J. Comput. Educ. (2019) 6(2):277–313
https://doi.org/10.1007/s40692-019-00137-5

Prepare pre-service teachers to teach computer programming skills at K-12 level: experiences in a course

Funda Dağ[1]

Received: 26 June 2018 / Revised: 20 December 2018 / Accepted: 11 March 2019 /
Published online: 21 March 2019

Abstract Today, teaching computer programing (coding) at the K-12 level is one of the priority areas of many countries. On the other hand, teachers with different levels of knowledge about computer programming face with questions related to what to teach and how to teach in a wide range of settings. Considering that the educational programs related to computer programming skills for K-12 students may be increased in the future, during the pre-service training of teachers, development of their professional skills to teach computer programming skills should be supported. In this research, the design of an elective course organized to teach computer programming skills to the pre-service teachers (PSTs) in a degree program that trains computer teachers for K-12 classes in Turkey was presented. In addition, the factors affecting the PSTs' perceptions and their success in that course were investigated. Additionally, PSTs' opinions about the course and the teaching of coding in K-12 classes were examined. According to the findings of the research, it was determined that the PSTs' perceptions related to the course differed according to their general self-efficacy, whereas they did not show difference according to their gender, level of knowledge about computer programming, and their self-efficacy related to coding. It was also found that their success in this course did not differ according to their gender, their achievements in previous computer programming courses and their general academic achievement. Besides, the opinions of the PSTs related to the teaching of coding in K-12 classes and about the elective course were positive; however, their opinions about the computer programming environments differed according to the programming environments they experienced. In this article, based on the findings of the research, discussions, and suggestions for future studies regarding the teaching of computer programming at the K-12 level are presented.

✉ Funda Dağ
fundadag@kocaeli.edu.tr

[1] Department of Computer Education and Instructional Technology, Faculty of Education, Kocaeli University, Kocaeli, Turkey

Keywords Teaching of computer programming · Computer programming environments · Training of pre-service teachers · Self-efficacy

Introduction

In recent years, the field of Computer Science (CS) education at the K-12 level has drawn great interest. There are a number of scientific studies conducted to show that CS education from early ages contributes to the students' critical and analytical thinking and problem-solving skills (Clements and Fullo 1984; Duncan et al. 2014; Fessakis et al. 2013; Gorman and Bourne 1983). Besides the scientific studies, internet resources which are drawing attention especially to the necessity of the teaching of computer programming (coding) skills from the early ages, and highlighting the practical applications (Prensky 2008; programlamacocukoyuncagi 2018; Rushkoff 2012; Tynker 2018), and the recently increased block-based programming tools (code.org 2018; codemonkey 2018; Google Blockly 2018; Öncel 2016) can be expressed as a reflection of the growing interest in CS education for the K-12 classes.

In this direction, the teaching of computer programming have started to take part in the education curricula of K-12 grades in many countries (Akpınar and Altun 2014; Passey 2017). In Turkey, software and computer programing issues have been introduced within the framework of "Information Technology and Software" course at the 5th grade level and afterwards in primary and secondary schools since 2012 (Demirer and Nurcan 2016). Algorithm, coding, and software units have been included in the curricula over time and the concept of computational thinking was included in the curriculum proposed in 2017 (Mercimek and İlic 2017). According to the European Schoolnet, coding in both formal and informal learning environments continues to be a worldwide trend and major European countries, such as France and Spain, have just introduced it in their school curricula (Balanskat and Engelhardt 2015). The concept of "computing" defined as how computers and computer systems work and how they are designed and programmed was also included in the curricula of the UK national primary and secondary level schools in 2014 (Berry 2013).

In order to ensure the students to have the computer programming skills, there are different methods and tools that can be applied at different grade levels. However, although the studies in this field are still at the development stage, it is indicated that there are some limitations regarding the lack of training of teachers in using the computer programming tools, the lack of equipment in the countries with fewer resources and the lack of sources and tools that teachers can use in their instructions (Lockwood and Mooney 2017). In this study, an elective course designed for pre-service teachers in order to improve their knowledge and skills related to the teaching of computer programming in the K-12 classes is presented and the effects of various factors on perceptions and academic achievement of the pre-service teachers regarding with the course is investigated. In addition, their opinions about the course and the teaching of the coding at the K-12 classes are examined. The findings of this study are expected to be guide for similar courses and training programs that will be offered to address the deficiencies in the teaching of computer programming at K-12 grades. In addition, it is thought that the findings of this study will contribute to the literature in terms of the needs of the scientific studies regarding the teaching of computer programming and programming tools.

Teaching of computer programming and teacher characteristics

In relation to the teacher training policies of countries in this field, teachers, who teach information technology or newly computer programming courses in primary and secondary education institutions, show differences in terms of teaching pedagogical information and pedagogical knowledge related to the teaching of computer programming (Diethelm et al. 2012; Dursun 2013). In Turkey, at the Computer and Instructional Technology Departments of Education faculties where computer teachers are trained for K-12 students, although the pre-service computer teachers take some courses about programming languages and skills, they don't take any courses about how to teach computer programming and which tools can be used at K-12 classes (YÖK 2018a, b).

Based on the feedback of the participants in a workshop about computer programming tools and methods for pre-service computer teachers, Yükseltürk and Altıok (2015, 2016) stated that pre-service teachers were not sufficiently knowledgeable about the visual tools and current methods related to the computer programming although they saw themselves adequate to teach computer programing at the level of K-12. In addition, researchers have indicated that some elective courses that can be added to undergraduate curricula of the teachers in order to improve themselves in teaching of computer programming and that include some practical training activities, various community and club activities, and some alternative methods and tools would contribute to the development of pre-service teachers.

In their qualitative research, in which they examined the successful teaching methods used in coding instruction at primary and secondary schools and difficulties encountered by 339 computer teachers in England, Sentence and Csizmadia (2017) stated that the terminology and methods used in CS education should be different from the teaching techniques used in the basic ICT courses and that both the teachers and the students should adapt to the new teaching methods and different content types. The researchers emphasized that some of the difficulties teachers encounter in teaching of computer programming originated in the internal and external reasons related to the teachers, while others originated in students. They stated that the internal difficulties of the teachers were the lack of self-confidence related to CS issues and their inadequate knowledge in determining the appropriate pedagogy, whereas their external difficulties were the lack of the instructional sources. On the other side, the student-related difficulties were related to the fact that the students had difficulties in understanding the content and problem-solving. In the light of the findings of their research, they said that teachers needed to get more training in how to teach computer programming at K-12 classes, more instructional sources needed to be produced to deal with the internal and external difficulties and that teachers can improve their pedagogical skills through this way (Sentance and Csizmadia 2017).

On the other hand, teachers' perceptions, attitudes, and self-efficacy related to teaching areas are also of great importance in order to increase their field knowledge and to improve their pedagogical competence (Akdağ 2016; Bray-Clark and Bates 2003; Demirtaş et al. 2011; Durmuşoğlu et al. 2009; Şahin and Şahin 2017). Finson et al. (2000) noted that teachers' own perceptions and their role in the teaching environment may have been partly derived from self-efficacy. In their research, Rubeck and Enochs (1991) stated that the self-efficacy of the teachers with a strong knowledge of the field might be higher than those with a weaker knowledge of the field, that these teachers were prone to create a student-centered learning environment and that they had more information about the level of the development of their students (as cited in Finson et al. 2000). A strong self-efficacy is a feature that contributes to the individual in terms of personal development and diversification of skills (Yıldırım and İlhan 2010). The concept of self-efficacy, which is defined as the belief that an individual can achieve and perform a certain performance (Bandura 1994), is related to the degree to which one relies on his own resources, not his abilities. Although it is stated that the self-efficacy belief specific to a particular area is a predictor (as cited in Yıldırım and İlhan 2010), the concept of general self-efficacy also come into prominence in many areas (Yıldırım and İlhan 2010). In his study in which he found that teachers' self-efficacy beliefs were related to the use of technology in teaching, Albion (1999) emphasized that as the expectations of the community for the integration of information technology into daily teaching practices increased, it would be more important for all teachers to have prepared themselves professionally enough. He also emphasized that teachers were directly connected with the applications of self-efficacy beliefs about using technology for teaching.

Hiltunen (2016) emphasized that the main challenge in the teaching of computer programming skills stems from the fact that the teachers do not know how to obtain adequate training and technical know-how in programming and how to reflect these acquisitions in educational environments. Diethelm et al. (2012) emphasized that computer science teachers use different ways to design lessons on a particular topic, they do not have a clear idea of students' perspectives and they often feel a lack of knowledge on CS field. Researchers have addressed the need to take into account the individual abilities of teachers and their perspectives in a particular context when teaching CS field. In their study, in which they examined the pre-service computer teachers' attitudes towards computer programming, Erol and Kurt (2017) stated that although the pre-service teachers had a positive attitudes towards computer programming, they expressed some difficulties related to the understanding of the rationale of algorithm, coding and implementation skills. Korkmaz and Altun (2013) examined the attitudes of the students of CS teaching and engineering department towards the learning of computer programming. Their findings showed that though students' perceptions about the importance of the learning of computer programming and their attitudes towards the learning of programming were positive, their willingness to learn these skills was lower. Based on this information, in Turkey and in the world and when the current curriculum emphasizing the importance of computer programming skills is considered, it can be said that self-efficacy and attitudes of the computer teachers who will design and teach various courses regarding computer programming skills may be influential on their professional and personal development especially in terms of adapting themselves to the current developments and practices in the field of CS education at K-12 level.

The undergraduate courses students take during their education in different undergraduate programs increase their self-efficacy beliefs (Transfer. Çapri and Çelikkaleli 2008). According to the findings of his doctorate study, in which he evaluated computer teachers' special field competencies according to the opinions of the university lecturers, pre-service teachers, and teachers, Dursun (2013) stated that the education that teachers had at the university was inadequate. He also highlighted that teachers could not use the training they had received and that they could only gain some proficiency by experience. In his study in which the special field efficacy of the pre-service computer teachers was examined, Akdağ (2016) stated that the studies related to the pre-service computer teachers' knowledge and skill proficiency they gained in their pre-service education were limited. In their research, in which they evaluated the proposal for updating the Information Technologies and Software Course Curriculum, Mercimek and Ilic (2017) pointed out that in-service training and professional development programs had become necessary in order to ensure the teachers' competencies in the field of computer programming in K-12 education so that the proposed program could correspond to the practice. They also emphasized that the curriculum of undergraduate computer teacher programs did not include a direct course for the acquisition of this skill and that this skill was not mentioned in the program outputs. In this respect, teachers should take in-service trainings on the current computer programming teaching tools in order to follow up the developments in this area. Also, pre-service teachers should take courses to improve their professional knowledge and skills about the teaching computer programming skills, and tools used in this field. This may provide a professional contribution to the development of both their field knowledge and pedagogical knowledge.

