were included in the "Information Technology Course" curriculum and regulations were made to enable students to program the applications on their own mobile phones (Burns, 2012). In America, the programming campaign that began with the president's “anyone can learn coding” call was extended as far as the "Coding Olympics" regulation (USA Computing Olympiad, 2015).

Examining the researches on programming teaching, it's seen that the programming skill is considered to be a difficult and complex process, that the attempts to facilitate the programming

teaching make some concepts easier to comprehend, but they are not successful enough in the development of the programming skill (Al-Tahat, Taha, Hasan, & Shawar, 2016; Bosse & Gerosa, 2017; Gulbahar & Kalelioglu, 2018). Therefore, it is important to identify the factors that affect the process of the programming teaching and to develop solutions for them.

In literature review, only a limited number of studies on cognitive problems experienced during the teaching of direct programming are available (Gomes & Mendes, 2014; Ouahbi et al., 2015; Cevahir & Ozdemir, 2017; Bosse & Gerosa, 2017; Sayginer & Tuzun, 2017; Lazarinis et al., 2018). Most of them are studies in which the critical (Solmaz, 2014; Shahmoradi, Nosratinia, & Shangarffam, 2018), reflective, analytical, computational (Lye & Koh, 2014; Kalelioglu, 2015; Wong & Cheung, 2018) and problem solving skills (Gomes & Mendes 2014; Taheri, Sasaki, Chu, & Ngetha, 2016) related to programming teaching have been researched. There are also studies on cognitive problems experienced during the process of programming teaching (Ersoy, Madran, & Gulbahar, 2011; Smith and Burrow, 2016). Cognitive problems identified in researches conducted in various contexts are shown in Table 1.

Table 1

Cognitive Problems Determined Based on Literature

Problems	Resources
Programming logic and abstract thinking	(DuBoulay, 1986; Linn & Clancy, 1992; Thomas, Ratcliffe, & Thomasson, 2004; Lahtinen, Ala-Mutka, & Jarvinen; 2005; Kinnunen & Malmi, 2008; Renumol, Jayaprakash, & Janakiram, 2009; Ersoy, Madran, & Gulbahar, 2011; Ozmen & Altun, 2014; Segmen, 2016; Cevahir & Ozdemir, 2017)
The complex structure of programming	(Bayman & Mayer, 1983; Esteves & Mendes, 2004; Arabacioglu, Bulbul, & Filiz, 2007; Imal & Eser, 2009; Ozoran, Çagiltay & Topalli, 2012; Sirkia, 2012; Sorva, 2012; Mayer, 2013; Altadmri & Brown , 2015)
Pattern building, sequential and cyclic thinking	(Byrne & Lyons, 2001; Esteves & Mendes, 2004; Gomes & Mendes, 2007; Kinnunen & Malmi, 2008; Fesakis & Serafeim, 2009; Imal & Eser, 2009; Ozoran, Cagiltay, & Topalli, 2012; Biju, 2013; Mhashi & Alakeel, 2013; Ozmen & Altun, 2014; Akcay & Coklar, 2016)

When the problems based on the literature are analyzed, it is seen that problems occur in the subjects such as the complex structure of programming, programming logic, abstract thinking, pattern building, sequential and cyclic thinking.

In this context, regulations such as adding software and programming subjects to the "Information Technologies" course content, changing the name of the course "Information Technologies" to "Information Technologies and Software", adding new gains to "Computer Science Course" curriculum and updating the "Computer Education and Instructional Technology" undergraduate program, show the importance given to the process of programming teaching. It also demonstrates the need for permanent solutions in the process of programming teaching. Examining the experiences of information technologies teacher candidates who are the implementer of the above mentioned regulations in programming teaching and instructors about the problems they faced in this process is important. It is thought that identifying the cognitive problems experienced during the process of programming teaching will give important clues to designers and practitioners.

Purpose of the Research

The aim of this research is to examine the experiences of students and instructors regarding the cognitive problems they face during the process of the programming teaching. In this context, answers to the following questions were searched.

1. What are the experiences of the instructors regarding the cognitive problems they face during the process of programming teaching?
2. What are the experiences of higher education students regarding the cognitive problems they face in the process of programming teaching?