Computer programming tools for teaching coding

The main aim of the teaching computer programming skills from the early ages is to make the individuals, who constitute the information society in the 21st century, gain the ability of computational thinking (Demir and Seferoğlu 2017). Iste (2011) states that the computational thinking covers creative thinking, algorithmic thinking, critical thinking, problem-solving, cooperative learning, and communication skills, and that it is a problem-solving approach reinforcing the convergence of technology and thinking that can be regarded as an outgrowth. It is emphasized that within this framework the purpose of teaching computational thinking skills to students is not to advance students in the field of computer science but to make them acquire the habit of practicing computational thinking skills in different courses in their lives and education processes (Iste 2011). In many countries, elective or compulsory courses are added to the curriculum of primary and secondary schools or modules are applied in courses in order to teach the students the computer programming skills offered as one of the approaches of the teaching of computational thinking (as cited in Kukul et al. 2017; Lockwood and Mooney 2017). The number of tools developed for the teaching of computer programming skills at different teaching levels is quite high. As a result of their systematic literature review based on research focusing on acquisition of computational thinking skills, Lockwood and Mooney (2017) stated that 50 different tools were used, developed, or integrated into training activities.

In their research in which they presented a perspective of what, how and why to teach as computational thinking skills, Shailaja and Sridaran (2015) suggested that at the level of 3rd–5th grades, block-based tools such as Scratch can be used to develop algorithmic thinking and writing skills; and code-based programming tools such as Basic, Visual Basic can help 6th–8th grade level students to learn what computer programming is by creating basic programs. They also said that computer programming languages such as Phyton, Java can be taught at 9th–12th grade levels. In his master thesis exploring the importance and role of teaching and learning of primary school-level computer programming skills in Finland, Hiltunen (2016) emphasized that teaching computer programming should be started with logical "real life" games and digital game-based learning tools, and the syntax/code-based programming tools should be used after the activities done first with drag & drop tools.

In Turkey, in the current 5th and 6th-grade Information Technology and Software Course program, it is expressed that tools such as open source or free accessible software can be used to teach the concepts of computer programming (MEB 2018). In the 1st–8th grades Information Technologies and Software Course draft program offered as a suggestion in 2017, for 6th-grade level, Scratch which is a block-based programming tools, and for 8th-grades, Small Basic which is code-based programming tool are recommended (MEB 2017). For the high school level, in the basic learning objectives of the computer science elective course curriculum, it is expressed that students should use at least one of the computer programming languages at a good level and gain experience in robot programming, mobile programming and web programming (MEB 2016).

While designing the course that is the subject of this research, some determinations about the computer programming tools to be included in the curriculum have been made by taking into account suggestions related to the computer programming tools in the literature mentioned above, and the experiences gained in the previous computer programming courses. Accordingly, while selecting the computer programming tools, the ones which would facilitate the adaptation of pre-service teachers to different tools and could be used for the teaching of computer programming skills at different levels were included in the course content.

Within this context, three basic categories were identified as block-based programming tools, code-based programming tools and object-based (3D animation based) programming tools. In order to determine a computer programming tool under each category, priority has been given to computer programming tool that are recommended for certain age levels in the literature (Hiltunen 2016; Shailaja and Sridaran 2015; Vicki 2018; MEB 2016, 2017, 2018) and the curriculum for the K-12 classes in Turkey. Besides, taking into account the need of teachers for more teaching/learning resources (Lockwood and Mooney 2017), priority has been given to computer programming tools that are accessible to Turkish learning and teaching sources. In this respect, the computer programming tools included in the course content are as follows;

Scratch 2 This tool is a project of the Lifelong Kindergarten group in the MIT Media Lab (Scratch 2018). It is a block-based coding tool that is basically capable of 2D design that can be used with drag-and-drop technique. Scratch is used in teaching of basic computer programming concepts/skills (Duncan and Bell 2015), game design (Boechler et al. 2014), and algorithm building (Burgett et al. 2015; Nesiba et al. 2015). It is a free tool that can be run through the web or installing on the computer. In addition, Scratch provides opportunity to access a lot of learning and teaching resources in different languages. Moreover, a Turkish education portal related to the Scratch is presented to everybody by the Education Information Network of the Ministry of Education in Turkey.

Small Basic Presenting an easy development tool for understanding and implementing basic computer programming concepts, Microsoft Small Basic is a tool that directs both children and adults to the computer programming world (Microsoft 2017a, b). Compared to other computer programming languages such as C, Java, and Phyton, it has much easier syntax. With Small Basic, it is possible to create programs both graphically and on console screen. Also, it offers a compiler feature that provides error analysis in the written programs. With these features, it provides an environment that allows basic entry into the computer programming before proceeding to more advanced programming languages. IntelliSense feature, which allows users to use language-related code expressions without having to memorize, and the Turtle plug-in, which provides opportunity to create two-dimensional programs on a graphical display, are seen as remarkable features of it. In an action research subjecting to a master thesis and conducted with fourth and fifth-grade primary school students and their teachers, Akçay (2009) stated that Small Basic was accepted by teachers and students as a useful and easy-to-use technology. Besides, he emphasized that teachers and students had found it advantageous to use Small Basic as a new technology in learning environments and that it had positive effects on the motivation of students. The Handbook of the Small Basic programming tool, which is available in several languages including the Turkish, provides information on the use of the tool, as well as being guidance for teachers in lessons they will design with some Project examples.

Alice Cooper et al. (2000) described Alice, which is the three-dimensional interactive animation tool based on the Java language, as a computer programming tool that helps students “combine pieces” to understand how an object-oriented program is executed. Researchers have stated that Alice can help a student to understand which task is expected to perform by each piece in a program and how the pieces work together to solve a problem. In Alice, which use event-based and object-oriented programming paradigm, users can add objects to a scene and manipulate the attributes of objects by using an object library containing three-dimensional objects, and sounds, and they can provide animations of objects by using events (Price and Barnes 2015). Şendağ and Erol (2012) examined Alice software in terms of usability as a cognitive tool and evaluated the opinions and projects of primary school students who use the software. They pointed out that Alice could be used as a design and problem-solving tool in the second level of elementary schools. In their research conducted with girls at the secondary school level, Kelleher et al. (2007) pointed out that, with its character-driven and easy-to-animate features, Alice can be adapted to different purposes, such as digital storytelling. Various learning and teaching sources related to Alice programming tool can be found on the website of Alice. Turkish video training materials can also be reached through the Education Information Network of Turkey.

It is thought that the pre-service teachers who develop their knowledge and skills in these computer programming tools can easily adapt to future programming tools with similar characteristics. In a semi-experimental study comparing block-based and text-based programming tools in a high school computer programming course, Weintrop and Wilensky (2017) found that students who had experiences related to the programming concepts in a block-based or text-based environment can easily transfer information to a different computer programming environment with similar characteristics and that students who carried out coding in a text-based environment showed similar computer programming skills as professional programmers did. According to this, it can be considered that a pre-service teacher who has gained experience in animation, scene, character, and object-based programming concepts in Alice can learn and teach a different programming tool such as Kodu, which is presented as a visual programming tool for three-dimensional game and digital story design, by using his/her experiences. However, for those who learn to write code both graphically and on console screen of Small Basic, it may be easier to switch to a more advanced computer programming language such as C #, Java, or Phyton, where the syntax can be more complicated.

How to teach computer programming skills with computer programming tools?

Computer Programming tools whose main purpose is to visualize the programming are tools that target mostly the young people with no computer programming experience, reflect the syntax and structural characteristics of existing programming languages and allow computer programming concepts to be adapted to different cultures in a multimedia context. In these tools, by preparing digital stories, animated computer games and various multimedia applications in different fields such as art, science, mathematics, and music, teaching the coding to the students may contribute to students' learning by fun (Price and Barnes 2015). In this direction, in order to teach these tools, a constructive approach based on problem-solving learning that promotes information processing practices and perspectives depending on information processing, guidance, and reflective activities should be created (Lye and Koh 2014). On the other hand, although computer teachers already have more knowledge and experience in using computers than other field teachers, they still need training in order to integrate the technological tools into their teaching. For this, it is necessary to develop case studies and daily life examples through which students can learn and teachers can develop their teaching (Doukakis et al. 2010).

More research is needed to determine the supports that teachers need and to explore the instructional strategies to support different students' computational thinking skills (Israel et al. 2015). For this reason, more evaluation and evidence-based research to support making progress on computer programming skills or computational thinking skills should be carried out. (Fincher 1999). In this respect, the course program, which is the subject of this research, has been presented in detail and the perception, achievement, and opinions of the students about the course have been evaluated in terms of various factors. During the implementation of the concerned elective course, all the teaching activities, project work, and homework tools were planned according to the mentioned characteristics and studies were carried out for the pre-service teachers in order to produce solutions by using various problems and examples. The course outline, the works done according to this outline, and the evaluation criteria have been presented in detail. Because the purpose of the course was to improve pre-service teachers' knowledge and skills in teaching with the computer programming tools, teaching the usage features by means of demonstration of tools was not given importance. Instead, in order to create a learning environment in which students can participate in the learning process at the highest level, student-centered teaching methods such as exploratory learning and project-based learning have been emphasized. The purpose of using these methods is not only to develop knowledge about computer programming tools but also to enable them to develop their skills to use these tools, as well as to make inquiries about how and what to teach with them.

Purpose of the research

In this research, an elective course organized for pre-service computer teachers in order to improve their professions in teaching computer programming at the K-12 level is presented and the impacts of several variables (*gender, perceived programming knowledge levels, programming self-efficacy and general self-efficacy, previous programming course performances, and general academic success levels*) on pre-service computer teachers' perceptions and their academic achievement in the course are examined. In addition, the opinions of pre-service teachers about the course and teaching the computer programming at K-12 classes have also been examined within the scope of this research. Accordingly, the research questions of this study are:

1. What are the perceptions of pre-service teachers about the course, their programming self-efficacy, and general self-efficacy?
2. Do the perceptions of the pre-service teachers about the course differ according to their gender, perceived level of programming knowledge, self-efficacy perceptions related to computer programming, and general self-efficacy perceptions?
3. Do the academic achievements of the pre-service teachers regarding the course differ according to their gender, their general academic achievements, and the level of success in their previous computer programming courses?
4. What are the opinions of the pre-service teachers about the elective course and the field of computer programming education at K-12 classes after the course?

Method

Research design

The research is a descriptive research in which survey model is applied. In the descriptive research that constitutes the first stage of scientific studies, the purpose is to better understand group, and to determine the relations of the existing situation or object by the analysis of the data gathered through various data collection methods in order to understand and explain the objects and situations (Creswell 2016, p. 155; Kaptan 2011, p. 59). Since it has been conducted to determine the design of a course and how the course content is perceived and interpreted by the target group, this research is a descriptive survey research. Whether there is any effect of the various variables on the academic success and perceptions of the participants concerning the course has been examined in this study. In the study, qualitative data were collected in order to support quantitative data and to reveal the opinions of pre-service teachers on the course and computer programming education at K-12 classes. As a result, it is a research study that reveals an overall statement about the perceptions of pre-service teachers in terms of both the applied course and computer programming teaching at the K-12 grades and that discusses the inferences related to the implications of the computer programming teaching at K-12 grades according to the study results.

Participants

The participants of the study consisted of 26 pre-service teachers who were student in their final year during fall semester of 2017–2018 at the Department of Computer Education of the Faculty of Education at a state university in Turkey and who took this course as an elective course. Table 1 shows the participants' gender, perceived level of computer programming knowledge, academic achievement level of this course, previous level of computer programming achievement (*achievement average of Programming Language 1 and 2 courses)*, and general academic achievement level.

According to Table 1, 10 of the students (38.5%) who took the course were female and 16 of them (61.5%) were male. Most of the participants expressed their programming knowledge level as "*intermediate*" (20, 76.9%). It was calculated that the course achievement rate of the study group was 74.19 (Sd: 9.69) and the success (final) grades ranged between 55 and 87. According to Table 1, it is seen that the average academic achievement of the students on the course is intermediate (15, 57.7%); whereas 9 (34.6%) students showed high success, 2 (7.7%) of them showed low success.

In order to determine the previous computer programming knowledge levels and general academic achievement levels of the students, the achievement grades of the students in the Programming Languages 1 (PL1) Course, in which the concepts of algorithmic and basic programming are taught, the achievement grades

Table 1 Demographic characteristics of the participants

	Gender		Total
	Female	Male	
Prog. knowledge level (perceived)			
Low	1 (3.8%)	0 (0%)	1 (3.8%)
Intermediate	7 (26.9%)	13 (50%)	20 (76.9%)
High	2 (7.7%)	3 (11.5%)	5 (19.2%)
Total	10 (38.5%)	16 (61.5%)	26 (100%)
Course aca. achievement			
Low	0 (0%)	2 (7.7%)	2 (7.7%)
Intermediate	4 (15.4%)	11 (42.3%)	15 (57%)
High	6 (23.1%)	3 (11.5%)	9 (34.6%)
Total	10 (38.5%)	16 (61.5%)	26 (100%)
Comp. lang. 1 aca. achievement			
Low	2 (7.7%)	6 (23.1%)	8 (30.8%)
Intermediate	6 (23.1%)	6 (23.1%)	12 (46.2%)
High	2 (7.7%)	4 (15.4%)	6 (23.1%)
Total	10 (38.5%)	16 (61.5%)	26 (100%)
Comp. lang. 2 aca. achievement			
Low	4 (15.4%)	5 (19.2%)	9 (34.6%)
Intermediate	6 (23.1%)	10 (38.5%)	16 (61.5%)
High	0 (0%)	1 (3.8%)	1 (3.8%)
Total	10 (38.5%)	16 (61.5%)	26 (100%)
General aca. achievement			
Low	0 (0%)	2 (7.7%)	2 (7.7%)
Intermediate	7 (26.9%)	10 (38.5%)	17 (65.4%)
High	3 (11.5%)	4 (15.4%)	7 (26.9%)
Total	10 (38.5%)	16 (61.5%)	26 (100%)

of students in the Programming Languages 2 (PL2) Course, in which object-based and visual programming concepts are taught, and the academic achievement averages belonging to the past seven semesters are presented in Table 1.

According to this, it was calculated that the students had an average score of 66.08 (Sd: 14.63) and that their grades were between the 41–89 range in PL1 Course. The students' average achievement score was 62.54 (Sd: 19.92) and their grades were between the 40–80 range in PL2 Course. Based on the data in Table 1, it can be said that the students' achievements in the basic programming course (PL1) are somewhat better than the achievements in the advanced programming (PL2) course. In addition, students' overall academic achievement is at the intermediate level.

Course design and implementation process

The course was implemented as a 15-week process. In this process, discussion about the activities in how and why computer programming instruction important for K-12 (*for 2 week*), study on Scratch 2 (*for 3 weeks*), study on Small Basic (*for 3 weeks*), midterm exam (*1 week*), study with Alice 3 (*for 3 weeks*), and digital story design study with Alice 3 (*for 3 weeks*) were carried out. For the evaluation of the course, the practice grade given on the basis of the arithmetic average of the five assignments done in the first 8 weeks and the arithmetic average of the midterm exam conducted in the form of the written exam were used as the assessment grade for the year. Final assessment of the students in the course was carried out through the digital story design and presentation in Alice 3. Peer evaluation and instructor assessment were used together in the final evaluation based on the evaluation rubric. The Moodle learning management system was used to share all content, documents, information and materials related to the course, and for students to present their homework and to transfer the evaluations. Students were enrolled at Moodle LMS at the beginning of the course and it was made possible for the instructor and the students to be in contact with each other outside of the class hours. In Fig. 1, outline of the elective course is presented.

As indicated in the course outline in Fig. 1, mainly student-centered teaching methods were applied in the elective course. A study was conducted on the means of grasping the use of tools by demonstrating each tool for 1 week during the total period of 12 weeks in which the studies on computer programming tools were carried out. In classroom applications related to computer programming tools, about three or four problems were studied during the 3-h face-to-face session of the course each week. After the problems were presented to the students, they tried to create their own solutions and tried to determine the best solution by comparing them according to the criteria such as accuracy and convenience level in presentation of solution in a class environment. In this process, the instructor undertook a role helping and guiding them in generating answers to their questions without dictating his/her own solution. During the semester, five assignments were given to the students. The assignments had been prepared with the evaluation rubrics and in a more complex way than the class discussions and practice questions. The rubrics used in the evaluation of the assignments and the final project were shared with the students in each study. In this respect, it was aimed to create examples for the pre-service teachers' evaluations that they will do when they apply these instruments in their lessons. For the presentation of the project proposals of the final exam, an average of 10 min was allocated for each student. For this reason, the final exam with 26 participants was completed in approximately five and a half hours.

Data collection tools

In this study "Self-efficacy Perception Scale Related to Programming", which had been adapted to Turkish by Altun and Mazman (2012), was used in order to

Week	Content	Method and Application	Resources and Evaluation
1	Computational Thinking and its components	Direct instruction Discussion	Field lietature about CS education and computer programming teaching on K-12 grades
2	Recent research in field literature	Discussion	**HomeWork-1**: Prepare research report regarding below concepts *Computational learning?, ... ? Indicators of computational thinking? Relation between computational thinking and programming .... how to teach programming at which age groups and by which tools?, programming teaching research related with k12 level in Turkey and in the world.*
3	Scracth 2	Demonstration and investigation (Interface, panels, learning/teaching resources)	eba.gov.tr, coding resources for kids and youngs (in Turkish)
4	Scratch 2	Learning by discovery (Implement basic computer programming concepts with blocks)	**Homework-2:** Animation design by using loop and decision structures (*If the crab catches the fish*)- **Evaluation Rubric**
5	Scratch 2	Learning by discovery (Interactive applications)	**Homework-3:** Interactive game design by using loop, desicion and variables (*four-mode operation game*) – **Evaluation of the project with Dr.Scratch Tool**
6	Small Basic	Demonstration and investigation (Computer programming environment, basic operations, learning/teaching resources)	eba.gov.tr, programmingcoding resources for kids and youngs (in Turkish)
7	Small Basic	Learning by discovery (Coding in Text Window)	**Homework-4:** Writing text window program by using basic computer programming concepts and subprogram (Menu *program*) - **Evaluation Rubric**
8	Small Basic	Learning by discovery (Coding in Graphic Window)	**Homework-5:** Interactive game design for given storyboard (*Ball capture game*) – **Evaluation Rubric**
9	**Midterm Exam**	Midterm Exam	It's related to first eight weeks.
10	Alice 3	Demonstration and investigation (3D Computer programming environment, scene and object properties, basic operations, using of method and procedures, learning/teaching resources)	eba.gov.tr, coding resources for kids and youngs (in Turkish)
11	Alice 3	Learning by discovery Digital story design (Stage 1)	eba.gov.tr, coding resources for kids and youngs (in Turkish)
12	Alice 3	Case study Digital story design (Stage 1)	eba.gov.tr, coding resources for kids and youngs (in Turkish)
13	Alice 3	Learning by Design (Students identify original project themes and prepare storyboards.)	Specifying animation type and theme, creating storyboards and information about evaluation criteria of the final project (*introduction of peer review and expert evaluation rubrics*)
14	Alice 3	Project-based learning (digital story design)	Implementing storyboards in the Alice 3 environment
15	Alice 3	Project-based learning (digital story design)	Implementing storyboards in the Alice 3 environment
16	**Final**	Final Project presentation	Presentation of digital story **Final Evaluation - Peer review and staff based evaluation**

Fig. 1 Course syllabus

determine the computer programming self-efficacy perceptions of the students. The internal consistency coefficient of the scale which was composed of nine items and two factors (self-efficacy perceptions of simple programming tasks and complex programming tasks) was calculated as 0.928. It was stated that the sum of the nine items explained the 80.814% of the total variance and that this model was verified by confirmatory factor analysis. In this respect, the scale expressed as a valid, reliable instrument measuring self-efficacy perception of computer programming is a 7-point Likert type scale. The maximum score that can be taken from the scale is 63, and the minimum score is 9. The internal consistency coefficient of the programming self-efficacy scale in this study was found as 0.91.

In this study, "General Self-Efficacy Scale", which had been adopted to Turkish language by Yıldırım and İlhan (2010) and which can be used as an evaluation tool in each field to determine the general self-efficacy of the participants, was used. This scale, whose validity and reliability studies were done, is presented as a one-factor including 17 items with five-point Likert scale. The highest score that can be taken from the scale is 85 and the lowest score is 17. The increase in score indicates the increase in the self-efficacy belief. The correlations between the scale and other different scales were statistically significant and the internal consistency coefficient (Cronbach alpha) of the scale was 0.80. In this study, the internal consistency coefficient of the scale was calculated as 0.85.

In this study, the “Course Evaluation Questionnaire” prepared by the researcher was used to determine the perceptions of the pre-service teachers in terms of both field knowledge and pedagogical aspects of the course. In accordance with the research related to the course evaluation criteria in the literature (Kember and Leung 2008, p. 344; Ramsden 1991) and by taking example the forms used in similar applications (Yükseltürk and Altıok 2015), an eight-item questionnaire has been developed in accordance with the scope of this course and the purpose of the research. Related to the questionnaire, opinions of a specialist in CS teaching field were taken and some expressions were corrected. In the seven-point Likert type structure, the lowest score that can be taken from the questionnaire was 8 and the highest score was 56. The internal consistency of the items of the questionnaire for this study was calculated as 0.93.

After the course success grades of the students were determined, the general academic achievement scores and the achievement scores of previous programming courses (PL1 and PL2) were taken from the information system of the university and used for the purpose of the research. The students who participated in this research had taken the PL1 and PL2 courses from the lecturer who gave this elective course. As shown in Table 1, all the success grades are presented in three categories as “high”, “intermediate”, and “low”.

Lastly, a semi-structured electronic form was developed by the researcher and in this form, five questions were specified to probe the opinions of the pre-service teachers on the three central topics (i.e., *computer programming instruction and personal development efforts, feelings about the course, feelings about programming tools*). Questions about the first topic are (*What do you think about the teaching of programming skills at the K-12 grades?*) and (*If you try to develop yourself in the teaching of programming at the K-12 grades, could you please explain what you are doing?*). Questions about the second topic are (*What do you think about the course that you took? Did it contribute to you? Could you please explain?),* and (*Did the information taught in this course contribute to you having sufficient knowledge about current methods and tools that can be used for teaching computer programming?*). Regarding the third topic, (*Were the programming tools (Scratch2, Small Basic, Alice 3) you learned in this course sufficient for you to teach programming to K-12 students?*) was asked.

Data collection and analysis

The data collection process of the research was carried out within the course process. General self-efficacy scale, self-efficacy scales for programming, and personal information form for the demographic characteristics of the participants were collected during the 4th week. The course evaluation questionnaire was conducted and collected after the final project exam. All of these forms were given to the students in print and an average time of 20 min was allocated for each of them to respond. The code names that are specified by the students on the forms were used in order to match the forms.