Method

Research Model

This study, aiming to determine learner-instructor experiences to cognitive problems during the process of programming teaching in higher education, is designed in accordance with case study method, one of the qualitative research designs. Case study is a qualitative research method in which data is examined in-depth in its nature as a whole and in which state dependent categories are described (Creswell & Plano Clark, 2007; Yıldırım & Şimsek, 2013). Case study primarily aims to investigate and understand the case in-depth (Büyüköztürk et al., 2008). The most prominent feature of this method is that the situation or phenomenon being investigated is

examined within its context due to its specific characteristics (Ersoy, 2016). In this context, the aim of the research is not to make generalizations on the universe, but to understand and explain the defining characteristics of the situation or phenomenon. For this reason, the case study method used is seen as an appropriate method in achieving the purpose of the research. Since the case studies allow a detailed examination of the situations that the researcher cannot control on the basis of “what”, “how” and in why ”questions (Creswell, 2009), in this study, an in-depth study approach was adopted to answer the question “What are the cognitive problems experienced in the programming teaching process? ”.

Participants

The participants of study composed of 14 undergraduate students (F=6, M= 8), enrolling in department of the Computer Education and Instructional Technologies from a state university in Southern East of Turkey during fall semester of 2017-2018 academic year and 4 instructors (F= 1, M= 3) from the same university. It was aimed to ensure maximum diversity in the selection of the participants. Sticking to this objective, 14 volunteer students who previously got programming course from the third grade and 8 volunteer students that were still having programming course at second grade and four instructors who had taught programming course previously were randomly selected. The main reason here was to constitute a small sampling group to allow participants explicitly state their experiences and enable researcher to examine and explain the problem in-depth (Yıldırım & Şimşek, 2013). Therefore, typical sampling method, one of the purposive sampling method, was used in the present study. Purposive typical case sampling allows an in-depth study of one of the many cases in the axis of the research problem. (Marshall, 1996; Büyüköztürk et al., 2017)

Data Collection

Data for the research were collected through interviews, semi-structured focus group interviews, reflective student diaries and observer reports. Observer reports, reflective student diaries and semi-structured focus group interview forms were used to identify the problems that students experienced while learning programming. Interviews were conducted to determine the experiences of the instructors regarding the cognitive problems they faced during the process of programming teaching. During the preparation of the data collection tools, an instructor specialized in programming and an expert in the field of assessment and evaluation were

consulted. In line with the suggestions of the experts, logical and language errors in the forms were corrected. A pilot application was carried out using the forms developed and as a result of this pilot application, some of the items were edited.

Data were collected from students and instructors for eight weeks to determine the cognitive problems experienced during the process of programming teaching. The researcher attended the classes with the instructor every week and observed the course. The reactions of the students to the subjects, their inclinations, participation and changes throughout the process were recorded with the observation forms. Each week, students' learning developments were monitored through reflective student diary forms and their participation in the course was ensured and recorded. In reflective student diaries, students were asked to reflect on the problems they faced and to think about these problems by asking questions about programming teaching. At the end of the eighth week, semi-structured focus group interviews were conducted with the students to get their opinions about the problems they experienced during the process of programming teaching. Also, the instructor who was teaching the programming courses and three instructors who had previously taught them were interviewed.

Data Analysis

In the present study content analysis, method was used to analysis data. The main purpose of content analysis is to reach the concepts and relationships that can explain the collected data (Krippendorff, 2018). In content analysis, coding of data, creating categories, finding themes and defining and interpreting the findings follow each other (White & Marsh, 2006; Çalık & Sözbilir, 2014). Although it embodies a systematic structure, there is not a universal consensus about content analysis. Researchers analyze data through different steps. Therefore, the process in the analysis of gathered data started by converting interview records into textual data. Then three researchers created coding table. In the next step, To determine the similarity rate for the same data set by using the encoding table, the reliability coefficient formula $\Delta = C \div (C + \partial) \times 100$ called consensus by Miles and Huberman (1994) was used (Δ :Reliability coefficient, C: number of terms that have consensus, ∂ :number of terms that have not consensus). At the end of this process, 93% inter-investigator coding agreement was obtained and then this ratio was increased to 100% by discussing non-common codes. After this step, in line with the opinions of field experts and Turkish language experts, codes, categories and themes were finalized.