The data obtained by the semi-structured form related to students' opinions on the course were collected in the electronic environment after the beginning of the spring semester. The purpose of this was that pre-service teachers had begun teaching practice in schools in the spring semester, and they were more likely to be able to define their opinions on the course more clearly because they had the opportunity to apply the information they had learned in the elective course. Besides, it was thought that because they did not take any course from the lecturer of the elective course, they could share their opinions more objectively. On the other hand, since the form of opinion for the course had been presented on the web quite a long time after the end of the course (about 7 weeks later), the number of participants who responded the semi-structured interview form was about one-third of the pre-service teachers that participated in the study. Only 8 of the 26 pre-service teachers filled out the semi-structured interview form.

The range coefficients were calculated by the ratio of the difference between the highest and lowest points that can be taken from the scales used in the research to the number of groups of the used scales/questionnaires (Tekin 1996). The range coefficients used in the re-interpretation of the item scores and the total scores obtained from the scales/questionnaires are presented in Table 2. The descriptive statistics based on the recalculation of the total scores of the participants from the surveys according to the range coefficients are presented with minimum and maximum values, mean value, and standard deviations. In addition to this, determining the academic achievement levels of the students, the grade table consisting of 9 categories was divided into three groups by considering the success grade system of students' university. Accordingly, grades of 80 and above were accepted as "high", 58–79 range as "intermediate", and grades below 57 as "low".

Apart from the basic descriptive statistics, non-parametric test methods were used in the analysis of differences in the context of research questions of the research. Mann–Whitney-*U* and Kruskal–Wallis-H tests were used to test differentiation of pre-service teachers' perceptions about the course according to gender, course success grade, perceived level of computer programming knowledge, self-efficacy perceptions of programming, and general self-efficacy perceptions. Mann–Whitney-*U* Test can be used in the case of the distribution of the scores of two unrelated samples that do not meet the normality assumption; the Kruskal–Wallis-H test can be used to test whether the mean of two or more samples differs significantly from each other. In cases where the difference was significant according to the Kruskal–Wallis-H test result, comparisons between groups were made using the Mann–Whitney-*U* test. In all statistical analyses, the results were assessed by taking into consideration the 95% confidence interval ($p = 0.05$ significance level). However, when the difference was significant, the level of significance was adapted by using Bonferroni correction (Akbulut 2010; Büyüköztürk 2005).

Table 2 Range values of the scores obtained from the measurement instruments used in the research

Scales/Factors	(T−)[a]	(T−)[a]	(LT)[a]	(T±)[a]	(HT)[a]	(T+)[a]	(T++)[a]
Programming self-efficacy perception scale/*simple tasks*	3.00–5.57	5.58–8.15	8.16–10.73	10.74–13.31	13.32–15.89	15.90–18.47	18.48–21.05
Programming self-efficacy perception scale/*complex tasks*	6.00–11.14	11.15–16.29	16.30–21.44	21.45–26.59	26.60–31.74	31.75–36.89	36.90–42.04
Programming self-efficacy perception scale *(total)*	9.00–16.71	16.72–24.43	24.44–32.15	32.16–39.87	39.88–47.59	47.60–55.31	55.32–63.03
Questionnaires	(AI−)[b]	(GI−)[b]	(SA)[b]	(A±)[b]	(OA)[b]	(GA+)[b]	(CA+)[b]
Course perception questionnaire *(item)*	1.00–1.86	1.87–2.71	2.72–3.57	3.58–4.43	4.44–5.29	5.30–6.14	6.15–7.00
Course perception questionnaire*(total)*	8.00–14.86	14.87–21.73	21.74–28.60	28.61–35.47	35.48–42.34	42.35–49.21	49.22–56.08
Scales/Factors	(N−)[c]	(L−)[c]	(Q ±)[c]	(G+)[c]	(VG+)[c]		
General self-efficacy perception scale *(total)*	0–17	18–35	36–53	54–71	72–89		
Success grade	Low	Intermediate	High				
Success grade	0–57	58–79	80–100				

[a]*T−* I don't trust at all, *T−* I generally do not trust, *LT* I trust a little bit, *T±* fifty–fifty, *HT* I highly trust, *T+* I generally trust *TT+* I completely trust

[b]*AI−* absolutely inadequate, *GI−* generally inadequate, *SA* somehow adequate, *A±*: fifty–fifty, *OA* quite adequate, *GA+* generally adequate, *CA+* completely adequate

[c]*N−* none, *L−* a little, *Q±* quite, *G+* good, *VG+* very good

In the analysis of qualitative data, the content analysis method was used. In order to ensure the validity of the study, care has been taken to the fact that the research questions and the research process were clear, consistent and verifiable by different researchers. In this respect, the purpose of the research, the characteristics of the study group, the characteristics of the data collection tools and the process of the course implementation are explained in a clear and detailed manner. Since the number of participants who filled out the semi-structured interview form used to determine their opinions on the course was very small, findings from the analysis of these questions related to the response of the research question were presented directly from the participants' opinions according to the themes.

Findings

Analysis of perceptions about the course, programming self-efficacy perceptions and general self-efficacy perceptions

Related to the answers to the first research question of this study (i.e., *What are the perceptions of pre-service teachers about the course, computer programming self-efficacy, and general self-efficacy?*), the lowest and highest score values of the participants obtained from the course perception questionnaire and self-efficacy scales, the standard deviations of the arithmetical score averages and distributions are presented in Table 3.

In Table 3, it is seen that the programming self-efficacy scale factors and the scale average, general self-efficacy scale and course perception questionnaire averages of the students are between 18.35 and 64.23. It is seen that pre-service teachers' programming self-efficacy perceptions for simple tasks are "I generally trust", their programming self-efficacy perceptions for complex tasks are "I highly trust" and they perceive programming self-efficacy in general as "I generally trust". It is also seen that pre-service teachers perceive their general self-efficacy as "Good". In addition, the pre-service teachers perceive the course as "Quite Adequate". In order to examine the perceptions of the pre-service teachers about the course in more detail, Table 4 presents the percentage (%) item-based distribution of perceptions of the participants.

Table 4 shows, how participants perceive the contribution of this course to their field knowledge (e.g., *m1, m2, m3*), the contribution of it to the knowledge and skills

Table 3 Perception levels of the pre-service teachers related to the course, their programming self-efficacy, and general self-efficacy

Scale/questionnaire	*N*	Min	Max	$\bar{X}$	Score Interval	Sd
Programming self-efficacy scale (simple tasks)	26	3	21	18.35	T+	5.17
Programming Self-efficacy scale (complex tasks)	26	19	42	30.42	HT	7.45
Programming self-efficacy scale (total)	26	25	63	48.77	T+	10.82
General self-efficacy scale (total)	26	45	80	64.23	G+	9.73
Course perception questionnaire	26	17	56	39.31	OA	12.05

Table 4 Item-based descriptive statistics about perceptions of the pre-service teachers

Items	AI−* %	GI−* %	SA* %	A±* %	OA* %	GA+* %	CA+* %	$\bar{X}$	Sd
M1.This course met my expectations and my learning needs	3.8	3.8	15.4	11.5	26.9	23.1	15.4	4.85	1.62
M2.The information presented in this course was suitable for my information and knowledge levels	0.0	15.4	3.8	7.7	19.2	34.6	19.2	5.12	1.68
M3.This course contributed positively to my personal development	11.5	7.7	11.5	7.7	15.4	26.9	19.2	4.65	2.04
M4.I gained new information and skills about teaching of programming at K-12 level by the help of this course	7.7	11.5	0.0	19.2	7.7	19.2	34.6	5.04	2.05
M5.This course has increased my motivation for programming instruction at K-12 level	7.7	15.4	7.7	11.5	23.1	23.1	11.5	4.42	1.88
M6.This course has increased my interest in programming tools to teach programming skills at K-12 level	11.5	7.7	3.8	11.5	23.1	30.8	11.5	4.65	1.90
M7.In this course, I have gained strategic and effective knowledge and skills that I can practice in my teaching profession	3.8	3.8	11.5	11.5	11.5	23.1	34.6	5.31	1.78
M8.This course has given me new professional knowledge and skills that I can share with my colleagues who are teachers or pre-service teachers	0.0	15.4	3.8	3.8	23.1	23.1	30.8	5.27	1.76

AI− absolutely inadequate, *GI−* generally inadequate, *SA* somehow adequate, *A±* fifty–fifty, *OA* Quite adequate, *GA+* generally adequate, *CA+* completely adequate

*$N = 26$

related to programming instruction (e.g., *m4, m5, m6*) and its contribution to their professional development (e.g., *m7, m8*).

According to the item-based analysis of the course perception questionnaire in Table 4, the course meets the students' course expectations and learning needs "quite adequately" (26.9%), the students find the information presented in the course "generally adequate" to the levels of their field knowledge (34.6%), and it often contributes to their personal development (26.9%). Based on these findings, it can be stated that the course is "generally adequate" for the development of the pre-service teachers' field knowledge.

In Table 4, it is seen that the course has definitely contributed to the acquisition of new knowledge and skills about programming instruction (34.6%), the students believe that it is quite influential on their motivation about programming instruction (23.1%) and their interest in computer programming tools have increased by the help of this course (30.8%). According to this, it can be said that this course has given new knowledge and skills related to the field of computer programming teaching to the pre-service teacher at a sufficient level.

According to Table 4, it is seen that this course is absolutely adequate (34.6%) in terms of acquiring strategic and effective knowledge and skills that the pre-service teachers can apply in their teaching profession (30.6%) and that they found the course absolutely adequate in terms of acquiring new professional knowledge and skills they can share with their colleagues (30.8%). According to these findings, it can be said that the pre-service teachers perceive the contribution of this course to their professional development very positively. Therefore, regarding the teaching of computer programming skills, it can be said that this course has contributed to the development of pedagogical knowledge of pre-service teachers.

Analysis of the perceptions related to the course according to various variables

Whether perceptions of the students differed according to the variables mentioned within the frame of the second research question of the study (i.e., *Do the perceptions of the pre-service teachers about the course differ according to their gender, perceived level of programming knowledge, self-efficacy perceptions related to programming, and general self-efficacy perceptions?)* was analyzed.

Table 5 shows the analysis of the total scores of the students in the course perception questionnaire according to gender variable by Mann–Whitney-*U* Test. The perceptions of the pre-service teachers about the course did not significantly differ ($U=76.00$; $z=-0.211$; $p=0.833>0.05$) according to the gender. The

Table 5 Mann Whitney-*U* test results of the perception questionnaire scores of the pre-service teachers according to the gender

Gender	*N*	Mean rank	Sum of ranks	*U*	*Z*	*p*
Female	10	13.90	139.00	76.00	−0.211	0.833
Male	16	13.25	212.00			

Kruskal–Wallis test was applied to determine whether the perceptions of students about the course differ according to their perceived computer programming knowledge levels, programming self-efficacy, and general self-efficacy (Table 6).

As seen in Table 6, the perceptions of the students on the course are not significantly different in terms of the level of knowledge they perceive about programming ($x^2 = 0.642$, sd = 2; $p > 0.05$). The perceptions of the students about the course do not differ significantly in terms of programming self-efficacy perceptions ($x^2 = 2.944$, sd = 4; $p > 0.05$). However, the perceptions of the students about the course are significantly different ($x^2 = 9.164$; sd = 2; $p < 0.05$) in terms of general self-efficacy perceptions.

Mann–Whitney-*U* tests were conducted to identify on which group or groups the difference was based according to Table 6. Bonferroni correction was applied and the significance level for all effects was accepted as 0.0167 (Akbulut 2010; Büyüköztürk 2005). In the bilateral comparisons, it was found that while there was no significant difference between the total scores of the course perception questionnaires of the students with "very good" and "good" general self-efficacy ($U = 21.00$; $z = -1.572$, $p = 0.116 > 0.0167$), there was a significant difference between the total scores of the students with "very good" and "quite good" general self-efficacy. This difference was in favor of the group having "very good" self-efficacy ($U = 0.500$, $p = 0.012 < 0.0167$, $z = -2522$, $r = 0.80$) and it was at the greatest effect level ($r > 0.5$). However, it was found that there was a significant difference between the total scores of the students whose course perception questionnaire and the general self-efficacy perceptions were "good" and "quite" good. This difference was in favor of the group with "good" general self-efficacy ($U = 11{,}00$; $p = 0.160 < 0.0167$; $z = -2398$; $r = 0.52$) and it also had a significant effect ($r > 0.5$)(Field 2005, pp. 565–566; Fritz et al. 2012, p. 12).

Based on these findings, it can be concluded that gender, perceived programming knowledge levels, and programming self-efficacy have no effect on the perceptions of pre-service teachers related to the course they had taken about teaching of computer programming at the K-12 grades, but that general self-efficacy perceptions have a significant effect on the perception of this course.

Analysis of the course academic achievements according to various variables

The academic achievement scores of the students were analyzed according to the variables mentioned within the frame of the third research question of the study: (e.g., *Do the academic achievements of the pre-service teachers regarding the course differ according to their gender, the level of success of their previous programming courses, and their general academic achievements?*).

Mann–Whitney-*U* test was used to determine whether the academic achievement scores of the students differ according to the gender variable. Results showed that the students' success scores of the course did not differ significantly according to gender ($U = 46.50$, $z = -1.768$, $p > 0.05$) (Table 7).

The Kruskal–Wallis test was applied to determine whether the academic achievement scores of the students in the course differ according to the success scores of the previous computer programming courses (PL1 and PL2).

Table 6 Kruskal-Wallis test results related to the pre-service teachers' scores obtained from course perception questionnaire in terms of perceived programming knowledge levels, programming self-efficacy, and general self-efficacy

Scale	Score interval	*N*	Mean rank	Sd (*df*)	X^2 (χ^2)	Difference	*p**
Perceived programming knowledge levels	High	5	14.70	2	0.642	None	0.725
	Intermediate	20	13.48				
	Low	1	8.00				
Programming self-efficacy perceptions	I completely trust	9	14.72	4	2.944	None	0.567
	I generally trust	7	13.43				
	I quite trust	5	15.40				
	Fifty-fifty	3	12.33				
	I trust a little	2	5.25				
General self-efficacy perceptions	Very good	5	19.70	2	9.164	Very good-quite, Good-quite	0.010*
	Good	16	14.13				
	Quite	5	5.30				

*$p < 0.05$

Table 7 Mann Whitney-*U* Test results of the course success scores of the pre-service teachers according to the gender

Gender	N	Mean rank	Sum of ranks	*U*	*Z*	*p*
Girl	10	16.85	168.50	46.50	− 1.768	0.077
Boy	16	11.41	182.50			

Table 8 Kruskal-Wallis test results in terms of pre-service teachers' course achievement scores according to their previous programming course achievement levels and general academic achievements

Achievement score	Score range	N	Rank average	Sd (df)	X^2 (Chi *S*quare)	Difference	*p**
Comp. lang I	High	6	14.92	2	4.794	No	0.091
	Intermediate	12	16.04				
	Low	8	8.63				
Comp. lang II	High	1	7.00	2	2.589	No	0.274
	Intermediate	16	15.31				
	Low	9	11.00				
Gen.aca. ach.	High	7	17.64	2	6.078	No	0.048*
	Intermediate	17	13.06				
	Low	2	2.75				

*$p < 0.05$

According to Table 8, the students' scores related to the course were not significantly different in terms of achievement scores of Prog. Lang.1 ($x^2=4.794$; sd = 2; $p=0.091>0.05$) and Prog.Lang.2 ($x^2=2.589$; sd = 2; $p=0.274>0.05$) courses.

According to the results of the Kruskal–Wallis test applied to determine whether the students' academic achievement scores of the course differed significantly in terms of general academic achievement scores, it was found that the course success scores significantly differed from the general academic achievement scores ($x^2=2.944$, sd = 4; $p<0.05$) (Table 8). Mann–Whitney-*U* tests were performed to find out on which group or groups the difference was based according to Table 8. Bonferroni correction was applied and the significance level for all effects was accepted as 0.0167. In the bilateral comparisons, it was determined that there was no significant difference between the students with "high" and "intermediate" general academic achievement scores in terms of course academic achievement scores ($U=37.50$, $z=-1.399$, $p=0.162>0.0167$); there was no significant difference between the students with "high" and "low" general academic scores in terms of course achievement scores when the Bonferroni correction was taken into account ($U=0.000$; $z=-2.058$; $p=0.040>0.0167$); and, there was no significant difference between the students with "intermediate" and "low" general academic scores in terms of the course achievement scores ($U=2.50$; $z=-1.929$; $p=0.054>0.0167$). Accordingly, it can be stated that the general academic achievements and the success in previous programming courses do not make any difference on the achievements of the pre-service teachers in the elective course related to the teaching of computer programming at K-12 level.

Pre-service teachers' opinions on the course and teaching of computer programming at the K-12 grades

Within the scope of the fourth research question of the study, opinions of the students about the course were analyzed under five main themes: (e.g., *opinions on the course, opinions on computer programming tools, recommendations for the course; necessity of teaching of computer programming at K-12 grades,* and *opinions on personal development efforts in this field*). Findings are presented in Table 9 with direct quotations from participants' opinions.

When they were asked about their opinions regarding the teaching of computer programming at the primary and secondary school (e.g., 5th–8th grades) levels and their efforts to develop themselves in this direction, the majority of the participants ($f_{\text{necessary}}=6$) said that the teaching of computer programming should be at K-12 level, and that they showed an effort to develop themselves about teaching of computer programming at K-12 level ($f_{\text{Ihave}}=4$) or they needed it ($f_{\text{shouldbe}}=2$). On the other hand, there were two participants ($f_{\text{none}}=2$) who indicated that computer programming teaching at K-12 level was unnecessary or that this was not the priority for them.

When the opinions of the participants about the elective course are examined, it is seen that there are no negative opinions and the opinions are positive ($f_{\text{positive}}=4$) or partially positive ($f_{\text{partlypositive}}=4$). Regarding the content of the course and the teaching methods and techniques used in the course, k1, k2, and k5 completely had positive opinions. On the other hand, k3 had similar opinions but did not fully believe the necessity of what he/she learned. In addition, in terms of teaching of computer programming, k4, k6 and k8 stated that Small Basic and Alice tools were not appropriate for K-12 grades and different (Code Game Lab, Code.org) computer programming tools might be more appropriate instead.

When participants were asked whether they had enough knowledge to be able to teach computer programming with the tools that they had learned in the course, almost all of them said that they found the information they had learned about the tools to be satisfactory ($f_{\text{adequate}}=2$, $f_{\text{adequate. But...=}}5$, $f_{\text{too much!=}}1$). However, it was found out that depending on their experiences of teaching practice, participants pointed out to the existing facilities of schools (k3) and they thought that there should be a focus on tools based on block-based programming because they were easier to learn (k3, k5, k8). K6 stated that he would need to develop himself more in order to use the programming tools he/she learned in the course process.

Pre-service teachers' responses regarding the improvement of the course for the future were coded in three sub-themes including suggestions for resources, suggestions for computer programming tools, and recommendations for teaching/learning materials in terms of teaching of computer programming at the K-12 grades. According to Table 9, k1 emphasized the need to concentrate on tools having Turkish sources. k2, k6, and k8 have made their suggestions especially for the part of the course related to Alice 3 tool. While k2 expressed that he/she liked the three-dimensional and animation design features of Alice 3 tool, k6 and k8 stated that they had difficulties related to the sources about Alice 3 and that the lack of sources was a disadvantage when he/she wanted to use this tool for teaching computer programming

Table 9 Opinions of pre-service teachers

Theme	Sub theme	*f*	Opinions
My views on the course	Positive	4	"It was efficient and it was a chance to recognize current coding software….. it certainly contributed to me in terms of the programming tools… I learned about the Alice program I had never knew." (K1) "Considering the curriculum, current tools have been used in this course… The programs we have studied in the coding course are the programs that we can use in our professional life… The evaluation criteria were also suitable in terms of both encouraging the lessons and encouraging the research. The course was very enjoyable although it was intensive. I have been involved both in the implementations and the project willingly. "(K2) "Overall, the training was good. I think it was efficient in terms of objectives of the course…. it has had a big contribution to my classroom practice about teaching coding. I learned about the important transfer methods in the teaching of coding. "(K5) "We learned to use more than one program…. it was good that we learned the programs used today" (k7)
	Partly positive	4	"The course was useful, but more emphasis should be given to how teachers should teach this issues to students. I think that it was a very effective learning process in terms of the tools we worked with, but I have doubts about the necessity. "(K3) "I had the chance to add more to my coding knowledge. Especially the projects we developed with Alice was very useful for me… I do not think that I can use only code-based tools such as Small Basic at K-12 level. But I think I have gained useful information about how to use Scratch in teaching."(K4) "I found the course related to the teaching of Coding useful except the one about Alice 3 program… but it was good that the programs we used were up to date."(k6) "I think that the rationale of coding was taught us in a comprehensive and sufficient way. The practices and activities of the course were adequate. But it would be better to teach Code Game Lab or Code.org instead of Alice 3. At the moment depending on my observations in my teaching practice and conversations with other teachers, there is no teaching in the Alice environment. Apart from this, the course aimed to cover all the learning outcomes at a sufficient level…..the information taught was a very good foundation for me. I think I got most of the necessary information through this course. We are currently teaching the Code Game Lab in our teaching practice. It would have been better if we also worked on it."(K8)

Table 9 (continued)

Theme	Sub theme	*f*	Opinions
The tool I learned in the course (Scratch2, Small Basic, Alice3) were…. for my teaching…	Adequate	2	"I think it was at an adequate level"(K1)
			"We already had a basic level of knowledge because we had previously taken courses about algorithm and programming, but we were incompetent because we did not have previous knowledge about the programs we studied in this course. However, I think we have learned about the programs at the basic level and gained information about how to teach these programs to K-12 students in a limited time." (K2)
	Adequate but	5	"It was adequate. To be honest, my friends and I are observing in our internship that students have difficulties even with the usage of "Microsoft word". For this reason, there are some problems in terms of the putting what we learned into practice."(K3)
			"I think Small Basic and Alice are challenging tools for that level of students. Simpler tools such as code.org and scratch will be more useful. "(K4)
			"It was at an adequate level but I think that we concentrated too much on some programs. For example, instead of focusing too much on the Alice 3 or Small Basic programs, we could concentrate on more appropriate programs for kids such as Code Game. I also think that there should be information about computer-free coding education. "(K5)
			"I think we have taken these courses at the simplest level. I think that we can teach programming to the students but not totally. I believe we should definitely improve ourselves" (k6)
			"Parts related to the Scratch and Small Basic were pretty good and adequate. However, more emphasis was placed on Alice-3 than it should be (in the project section). Information about Kodu Game Lab or Code.org could have been covered rather than Alice 3 environment. Or it would have been better to prefer completely different tools instead of this "(k8)"
		1	"I think it was too much" (k7)