Findings

In this section, the findings obtained in the research are presented in categories and codes. Excerpts from participant responses supporting the findings are included.

Findings Related to Cognitive Problems Students Experience While Learning Programming

The categories of cognitive problems students face while learning programming and the codes that constitute them are presented below.

Table 2

Cognitive Problems Students Experience While Learning Programming

Theme	Categories	Codes
Cognitive problems	Syntactic errors	Brackets (S7, S10) Quotation (S1, S3, S9) Space (S14)
	Conceptual misconceptions	Idioms (S10) Operators (S3, S5, S9) Variables (S4, S11, S13)
	Mathematical thinking	Operation priority (S8, S12, S13) Deduction (S2, S6, S14) Induction (S8, S12)
	Algorithmic thinking	Algorithm (S3, S7, S9, S10, S13) Cause-effect (S2, S8, S9) Separation into components (S6, S8, S10)
	Sequential and cyclic thinking	While-For-Do (S1, S3) Else if-if (S14) Sequence-increment (S7, S8)
	Pattern recognition and building	Arrays (S4) Relationship order (S8) Switch-case (S11)
	Making logical inference	If-Else (S1, S8, S13) Comparison (S9, S12) If (S2, S5, S7, S10)
	Abstract thinking and generalization	Variables (S3, S11, S12) String-integers (S1, S4, S5, S8, S12)
	Prior learning	General high school (S9, S12, S14) Java-C# complexity (S7)

Examining Table 2, it was determined that students expressed opinions with regards to the following categories about cognitive problems they experienced while learning programming;

syntactic errors, conceptual misconceptions, mathematical thinking, algorithmic thinking, sequential and cyclic thinking, pattern recognition and pattern building, making logical inferences, abstract thinking and generalization and prior learning. Findings by category and sample participant views are presented below.

Six students reported that they had problems due to incomplete or incorrect use of signs such as brackets, quotation marks, spaces. Sample participant views are as follows.

S3: *“Whenever I code something, I always forget the brackets or one of the quotation marks, and when this keeps happening to me I get really tired of it.”*

S14: *“When we write code in C# class, we use too many signs and after a while, I get them mixed up.”*

Regarding conceptual misconceptions seven students stated that there were some misconceptions about the use of key words that are the basic building blocks of programming. Sample participant views are as follows.

S10: *“I don't know exactly what the idioms we use mean, I keep getting confused.”*

P13: *“Some concepts are very similar in function, and I confuse them with each other because I don't know their meaning. Like Int-String.”*

Eight students reported that they had difficulty in basic mathematics, that their knowledge of mathematics was not sufficient and that they had problems in mathematical thinking. Sample participant views are as follows.

S2: *“In the class, we are asked to write a code that finds the sum of the numbers from 1 to 100 in the classroom. I don't even know how to do it normally, so am I supposed to write the code.”*

S8: *“I think one of the biggest difficulties we have is that we have little knowledge of mathematics, I think coding equals mathematical knowledge.”*

11 students stated that they had problems due to the inability to comprehend the real life use of algorithms. Sample participant views are as follows.

S6: *“I have difficulty in understanding the logic between writing algorithm and coding, I cannot relate them to one another. In our regular daily life, for example, it is easy to write the algorithm to come home from school, but it is very difficult for me to create an algorithm to write code.”*

S7: *“For example, let's address the algorithm of finding the largest of the three numbers entered on the keyboard. I can guess how it works in the background here, but I can't understand the logic in the writing of it.”*

Five students reported that they had difficulty with if-else structures, for, do-while cycles, and that they had problems with sequential and cyclic structures that should be formed according to a certain amount of sequence and increment. Sample participant views are as follows.

S1: "Loops are the most challenging subject for me, its logic is very difficult to grasp. The subjects I think I can never learn are if-else, while, for and do."

S14: "Nested conditions are very challenging for me, I think I can solve a lot of examples ranging from simple to complex and grasp the logic of it."

Three students reported that they could not understand the relationship order between objects and concepts in switch case, if-else, if-for structures and that they had problems in recognizing and building patterns. The statement of one of the students is as follows.