Table 9 (continued)

Theme	Sub theme	*f*	Opinions
Recommendations about the course		1	"Tools with Turkish sources should be preferred" (k1)
	Too much! Resources Tools	6	"I think that my most favorite tool used in this course was the Alice3 program and I believe that it is necessary to use it more in education because of its being three dimensional and the realism of the animations that are made" (K2) "I think Alice 3 should be in Turkish as well and it should be a little more useful." (K6) "I think the part related to Alice 3 in the course should be changed. It is not a highly-taught program, and there are also source constraints. Instead, there may be different alternatives"(K8) "Different tools can also be covered as part of this course." (k4), (k7) "Different tools can be used for the K-12 education. In most schools, there are no laboratories so more appropriate tools should be introduced in this case. "(k5)
	Activity Material	2	"I can say that the activities should be more." (k1) "There should be more activities on how the teacher can teach this issues to his students in the class" (k3)
K-12 grades prog. teaching	Is necessary	6	Giving the coding instruction to the students from the early ages is very positive and useful in terms of presenting suitable tools for the development of their creativities. "(k1) "I do not think the necessary support has been given. The infrastructure should be provided and it should be given in a more serious way as soon as possible. "(k5) "Coding is a course that should be taken as a basis for future programming experiences… I think this course should be given to students at a young age. But this does not mean that the course should be comprehensive and compulsory for all students who love or dislike it. Those who are interested in can be involved in the field of computer programming. And it develops the algorithmic thinking in the ones who have no interest. As a result, I do not think there will be anybody who will not benefit from it. These are my personal thoughts. "(k8)
	I'm undecided about	2	"I do not know exactly if it is appropriate to be given at the level of K-12" (k7) "I have doubts about its necessity." (k3)

Table 9 (continued)

Theme	Sub theme	*f*	Opinions
Personal development effort for the teaching of programming.	I have	4	"Yes, because I want to go on with the teaching profession, I think that I need to improve myself in K-12 level computer programming and especially because I find Alice3 program very enjoyable, I am trying to improve myself in this program " (k2) "Yes, I am trying to create interactive materials. I try to develop myself in making designs that can attract the attention of the K-12 students. "(k4) "Yes, I am working on new tools and methods on learning with Code Game and computer-free code education." (k5) "Yes, I am trying to get a certificate of programming." (k6)
	Should be	2	"I want to improve myself… we have to develop myself in order to keep up with the latest technology and new developments " (k1) "Yes, actually. I bought several books on coding. I was learning projects related to Arduino. But the KPSS process is more important for me. Therefore, For now, I have given a break to my studies on coding. "(k8)
	None	2	"No, because I have no interest" (k3) "To be honest, my priority is to finish my university" (k7)

at K-12 grades. Therefore, they recommended that this part of the course should be changed. While k4 and k7 stated that it was possible to focus on other different tools in general, by drawing attention to the lack of laboratories in schools, k5 indicated that different computer programming tools could be taught. On the other side, k1 and k3 suggested that more activities about the usage of these tools in their teaching process should be integrated into the elective course.

According to the findings in Table 9, it can be said that the opinions of pre-service teachers who took this course were positive and consistent with the findings of the first research question of the study to determine their perceptions of the course. However, although they found the learning outcomes of the course related to computer programming tools adequate, they suggested that tools such as Kodu would be easier to use than Alice 3 and Small Basic especially at the secondary school (e.g., 5th–8th grades) level. It can be said that pre-service teachers' recommendations for the curriculum of the course are directed more towards the computer programming tools taught in the course and that according to them, the course content needs to be updated in line with the different tools that could be used more easily at the secondary school (e.g., 5th-8th grades) level. It can be also stated that according to Table 9, the pre-service teachers believe that they should be trained to be able to teach at K-12 grades and they should give importance to their personal development in this direction.

Discussion and conclusion

In this study, an elective course design for the teaching of computer programming at the K-12 grades that pre-service computer teachers take in their final year of undergraduate education was explained and the factors affecting perceptions and academic achievements of the pre-service teachers were investigated. In this student-centered course, pre-service teachers gained knowledge about study and research related to the teaching of computer programming at the K-12 grades and gained experience in block-based Scratch 2, code-based Small Basic and Alice 3 tool which was predominant with its three-dimensional and object-based features.

In the study, it was seen that pre-service teachers' perceptions and opinions about the course were generally positive. They perceived the course adequate in terms of its contribution to their field knowledge, acquiring new knowledge and skills in the field of computer programming teaching at the K-12 grades and the improvement of their professional skills, in other words, their pedagogical content knowledge.

It was seen that they had positive opinions indicating that they had gained sufficient knowledge about the programming tools they experienced (e.g., *Scratch 2, Small Basic, Alice 3*) in the course. Accordingly, it can be concluded that the course designed on the basis of student-centered methods is sufficient to provide the pre-service teachers with the knowledge and skills related to computer programming teaching at K-12 grades. There are research findings supporting this result in the literature (Yükseltürk and Altıok 2015, 2016). Within this framework, it can be emphasized that the teaching of computer programming instruction to pre-service computer teachers at the undergraduate courses will contribute positively to their professional and personal development.

In this research, it was determined that the gender of the pre-service teachers, their perceived programming knowledge level and programming self-efficacy did not make any difference in their perceptions about the course, but they differed significantly according to the general self-efficacy. According to this, it was seen that the pre-service teachers with good and very good self-efficacy perceived the course more positively than the teachers with medium general self-efficacy. Nevertheless, it was determined that there was no difference in the academic achievement of the course according to the gender of the pre-service teachers, the success in the programming courses they had in the previous semesters and the general academic achievements. Based on these findings it can be said that the activities for providing pre-service computer teachers with knowledge and skills in computer programming tools related to the teaching of computer programming at K-12 grades can also be presented in different vocational and field courses in 1st, 2nd or 3rd classes of the undergraduate education. However, in their study in which they examined the TPACK level of the secondary school computer teachers, Doukakis et al. (2010) emphasized the need for training on methods for integrating technological tools into teaching even though computer science teachers have more experience and knowledge than the teachers who teach other subjects. They also indicated that teachers needed to study on appropriate training scenarios including examples from daily life for their students. In addition, there are also research findings suggesting that self-efficacy of pre-service teachers develops through the training they receive during their undergraduate education (as cited in Çapri and Çelikkaleli 2008). Accordingly, it can be stated that it is better to develop the professional skills of the pre-service teachers in the teaching of computer programming at the K-12 grades in the periods when they are closer to the professional life.

As another implication of the finding of this research that the perceptions of pre-service teachers differ only according to their general self-efficacy, and that the achievements of the course are not related to the general academic success and overall academic achievements of the previous programming courses. It can be said that programming tools which are used for teaching computer programming at the K-12 grades can also be taught to different field teachers or pre-service teachers. The results of the research studies focusing on teaching the computer programming tools to students who have no computer programming experience, and indicating that students can use these tools easily and they develop their computer programming skills (Akçay and Özden 2011; Kelleher et al. 2007; Mladenovic et al. 2016; Wilson and Moffat 2010) can be used as a support for this.

In recent years, the importance of computational thinking has been emphasized in order to achieve success, to develop and to ensure the economic competitiveness in the digital society (ISTE 2011, Meb 2017). Since computational thinking focuses on enhancing the competence of the individual with regard to skills such as problem-solving and analytical thinking to overcome complicated problems, it is emphasized that it is needed to investigate the ways of placing computational thinking in the programs and practices of the K-12 teachers and managers and to integrate it with the existing programs (Israel et al. 2015; Yadav et al. 2016). In his study in which he presented the first findings of a project on integrating the computational thinking into STEM fields, Swaid (2015) emphasized that technological developments make it necessary for STEM educators to associate teaching curricula and instruction with computational thinking skills and computer science. Assessing within this framework, it can be said that the education that different field teachers/or pre-service teachers take to be able to improve their knowledge and skills related to the concepts of computer programming and the programming tools will contribute to the integration of the computational skills and their teaching process.

On the other hand, the reason for the finding that the pre-service computer teachers' perceptions and academic achievements of the course are not affected by previous knowledge and perception levels of the computer programming might also be due to the nature of the computer programming tools they experience in this course because computer programming tools are environments whose main purpose is visualization of programming and whose target group is the novices in the computer programming. These tools are appropriate for developing multimedia applications such as games, animation, and the digital story in the context of the computer programming basics (Price and Barnes 2015). So, it could be used in different teaching fields such as math, science, or art. In addition, it is said that any field teachers who are accepted as expert students can help their students or guide them to create, discover, and improve their strategies to use in their own learning environments. This is also true for K-12 CS education field where there shouldn't be any expectation that each teacher can be a sufficient coder (Nickels 2018). Israel et al. (2015) conducted a research with elementary school teachers who had limited experience in the field of computer science, and who participated in a training program designed to integrate computational thinking into their courses. In this research, they stated that these teachers had an important progress in terms of their attitude towards computer science and the products they created at the end of a four-month training period. For this reason, it can be said that, if appropriate support is created for different field teachers, they will be able to develop their computer programming skills, apply them to their own teaching and continue to learn and develop by experiencing them together with their students. However, in order to be able to support this judgment, there is a need for further research showing how and to what extent teachers develop their computer programming skills, how they relate it to their fields, and how they teach (Lockwood and Mooney 2017; Yadav et al. 2016).

Based on the opinions of the pre-service teachers in this study, it is seen that especially Small Basic and Alice 3 tool were assessed as difficult tools to use and teach at the secondary schools (e.g., 5th–8th grades) in Turkey. Instead of these tools, they offered suggestions for teaching different block-based tools within the scope of the course. On the other hand, in the literature, although there are a number of studies related to the use of different approaches and strategies (such as block-based, code-based, coding without tools, collaborative learning, numerical thinking, etc.) on how to teach computer programming skills, there isn't explicit evidence about which of them can minimize teaching and learning difficulties for teachers and students (Pears et al. 2007; Sentence and Csizmadia 2017). Therefore, based on this finding of the research, it can only be said that pre-service teachers perceive block-based programming tools as easier-to-use tools.

Based on the qualitative findings of the study, it was observed that the pre-service teachers had initiatives and positive attitudes towards self-improvement in the field of computer programming teaching at K-12 grades. According to this, it can be said that the course contributes to the creating awareness in pre-service teachers about the field of computer programming teaching. However, because of the very small number of participants reporting qualitative opinions in this research, it can be said that further research studies depending on the experience of teachers and pre-service teachers in computer programming tools are needed to find out which computer programming tools are preferred for the K-12 grades and the reasons for these preferences. In this way, regarding their goals, it would be easier for the teachers to choose the right tool among this relatively large number of computer programming tools which are used for teaching at the K-12 grades. It is also possible to promote awareness and information about tools in which they can present their own knowledge (e.g., forum, social network, etc.) and can access information about learning/teaching sources (e.g., eba.gov.tr/kod; code.org).