S4: "Examples where if and for loops are used together really challenge me because it is not easy to understand their logic. It is extremely complex and difficult, as can be seen in the example of writing a code that will take the third number's power in the amount of the second number and add it to the first number."

Nine students reported that they had problems in making logical inferences about what kind of codes to write in which steps while coding operation steps. Sample participant views are as follows.

S12: "What I find most difficult is that I don't know what type of code I want to write when I encode the operation steps. Because that's where reasoning comes into play and sometimes that's not enough either. To give an example, program codes for paying the salaries of workers in a workplace according to the number of days they work per month are very difficult for me."

S13: "When the teacher gives examples in class and asks us to write their codes, the moments I have the most difficulty is when I can't decide what to do first, and what operations steps I have to do first."

Eight students reported that they had problems in defining and generalizing abstract concepts such as variables. Sample participant views are as follows.

S5: "The subject of variables is the most challenging subject for me this week."

S8: "Programming itself is something abstract, so it is very difficult to learn it through abstract concepts..."

S11: "In fact, we reduce the complexity of the real world by writing program codes, sort of simplifying and classifying everything in our minds, which is not easy..."

Four students reported that they had problems with programming due to previous learning experiences. Sample participant views are as follows.

S7: "While we were in vocational high school, we used Java in programming class and now we use C#, which completely confuses me. Sometimes when the teacher writes a code, I say to myself, "But we didn't do it like that in Java." This makes it completely difficult for me to learn."

S9: "I wish I had little more hours of computer class in high school, then maybe it wouldn't be so hard for me now. My coding knowledge is virtually non-existent, so writing a program is a big problem for me."

Findings Regarding Cognitive Problems Experienced by Instructors in the Process of Programming Teaching

The categories of cognitive problems instructors face while teaching programming and the codes that constitute them are presented below.

Table 3

Cognitive Problems Instructors Experience While Teaching Programming

Theme	Categories	Codes	
Cognitive problems	Syntactic errors	Syntax (L1)	
	Conceptual misconceptions	Idioms (L3)	
	Mathematical thinking		Systematic (L1, L3)
			Deduction (L2)
			Induction (L4)
	Algorithmic thinking		Finding patterns (L3)
			Problem solving (L1) Algorithm (L2, L4)
	Sequential and cyclic thinking	Repetitive structures (L1, L2, L4)	
	Pattern recognition and pattern building	Arrays (L3)	
	Making logical inference		Conditional structures (L2, L3, L4)
		Comparison (L1)	
Abstract thinking and generalization		Variables (L1, L3, L4)	
		Synthesis (L3)	
Prior learning	Inhibitions of prior learning (L4)		

Going over Table 3, it was determined that instructors expressed opinions with regard to the following categories about cognitive problems they had experienced while learning programming; syntactic errors, conceptual misconceptions, mathematical thinking, algorithmic

thinking, sequential and iterative thinking, pattern recognition and pattern building, making logical inferences, abstract thinking and generalization and prior learning. Findings by category and sample participant views are presented below.

It was observed that an instructor who participated in the research stated that there were problems due to incomplete or incorrect use of spelling and punctuation rules. This issue was evaluated under the category of "syntactic errors." The views of the instructor are as follows.

I1: *"I am always surprised by the operators, and I see that students have problems due to incomplete or incorrect use of the spelling and punctuation rules..."*

An instructor reported misconceptions about the use of keywords such as while, for, print, string, int etc. This issue was evaluated under the category of "conceptual misconceptions." The views of the instructor are as follows.

I3: *"I often see that the key concepts such as while, for, printf, string, int, etc. which are perhaps the most basic structures in the programming teaching are confused."*

All the instructors participating in the research stated that there was a problem about the level of readiness for mathematical thinking skills. This issue was evaluated under the category of "mathematical thinking." The views of the instructor are as follows.

I2: *"There are students who have not yet fully learnt the four operations, and it is not easy to teach them the logic of programming. The students I have the hardest time with are those who are inadequate at mathematics."*

I4: *"Students who are poor at mathematical thinking are the students who have the most difficulties when learning programming."*

All the instructors stated that they experienced problems with defining a set of steps listed to perform a task or solve a problem. This issue was evaluated under the category of "algorithmic thinking." The views of the instructor are as follows.

I1: *"The fact that the algorithmic thinking skills of the students are not quite formed makes it difficult for them to learn programming."*

I3: *"I think that even though we do a lot of practice about the algorithms, the students still have problems in this regard. Maybe we should develop new methods for algorithms. Or we should increase the number of courses."*

The three instructors reported that it was not possible to predict how a sequential and cyclic operation would behave in any given cycle or step and therefore the analysis could not be

performed accordingly. This issue was evaluated under the category of "sequential and cyclic thinking." The views of the instructor are as follows.

I1: *"When mathematical knowledge comes into play in the repetitive-repetitive structures, we can say that that's where the real works starts."*

I4: *"Problems arise in sequential structures that need to be constructed according to a certain sequence and increment amount. We can say that students get confused with elements such as if-else structures, for, do-while loops."*

An instructor stated that students had problems regarding the ability to make predictions and estimations for the next step by determining the cause-effect relationship between events, the order of relationship between objects and concepts, the systematic, logic and rules of repetitive structures. This issue was evaluated under the category of "pattern recognition and pattern building." The statement of one of the instructors is as follows.

L3: *"Both the cause-and-effect relationship and writing the systematic program codes that are repeated by establishing connections make it very difficult for the students. Because coding information alone is not enough here, students also need to logically predict the next step."*

All of the instructors stated that, at the end of the mental processes, there were problems with the selection of one of the various alternatives or the execution of the selected statements according to whether the result of a condition statement was true or false. This issue was evaluated under the category of "making logical inferences." The statement of one of the instructors is as follows.

I4: *"Basic if logic structures are comprised of the comparison between two states/variables, but our students are inadequate about what to do if the comparison is true or false"*

All the instructors stated that there were difficulties in solving problems related to real objects from the mind or associating them with the information in human memory. This issue was evaluated under the category of "abstract thinking and generalization." Sample participant views are as follows.

I5: *"We do a lot of practice about various structures in the classroom, as long as we go through the same examples, there is no problem, but when I change the sample situation a little bit, I witness how they can't come up with ideas about how to produce a solution."*

I8: *"Programming is inherently abstract and difficult to understand. In this context, I see that the abstract thinking skills of the students are not sufficiently developed."*

One instructor stated that students had never taken a programming course before, or that prior learnings of those who had previously taken a programming course interfered with their new learning and they had difficulties. This issue was evaluated under the category of "prior learning." The expression of the instructor involved is as follows.

I4: *“The vast majority of students come having never taken any programming course. I always see the students complaining about it. The language in which secondary and high schools students take programming courses and the language in university can often be different. When students perpetuate the mistakes induced by their prior learning, it constitutes a problem.”*

Results, Discussion and Recommendations

When the findings of the research were examined, it was concluded that the experiences of students and instructors regarding the problems experienced during the process of programming teaching were generally similar. It was concluded that there are syntactic problems caused by missing or incorrect use of "brackets", "quotation marks", "comma", "space" or other characters in the process of programming teaching. Problems arising from misuse of key concepts such as "for", "do", "while" or "if" which constitute the basic structure of programming process or from not perceiving their meaning have been identified. Due to the insufficient or incomplete mathematical knowledge of the students, problems have been identified about "order of operations" or "basic four operations." Accordingly, it was concluded that there were problems in the computational thinking skills of the students. It was seen that the students skipped some of the operation steps in the algorithms and could not relate them to the operation steps they used in real life. This showed that a set of steps listed to perform a task or solve a problem could not be identified and caused problems in algorithmic thinking. It was observed that students had problems in conditional structures and loops such as "if-else," "for," "do-while." Therefore, it was determined that there were problems related to sequential and cyclic thinking skills due to the inability of analysis resulting from the unpredictability of the cycle or step in which the sequential and cyclic process would behave. It was seen that the students had problems in the cause-effect relationship between events, the order of relationship between objects and concepts, the systematics, logic of repetitive structures, and the ability to make predictions or estimations for the next step. It has been found that at the end of the mental processes, learners had problems with the selection of one of the various alternatives or the execution of the selected statements according to whether the result of a condition statement was true or false.