As seen in the findings of this research, pre-service teachers have also suggested that tools with more Turkish learning/teaching sources should be included in the course and that the number of activities carried out in the course should be increased. While they made some suggestions for the tools which they experienced in the course, they did not have any negative comments about the activity-based and student-centered implementation of the course. Considering this feedback and research findings, the following suggestions can be presented for course/module/training programs about the teaching of computer programming at K-12 grades:

- It is not possible to know/teach every computer programming tools at K-12 grades. Therefore, a content should be created according to the needs of the target group.
- Trainings for teaching of computer programming to the computer teachers/pre-service teachers should be designed according to student-centered methods.
- In these courses, activities should be designed to provide more material/activities for different grade levels/learning area by using programming tools rather than only focusing on the use of the tools.
- A forum or social networking where teachers/pre-service teachers can share their experiences regarding computer programming tools and share their learning and teaching resources about computer programming tools can be used.
- In their qualitative research in which they have evaluated the secondary school information systems and software course teaching program depending on the opinions of computer teachers, Çakır and Tazıcı (2016) have emphasized that there is a need for making teachers knowledgeable about computer programming tools and that there are issues to be examined regarding the technical infrastructure of the schools in the effective implementation of computer programming teaching. As a result of this study, taking into account the qualitative opinions of pre-service teachers indicating that there are inadequacies of schools and laboratories for teaching computer programming, it can be suggested to include content and activities related to the coding without tools in computer teacher training programs.

In the light of the results of this study, further research studies to be carried out on the following topics will contribute to the field of computer programming education at K-12 grades.

- It is recommended to investigate when and at which class level the pre-service teachers should take the courses focusing on computer programing tools in order to improve their professional knowledge and pedagogical skills related to these tools.
- There is a need for further evidence to determine the contribution and impact of the development of knowledge and skills of different field teachers in computer programming tools on their teaching. For this, it may be suggested to conduct research to reveal the effects of teaching the computer programming tools to different field teachers or pre-service teachers on their learning and teaching skills in their field.
- In terms of developing effective applications in the field of teaching computer programming, it is considered necessary to investigate the reasons why pre-service teachers prefer block-based programming tools as more usable.

The most important limitation of this research is that only a limited number of pre-service teachers participated in this research and that a one-term study was conducted. For this reason, in order to examine the findings of this study in depth, further research designing similar courses and examining the positive and negative aspects of them is needed.

Finally, about 3 months after the completion of this research, changes which would be started to be implemented in the next academic year, were made in the curriculum of education faculties in Turkey (Yök 2018b). Some courses for teaching computer programming skills have been added to new teacher education programs to train computer teachers for primary and secondary school levels and also teachers from different fields(e.g., secondary school mathematics teaching program) (Yök 2018b). In the new program, some courses related to teaching of computer programming at the K-12 grades have been added to the program from the first year until the fourth year of the computer teacher education. Similarly, a course related to the learning computer programming has been added to the program of teaching the secondary school mathematics teaching program. To what extent the new teacher education programs that have quite drastic changes in Turkey how will be effective in the integration of computer science field at the K-12 grades, should be considered as a new and debatable topic. On the other hand, it should be stated that the findings of this research study support the changes in the new program.

Acknowledgements The first version of this work was presented as an oral presentation at 12th Symposium of Computer and Instructional Technologies in Izmir, Turkey.

References

Akbulut, Y. (2010). *Sosyal bilimlerde SPSS uygulamaları: Sık kullanılan istatiksel analizler ve açıklamalı SPSS çözümleri [SPSS applications in social sciences: Frequently used statistical analyzes and SPSS solutions with explanation]*. İstanbul: İdeal Kültür Yayıncılık.

Akçay, T. (2009). *Perceptions of students and teachers about the use of a kid's programming language in computer courses* (Unpublished Master's Thesis). The Graduate School Of Natural And Applied Sciences Of Middle East Technıcal Unıversity, Ankara, Türkiye.

Akçay, T., & Özden, M. (2011). Perceptions of students and teachers about the use of a kid's programming language in computer courses. In A. Méndez-Vilas (Ed.), *Education in a technological world: Communicating current and emerging research and technological efforts* (pp. 321–336). Badajoz: Formatex Research Center.

Akdağ, M. (2016). Bilişim teknolojileri öğretmen adaylarının özel alan yeterliliklerinin çeşitli değişkenler açısından incelenmesi [Investigation of special field qualifications of teacher candidates in computer technologies in terms of various variables]. *Elektronik Sosyal Bilimler Dergisi [Electronic Journal of Social Sciences], 15*(59), 1215–1231.

Akpınar, Y., & Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi [The need for programming education in information society schools]. *İlköğretim Online [Elementary Education Online], 13*(1), 1–4.

Albion, P. R. (1999). Self-efficacy beliefs as an indicator of teachers' preparedness for teaching with technology. In *Proceedings of the Society for Information Technology & Teacher Education International Conference* (pp. 1602–1608). Association for the Advancement of Computing in Education (AACE).

Altun, A., & Mazman, S. G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin türkçe formunun güvenirlik ve geçerlik çalışması [Reliability and validity study of the Turkish form of the perceived self-efficacy perception about programming scale]. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi [Journal of Measurement and Evaluation in Education and Psychology], 3*(2), 297–308.

Balanskat, A., & Engelhardt, K. (2015). *Computer programming and coding: priorities, school curricula, and initiatives across Europe: computing our future*. European Schoolnet. Retrieved from http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0 on April 16, 2018.

Bandura, A. (1994). *Self-efficacy*. V. S. (Ed.), Encyclopedia of Human Behavior (Vol. 4) (pp. 71–81). New York: Academic Press (Reprinted in H. Friedman [Ed.], Encyclopedia of mental health. San Diego: Academic Press, 1998).

Berry, M. (2013). *Computing in the national curriculum: A guide for primary teachers*. Computing at school. Retrieved from http://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf on April 10, 2018.

Boechler, P., Artym, C., Dejong, E., Carbonaro, M., & Stroulia, E. (2014, July). Computational Thinking, Code Complexity, and Prior Experience in a Videogame-Building Assignment. In *Proceedings of IEEE 14th International Conference on Advanced Learning Technologies (ICALT)* (pp. 396-398). Athens, Greece: IEEE.

Bray-Clark, N., & Bates, R. (2003). Self-efficacy beliefs and teacher effectiveness: Implications for professional development. *Professional Educator, 26*(1), 13–22.

Burgett, T., Folk, R., Fulton, J., Peel, A., Pontelli, E., & Szczepanski, V. (2015). DISSECT analysis of pedagogical techniques to integrate computational thinking into K-12 Curricula. In *Proceedings of Frontiers in Education Conference (FIE)* (pp. 1338–346). El Paso, Texas: IEEE.

Büyüköztürk, Ş. (2005). *Sosyal bilimler için veri analizi el kitabı: istatistik, araştırma deseni, spss uygulamaları ve yorum [Data analysis handbook for social sciences: statistics, research design, spss practices and interpretation]*. Ankara: Pegem Akademi Yayıncılık.

Çakır, H., & Tazıcı, K. (2016). Bilişim teknolojileri ve yazılım dersine ilişkin öğretmen görüşleri [Teachers' views on computer technologies and software course]. *Türkiye Sosyal Araştırmalar Dergisi [Turkey Social Research Journal], 2016*(1), 29–45.

Çapri, B., & Çelikkaleli, Ö. (2008). Öğretmen adaylarının öğretmenliğe ilişkin tutum ve mesleki yeterlik inançlarının cinsiyet, program ve fakültelerine göre incelenmesi [Investigation of preservice teachers' attitudes towards teaching and professional self-efficacy beliefs according to their gender, programs, and faculties]. *İnönü Üniversitesi Eğitim Fakültesi Dergisi [Inonu University Faculty of Education Journal], 9*(15), 33–53.

Clements, D. H., & Fullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology, 76*(6), 1051–1058.

codemonkey. (2018). *playcodemonkey.com*. Retrieved from February 23, 2018, from https://www.playcodemonkey.com/.

code.org. (2018). *code.org*. Retrieved February 23, 2018, from https://code.org/.

Cooper, S., Dann, W., & Pausch, R. (2000). Alice: A 3-D tool for introductory programming concepts. In *Proceedings of the fifth annual CCSC northeastern conference (CCSC '00) and on the journal of computing in small colleges, 15*(5), (pp. 107–116). Mahwah, NJ, USA: Consortium for Computing Sciences in Colleges.

Creswell, J. (2016). *Araştırma deseni: nitel, nicel ve karma yöntem yaklaşımları [Research design: qualitative, quantitative and mixed method approaches]*. (S.B. Demir, Trans.) Ankara: Eğiten Kitap.

Demir, Ö., & Seferoğlu, S. S. (2017). Yeni kavramlar, farklı kullanımlar: Bilgi-işlemsel düşünmeyle ilgili bir değerlendirme [New Concepts, Different Uses: An Evaluation of Computational Thinking]. In H. F. Odabaşı, B. Akkoyunlu & A. İşman (Eds.), *Eğitim teknolojileri okumaları 2017 [Educational technology readings 2017]* (pp. 801–830). Sakarya: TOJET & Sakarya University.

Demirer, V., & Nurcan, S. A. K. (2016). Programming education and new approaches around the world and in Turkey. *Eğitimde Kuram ve Uygulama [Journal of Theory and Practice in Education], 12*(3), 521–546.

Demirtaş, H., Cömert, M., & Özer, N. (2011). Öğretmen adaylarının özyeterlik inançları ve öğretmenlik mesleğine ilişkin tutumları [Pre-service teachers' self-efficacy beliefs and attitudes towards profession]. *Eğitim ve Bilim [Education and Science], 36*(159), 96–111.

Diethelm, I., Hubwieser, P., & Klaus, R. (2012). Students, teachers, and phenomena: educational reconstruction for computer science education. In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research* (pp. 164–173). Koli, Finland: ACM.

Doukakis, S., Psaltidou, A., Stavraki, A., Adamopoulos, N., Tsiotakis, P., & Stergou, S. (2010). Measuring the technological pedagogical content knowledge (TPACK) of in-service teachers of computer science who teach algorithms and programming in upper secondary education. Readings in technology and education: Proceedings of ICICTE, (pp. 442–452.). Corfu Island, Greece.

Duncan, C., & Bell, T. (2015). A pilot computer science and programming course for primary school students. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, (pp. 39–48). London, UK: ACM.

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, (pp. 60–69). Berlin, Germany: ACM.

Durmuşoğlu, M. C., Yanık, C., & Akkoyunlu, B. (2009). Türk ve Azeri öğretmen adaylarının öğretmenlik mesleğine yönelik tutumları [Turkish and Azerbaijani pre-service teachers' attitudes to their profession]. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi [H.U. Journal of Education], 36,* 76–86.

Dursun, F. (2013). *Bilisim teknolojileri ogretmen yeterliklerinin ogretim elemanı, ogretmen adayı ve ogretmen goruslerine gore degerlendirilmesi [Assessment of teacher competencies in information technologies according to teaching staff, teacher candidates and teacher opinions] (Unpublished Doctoral Thesis)*. Aydın: Adnan Menderes University, Social Science Institute.