This indicated that there were problems related to the ability to make logical inferences. In relation to the problems experienced in abstract thinking and generalization skills, it was found that there were problems in solving the problems related to real objects from the mind or associating them with the information in the human memory. It was concluded that students' readiness from the educational institution they had graduated from (general high school, Anatolian high school, etc.) was insufficient or there were problems arising from prior learning.

In summary, when the research results related to the cognitive problems experienced during the process of programming teaching were examined, problems due to "sequential and cyclic thinking," "pattern recognition and pattern building," "abstract thinking and generalization," "syntactic errors," "conceptual misconceptions," "mathematical thinking" and "prior learning" were found. In this context, it is possible to come across studies that have examined these cognitive problems in different contexts, if not as a whole. In the studies, problems relating to programming logic (Thomas, Ratcliffe, & Thomasson, 2004; Ersoy, Madran, & Gulbahar, 2011; Ozmen & Altun, 2014), abstract thinking (Renumol, Jayaprakash, & Janakiram, 2009; Segmen, 2016; Cevahir & Ozdemir, 2017) and complex programming structure (Imal & Eser, 2009; Mayer, 2013) were generally handled. However, there are also studies in which pattern recognition and pattern building (Gomes & Mendes, 2007; Kinnunen & Malmi, 2008; Biju, 2013) sequential and cyclic thinking (Byrne & Lyons, 2001; Esteves & Mendes, 2004; Gomes & Mendes, 2007) problems are revealed. In the research, it is stated that programming logic is an important threshold and this threshold is directly related to abstract thinking skills (Arabacioglu, Bulbul, & Filiz, 2007; Akcay & Coklar, 2016). The findings obtained from the research data coincide with these results. According to the findings of the research, it was found that students could not understand the order of relationship between objects and concepts while establishing cause and effect relationship between events and had problems in making predictions or estimations in the next steps. Similarly, it was seen that students had difficulty in choosing one of the alternatives that emerged as a result of mental processes and that they could not decide which statement to use according to the result of condition statements. This finding was interpreted as students having problems in logical inference, pattern recognition and pattern building. This finding is supported by the findings of Gomes and Mendes (2007), Kinnunen and Malmi (2008). In the research, it was found that there were problems caused by sequential thinking, cyclic thinking and prior learning during the process of the programming teaching. This was interpreted as the inability of the students to predict which operation would behave in

any given cycle or step and therefore perform analysis accordingly. This result is similar to the results of studies conducted by Byrne and Lyons (2001), Esteves and Mendes (2004).

When the results of the research were evaluated, the cognitive problems experienced in the process of programming teaching were put forward within the framework of the experiences of learners and instructors. The types of solutions that can be developed for these problems can be put forward by future research. It is recommended to conduct studies in this direction.

References

- Akçay, A., & Çoklar, A.N. (2016). Bilişsel becerilerin gelişimine yönelik bir öneri: Programlama eğitimi. *Eğitim Teknolojileri Okumaları*, 121-139.
- Akpınar, Y., & Altun, A. (2014). The need for programming education in information society schools. *Elementary Education Online*, 13(1), 1-4.
- Altadmri, A., & Brown, N.C. (2015, February). 37 million compilations: Investigating novice programming mistakes in large-scale student data. In *Proceedings of the 46th Technical Symposium on Computer Science Education* (pp. 522-527). ACM.
- Al-Tahat, K., Taha, N., Hasan, B., & Shawar, B.A. (2016, July). The impact of a 3d visual tool on female students attitude and performance in computer programming. In *2016 SAI Computing Conference (SAI)* (pp. 864-867). IEEE.
- Altun, A., & Mazman, S.G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin türkçe formunun geçerlilik ve güvenilirlik çalışması. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, 3(2), 297- 308.
- Arabacıoğlu, T., Bülbül, H.İ., & Filiz, A. (2007). Bilgisayar programlama öğretiminde yeni bir yaklaşım. IX. *Akademik Bilişim*. Dumlupınar Üniversitesi, Kütahya, Türkiye.
- Bayman, P., & Mayer, R.E. (1983). A diagnosis of beginning programmers misconceptions of BASIC programming statements. *Communications of the ACM*, 26(9), 677-679.
- Biju, S.M. (2013). Taking advantage of Alice to teach programming concepts. *E-Learning and Digital Media*, 10(1), 22-29.
- Bosse, Y., & Gerosa, M.A. (2017). Why is programming so difficult to learn? Patterns of difficulties related to programming learning mid-stage. *ACM SIGSOFT Software Engineering Notes*, 41(6), 1-6.
- Burns, J. (2012). School ICT to be replaced by computer science programme, *BBC News*. Retrieved November 27, 2018, from <http://www.bbc.co.uk/news/education>.
- Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2017). *Bilimsel araştırma yöntemleri*. Ankara: Pegem Yayınları.

- Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. In S. Fincher (Eds.), *The 6th Annual Conference on Innovation and Technology in Computer Science Education* (191-195). United Kingdom: ITICSE Press.
- Cevahir, H., & Özdemir, M. (2017, Mayıs). Programlama öğretiminde karşılaşılan zorluklara yönelik öğretmen görüşleri ve çözüm önerileri. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu* içinde (s. 304-319). Malatya: İnönü Üniversitesi.
- Coşar, M. (2013). *Problem temelli öğrenme ortamında bilgisayar programlama çalışmalarının akademik başarı, eleştirel düşünme eğilimi ve bilgisayara yönelik tutuma etkileri* (Yayınlanmamış doktora tezi). Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara.
- Creswell, J. W. (2002). *Educational research: Planning, conducting, and evaluating quantitative*. Boston: Pearson Education
- Demir, F. (2015). *Programlama öğretiminde eğitsel programlama dilinin farklı kullanımının programlama başarısı ve kaygısına etkisi* (Yayınlanmamış doktora tezi). Atatürk Üniversitesi, Eğitim Bilimleri Enstitüsü, Erzurum.
- Demirer, V., & Nurcan, S.A.K. (2016). Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
- Ersoy, H., Madran, R.O., & Gülbahar, Y. (2011). Programlama dilleri öğretiminde bir model önerisi: Robot programlama. *XIII. Akademik Bilişim Konferansı'nda sunulan bildiri*, İnönü Üniversitesi, Malatya.
- Esteves, M., & Mendes, A.J. (2004, October). A simulation tool to help learning of object oriented programming basics. In *34th Annual Frontiers in Education, 2004. FIE 2004*. (pp. F4C-7). IEEE.
- Fesakis, G., & Serafeim, K. (2009, July). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. In *ACM SIGCSE Bulletin* (Vol. 41, No. 3, pp. 258-262). ACM.
- Gezgin, D.M., & Adnan, M. (2016). Makine mühendisliği ve ekonometri öğrencilerinin programlamaya ilişkin öz yeterlik algılarının incelenmesi. *Ahi Evran Üniversitesi Kırşehir Eğitim Fakültesi Dergisi*, 17(2), 509-525.
- Gomes, A., & Mendes, A. (2014, October). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1-8). IEEE.

- İmal, N., & Eser, M. (2009). Programlama dili öğrenmedeki zorluklar ve çözüm yaklaşımları. *Elektrik Elektronik Bilgisayar Biyomedikal Mühendislikleri Eğitimi IV. Ulusal Sempozyumu*. [Çevrim-içi: http://www.emo.org.tr/ekler/8bd988bd20804a2_ek.pdf], Erişim Tarihi, 18, 2019.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 1(52), 200-210.
- Kinnunen, P., & Malmi, L. (2008, September). CS minors in a CS1 course. In *Proceedings of the Fourth international Workshop on Computing Education Research* (pp. 79-90). ACM.
- Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. Sage publications.
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H.M. (2005). A study of the difficulties of novice programmers. *Acm Sigcse Bulletin*, 37(3), 14-18.
- Lazarinis, F., Karachristos, C.V., Stavropoulos, E.C., & Verykios, V.S. (2019). A blended learning course for playfully teaching programming concepts to school teachers. *Education and Information Technologies*, 24(2), 1237-1249.
- Linn, M.C., & Clancy, M.J. (1992). The case for case studies of programming problems. *Communications of the ACM*, 35(3), 121-133.
- Lye, S.Y., & 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*, 1(41), 51-61.
- Marshall, M.N. (1996). Sampling for qualitative research. *Family Practice*, 13(6), 522-526.
- Mayer, R.E. (2013). *Teaching and learning computer programming: Multiple research perspectives*. New York: Routledge.
- Mhashi, M.M., & Alakeel, A.L.I.M. (2013, April). Difficulties facing students in learning computer programming skills at Tabuk University. In *Proceedings of the 12th International Conference on Education and Educational Technology (EDU'13), Iwate, Japan* (pp. 15-24).
- Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences*, 1(191), 1479-1482.