Erol, O., & Kurt, A. A. (2017). BÖTE bölümü öğrencilerinin programlamaya karşı tutumlarının incelenmesi [Investigation of CEIT students' attitudes towards programming]. *Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Dergisi [Mehmet Akif Ersoy University Journal of Education Faculty], 1*(41), 314–325.

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.

Field, A. (2005). *Discovering statistics using SPSS* (2nd ed.). London, UK: Sage Publications Ltd.

Fincher, S. (1999). What are we doing when we teach programming? In *Proceedings of Frontiers in Education Conference (FIE'99)* (pp. 29th Annual (Vol. 1, pp. 12A4-1)). San Juan, Puerto Rico, USA: IEEE.

Finson, K., Riggs, I. M., & Jesunathadas, J. (2000). *The Relationship of Science Teaching Self Efficacy and Outcome Expectancy to the Draw-a-Science-Teacher-Teaching Checklist.* Retrieved March 28, 2018, from https://files.eric.ed.gov/fulltext/ED442642.pdf .

Fritz, C. O., Morris, P. E., & Richler, J. J. (2012). Effect size estimates: current use, calculations, and interpretation. *Journal of Experimental Psychology: General, 141*(1), 2–18.

Google Blockly. (2018). https://developers.google.com/blockly/. Retrieved from March 12, 2018, from https://developers.google.com/blockly/.

Gorman, H., & Bourne, L. E. (1983). Learning to think by learning logo. Rule learning in third grade computer programming. *Bulletin of Psychonomic Society, 21,* 165–177.

Hiltunen, T. (2016). *Learning and teaching programming skills in finnish primary schools—The potential of games (Unpublished Master Thesis)*. Oulu: University of Oulu.

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education, 82,* 263–279.

ISTE. (2011). Computational Thinking in K–12 Education Leadership Toolkit. Retrieved from March 01, 2018, from http://www.iste.org/docs/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4.

Kaptan, S. (2011). *Bilimsel Araştırma ve İstatistik Teknikleri [Scientific Research and Statistics Techniques]* (11th ed.). Ankara: Bilim Yayınevi.

Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling Alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1455–1464). Montreal, Canada: ACM.

Kember, D., & Leung, D. Y. (2008). Establishing the validity and reliability of course evaluation questionnaires. *Assessment & Evaluation in Higher Education, 33*(4), 341–353.

Korkmaz, Ö., & Altun, H. (2013). Mühendislik ve BÖTE öğrencilerinin bilgisayar programlama öğrenmeye dönük tutumları [Engineering and CEIT student's attitude towards learning computer programming]. *International Journal of Social Science, 6*(2), 1169–1185.

Kukul, V., Gökçearslan, Ş., & Günbatar, M. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation, and reliability. *Educational Technology: Theory and Practice*, *7*(1), 158–179. https://doi.org/10.17943/etku.288493.

Lockwood, J., & Mooney, A. (2017). *Computational Thinking in Education: Where does it fit? A systematic literary review.* Retrieved from March 12, 2018, from https://arxiv.org/ftp/arxiv/papers/1703/1703.07659.pdf: arXiv preprint arXiv:1703.07659.

Lye, S. Y., & Koh, J. H. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41,* 51–61.

MEB. (2016). *Bilgisayar bilimi dersi öğretim programı (Kur 1 ve Kur 2) [Computer science curriculum (1*st. *and* (2nd ed.). Ankara, Turkey: Ministry of National Education General Directorate of Secondary Education.

MEB. (2017). *Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı (1-8.sınıf)-Taslak Program [Information Technologies and Software Course Curriculum (1-8th grade)].* Ankara: Turkey Ministry of National Education Basic Education General Directorate.

MEB. (2018). *Bilişim teknolojileri ve yazılım dersi öğretim programı (ortaokul 5. ve 6. sınıflar) [Information technologies and software course curriculum (5th and 6th grade in middle school)].* Ankara: Turkey Ministry of National Education.

Mercimek, B., & İlic, U. (2017). Bilişim teknolojileri ve yazılım dersi öğretim programı güncelleme önerisine yönelik bir değerlendirme [An evaluation for update suggestıon of informatıon technologies and software course curriculum]. *Academia Eğitim Araştırmaları Dergisi [Academia Educational Research Journal], 2*(1), 1–9.

Microsoft. (2017a). *Microsoft Small Basic*. Retrieved from September 01, 2017, from Microsoft Small Basic: http://smallbasic.com/.

Microsoft. (2017b). *Microsoft Small Basic: Programlamaya Giriş [Microsoft Small Basic: Introduction to Programming].* Retrieved September 01, 2017, from http://download.microsoft.com/download/6/3/1/631B7D3A-C24D-41DE-AC35-F421D2826271/Introducing_Small_Basic.pdf.

Mladenovic, M., Rosic, M., & Mladenovic, S. (2016). Comparing elementary students' programming success based on programming environment. *International journal of modern education and computer science, 8*(8), 1–10.

Nesiba, N., Pontelli, E., & Staley, T. (2015). DISSECT: Exploring the relationship between computational thinking and English literature in K-12 curricula. In *Proceedings of Frontiers in Education Conference (FIE)* (pp. 249–256). El Paso, Texas: IEEE.

Nickels, M. (2018). *How Do We Prepare Teachers to Teach Coding?* Retrieved April 28, 2018, from GettingSmart: http://www.gettingsmart.com/2016/06/prepare-teachers-teach-coding/.

Öncel, Ü. (2016). *Çocuklara özel 15 online kod yazmayı öğrenme servisi [15 child-specific online coding learning services].* Retrieved April 27, 2016, from https://webrazzi.com/2016/04/27/cocuklar-icin-kod-yazmayi-ogrenme-servisleri/.

Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies, 22*(2), 421–443.

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., et al. (2007). A survey of literature on the teaching of introductory programming. *ACM SIGCSE Bulletin, 39*(4), 204–223.

Prensky, M. (2008). *Programming Is the New Literacy*. Retrieved January 13, 2018, from https://www.edutopia.org/literacy-computer-programming#comments.

Price, T. W., & Barnes, T. (2015). Comparing textual and block interfaces in a novice programming environment. In *Proceedings of the eleventh annual International Conference on International Computing Education Research* (pp. 91–99). Omaha, NE, USA: ACM.

programlamacocukoyuncagi. (2018). *Programlama çocuk oyuncağı [Programming child player*]. Retrieved March 12, 2018, from http://www.programlamacocukoyuncagi.org.tr/.

Ramsden, P. (1991). A performance indicator of teaching quality in higher education: The course experience questionnaire. *Studies in Higher Education, 16*(2), 129–150.

Rubeck, M. L., & Enochs, L. G. (1991). A path analytical model of variables that influence science and chemistry teaching self-efficacy and outcome expectancy in middle school science teachers. In *Paper presented at the annual meetings of the National Association for Research in Science Teaching, Anaheim, CA*.

Rushkoff, D. (2012). *Code Literacy: A 21st-Century Requirement*. Retrieved November 13, 2002 from, https://www.edutopia.org/blog/code-literacy-21st-century-requirement-douglas-rushkoff.

Şahin, C., & Şahin, S. (2017). Öğretmen adaylarının öğretmenlik mesleğine yönelik tutumları, öz-yeterlik inançları ve öğrenciyi tanıma düzeyleri [Teacher candidates' attitudes towards teaching profession, self-proficiency beliefs and levels of recognition of students]. *Journal of Turkish Educational Sciences, 15*(2), 224–238.

Scratch. (2018). *Scratch*. Retrieved March 26, 2018, from Scratch: https://scratch.mit.edu/.

Şendağ, S., & Erol, O. (2012). Ilköğretim ikinci kademede Alice 3 boyutlu animasyon yazılımının bilişsel araç olarak kullanımına yönelik bir durum çalışması [A case study on the use of Alice 3d animation software as a cognitive tool in primary education second stage]. In Abstract Book of 6th International Computer and Instructional Technologies Symposium. Gaziantep, Turkey.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies, 22*(2), 469–495.

Shailaja, J., & Sridaran, R. (2015). Computational thinking the intellectual thinking for the 21st century. *International Journal of Advanced Networking & Applications, May 2015 Special Issue*, 39–46.

Swaid, S. (2015). Bringing computational thinking to STEM education. *Procedia Manufacturing 3 (2015)—6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015)* (pp. 3657–3662). Elsevier.

Tekin, H. (1996). *Eğitimde ölçme ve değerlendirme (9.baskı) [Measurement and evaluation in education (9th. Edition)].* Ankara, Turkey: Yargı Yayınları.

Tynker. (2018). *Four Reasons Why Kids Should Learn to Program.* Retrieved March 12, 2018, from https://www.tynker.com/blog/articles/ideas-and-tips/coding-at-school/four-reasons-why-kids-should-learn-programming/.

Vicki, D. (2018). 15 + Ways of Teaching Every Student to Code (Even Without a Computer). Retrieved March 23, 2018, from Edutopia: https://www.edutopia.org/blog/15-ways-teaching-students-coding-vicki-davis.

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE), 18*(1), 3.

Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. In *Proceedings of the 22nd Annual Psychology of Programming Interest Group*. Leganés, Spain: Universidad Carlos III de Madrid.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st-century problem solving in K-12 classrooms. *TechTrends, 60*(6), 565–568.

Yıldırım, F., & İlhan, I. O. (2010). Genel öz yeterlilik ölçeği türkçe formunun geçerlilik ve güvenilirlik çalışması [Validity and reliability study of the Turkish self-efficacy scale]. *Türk Psikiyatri Dergisi [Journal of Turk Psychiatry], 21*(4), 301–308.

YÖK. (2018). Bilgisayar ve Öğretim Teknolojileri Öğretmenliği Lisans Programı [Undergraduate Curriculum in Teaching Computer and Instructional Technologies]. Retrieved March 12, 2018, from http://www.yok.gov.tr/documents/10279/49665/bilgisayar_ogretim/86c99d2e-3973-41c6-9e98-ed6d816391db.

YÖK. (2018b). Yenilenen Öğretmenlik Lisans Programları [Renewed Teaching Undergraduate Programs]. Retrieved May 16, 2018 from, http://www.yok.gov.tr/ogretmen-yetistirme-lisans-programlari.

Yükseltürk, E., & Altıok, S. (2015). Bilişim teknolojileri öğretmen adaylarının bilgisayar programlama öğretimine yönelik görüşleri [Pre-Service Information Technologies Teachers' Views on Computer Programming Teaching]. *Amasya Üniversitesi Eğitim Fakültesi Dergisi [Amasya University Journal of Education Faculty], 4*(1), 50–65.

Yükseltürk, E., & Altıok, S. (2016). BT Öğretmen Adayları K-12 Seviyesinde Bilgisayar Programlama Eğitimi Hakkındaki Seminere Neden Katılmak İster? [IT Teacher Candidates K-12 Level Computer Programming Training Seminar Why Attend?] *In proceedings of 10th International Computer and Instructional Technologies Symposium (ICITS)*, (pp. 216–222). Rize, Turkey.

Funda Dağ is currently an assistant professor in the Department of Computer Education and Instructional Technology at Kocaeli University, Turkey. Her research interests focus technology integration in learning environments, on pre and in-service teachers' technology training, instructional design and constructivist approaches for teaching and learning. Additionally, she has researched in the usability of the educational tools, and about TPACK development of prospective teachers and personalized learning.