- Ozoran, D., Cagiltay, N., & Topalli, D. (2012). Using scratch in introduction to programming course for engineering students. In *2nd International Engineering Education Conference (IEEC2012)* (Vol. 2, pp. 125-132).
- Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: Difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 9-27.
- Özyurt, Ö., & Özyurt, H. (2015). Bilgisayar programcılığı öğrencilerinin programlamaya karşı tutum ve öz-yeterliliklerinin belirlenmesine yönelik bir çalışma. *Eğitimde Kuram ve Uygulama*, 11(1), 51-67.
- Popescu, E. (2018, August). Q2A-I: A Support Platform for Computer Programming Education, Based on Automated Assessment and Peer Learning. In *Advances in Web-Based Learning—ICWL 2018: 17th International Conference, Chiang Mai, Thailand, August 22-24, 2018, Proceedings* (Vol. 11007, p. 3). Springer.
- Renumul, V., Jayaprakash, S., & Janakiram, D. (2009). Classification of cognitive difficulties of students to learn computer programming. *Indian Institute of Technology*, 1(1), 1-12.
- Saygıner, Ş., & Tüzün, H. (2017). İlköğretim Düzeyinde Programlama Eğitimi: Yurt Dışı Ve Yurt İçi Perspektifinden Bir Bakış. *Akademik Bilişim Konferansı*.
- Seğmen, E. (2016). *Programlama öğrenme performansına etki eden bilişsel faktörlerin belirlenmesi ve programlama eğitimi için bir model önerisi* (Yayınlanmamış doktora tezi). Okan Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Shahmoradi, N., Nosratinia, M., & Shangarffam, N. (2018). The relationship among efl teachers' critical thinking, neuro-linguistic programming, and their sense of efficacy. *Journal of Applied Linguistics and Language Research*, 5(2), 41-56.
- Sirkia, T. (2012). *Recognizing programming misconceptions: An analysis of the data collected from the Uhistle program simulation tool* (Unpublished master's thesis). Aalto University, Espoo, Finland.
- Smith, S., & Burrow, L.E. (2016). Programming multimedia stories in scratch to integrate computational thinking and writing with elementary students. *Journal of Mathematics Education*, 9(2), 119-131.
- Solmaz, E. (2014). *Programlama dili öğretiminde alıce yazılımının ders başarısı, eleştirel düşünme ve problem çözme becerileri ile üstbilişsel farkındalık düzeyine etkisi* (Yayınlanmamış doktora tezi). Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara.

- Sorva, J. (2012). *Visual program simulation in introductory programming education* (Unpublished doctoral dissertation). Aalto University, Espoo, Finland
- Taheri, S.M., Sasaki, M., Chu, J.O., & Ngetha, H.T. (2016). A study of teaching problem solving and programming to children by introducing a new programming language. *The International Journal of E-Learning and Educational Technologies in the Digital Media*, 2(1), 31-36.
- Thomas, L., Ratcliffe, M., & Thomasson, B. (2004, March). Scaffolding with object diagrams in first year programming classes: some unexpected results. In *ACM SIGCSE Bulletin* (Vol. 36, No. 1, pp. 250-254). ACM.
- Thomas, M.K., Ge, X., & Greene, B.A. (2011). Fostering 21st century skill development by engaging students in authentic game design projects in a high school computer programming class. *Journal of Educational Computing Research*, 44(4), 391-408.
- Tsai, M.J., Wang, C.Y., & Hsu, P.F. (2018). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research*, 56(8), 1345-1360.
- Vee, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies*, 1(2), 42-64.
- Wong, G.K.W., & Cheung, H.Y. (2018). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, 1(1),1-13.
- Yıldırım, A., & Şimşek, H. (2005). *Sosyal bilimlerde nitel araştırma teknikleri*. Ankara: Seçkin Yayınları.