European Journal of Educational Research

Volume 8, Issue 2, 477 - 489.

ISSN: 2165-8714
http://www.eu-jer.com/

A Proposal of In-Service Teacher Training Approach for Computer Science Teachers

Serhat Bahadir Kert *
Yildiz Technical University, TURKEY

Received: January 3, 2019 ▪ Revised: February 19, 2019 ▪ Accepted: March 4, 2019

Abstract: Recently, there has been growing interest among practitioners and scientists in teaching children computer programming languages. The international efforts to raise generations who produce technologies are supported at the national level in Turkey too. Programming language education is included beginning from secondary school curricula of computer science education. However, it can be mentioned that up-to-date methodological and pedagogical requirements of the courses are not adequately researched. Therefore, the primary aim of the study is to share innovative methods regarding programming education processes with middle school computer science teachers. This paper presents the details of a project conducted to design an in-service training model for computer science teachers and funded by the Scientific and Technological Research Council of Turkey (TUBITAK) with the project number of 116B298. During the training period, up-to-date methodology and technologies were presented in workshops with an integrated approach. At the end of the courses, the participants' development and the efficiency of the activities were investigated based on the analyses of qualitative and quantitative data and positive results about the content of proposed in-service teacher training methodology were yielded.

Keywords: *Programming and programming languages, STEM teaching/learning strategies, secondary education, improving classroom teaching.*

To cite this article: Kert, S. B. (2019). A proposal of in-service teacher training approach for computer science teachers. *European Journal of Educational Research*, *8*(2), 477-489. doi: 10.12973/eu-jer.8.2.477

Introduction

The increased demand for technology consumption is noteworthy in our social life in which digital content use is becoming more widespread daily. The emerging consumption demand can be supported through computer training designed to develop individuals' digital productivity and computational thinking abilities. As Barger (2008) indicates, when computer literacy is limited to basic software use, it only includes computer literacy practices and lacks the authorship dimension. It is important to equip individuals with computer science thinking skills not only for programming skills but also to solve daily problems. In recent years, computer science education has been taken out of its technical boundaries and is considered important for both social life and the development of basic thinking skills that each individual should acquire. Social tendency directly affects and directs the development and renewal of science and technology. Wing (2008) refers to three important drivers that direct information technologies, which are society, science, and technology. The cyclical nexus among the three drivers is presented in Figure 1.

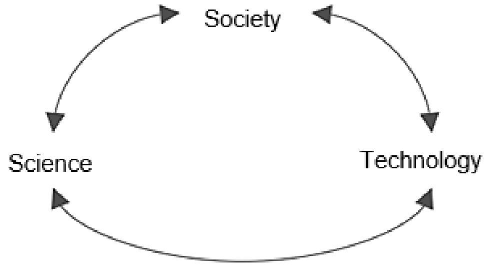

Figure 1. The cyclical nexus among society, science, and technology (Source: Wing, 2008)

* **Correspondence:**
Serhat Bahadir Kert, Yildiz Technical University, Computer Education and Instructional Technologies, Istanbul, Turkey.
🖂 sbkert@yildiz.edu.tr

Figure 1 shows the relationships between social demand, scientific invention, and technological production are noteworthy. Today, the tendency towards coding studies based on computer science is an output of social expectations related to technological productivity at the international level. From a general perspective, programming education around the world has attracted international interest in recent years. This perspective, which comes to the forefront with computer science practices such as block-based programming education studies and robotic applications, is based on information from almost 50 years ago. In the mid-1990s, Alan Perlis, a prominent mathematician and pioneer researcher in the field of computer science, was the first to express that algorithmic thinking processes can be seen in all areas of social life (Denning, 2017). On the other hand, the pioneer studies claiming students' procedural thinking skills can be developed through programming languages are based on Seymour Papert's studies with the LOGO programming language since 1980 (Grover & Pea, 2013). Papert, a computer scientist influenced by the constructivist approach in his studies with Piaget, asserted the constructionism approach. He focused on learning with the emphasis on the emergence of learning by internalizing the application process within the existing situation (Ackermann, 2019). Following Seymour Papert, important computer scientists have conducted studies on constructionism and studied the pedagogical characteristics of computer education (Brennan, 2015; Harel & Papert, 1991; Kafai & Resnick, 1996). In this regard, Papert (1980) first used computational thinking, which has become a part of the studies on computer science education, in his book Mindstorms: Children, Computers, and Powerful Ideas. Computer science education applications in which children used educational robots named Turtle and the LOGO programming language to draw were also included in this book.

The international community has renewed interest in Seymour Papert's studies, which first used the notion of computational thinking, in the last decade with Wing (2006)'s article titled Computational Thinking. Jeanetta Wing contributed to the formation of the concept with her subsequent studies (Wing, 2008; 2010; 2016). Within the basic framework, computational thinking indicates the logical process to reach a solution can be formulated in a way that a machine or person can apply it (Wing, 2010). The ever-increasing digital product range has made the existence of individuals with computational thinking and increased technological productivity increasingly more important. The growing interest in the development of technological productivity and thinking skills through computer-based applications makes it necessary to enhance the pedagogical competencies of computer science teachers. In this regard, in-service training programs in which computational thinking, block-based applications, and robotic training are provided play a crucial role.

In this study, an in-service training model for information and communication technology teachers was proposed and the efficiency of the model was examined based on the evaluations before, during, and after the training process. The applications of whole training process were part of a project funded by TUBITAK. In compliance with the main aim following research questions were investigated:

RQ.1. Is there a significant difference between pretest and posttest academic achievement scores of the teachers participating proposed in-service teacher training process?

RQ.2. What are the teachers' opinions on the efficiency of the proposed in-service teacher training process?

Literature Review

In recent years, there has been noteworthy growth of interest among the CS educators in computational thinking education has brought new pedagogical and technological approaches to existing CS learning environments. In accordance with these approaches, for an effective CS teaching process, teachers need to know how they will integrate block-based coding tools, robot development kits and unplugged CS activities into their courses. Therefore, CS teachers need to have a comprehensive in-service training including both new technologies and pedagogies used in CS education. In-service training means a renewal and development process in every professional field. The changing educational technologies and approaches reveal the need for renewal and development for teachers as well. In-service training provides the most effective support for teachers who play a facilitating and guiding role within the classroom in flexible, learner-centered, and technological education processes in order to provide occupational sustainability (Raud & Orehhova, 2017). Attempts to increase teachers' qualifications in multiple dimensions are among the main purposes of in-service training for teachers. In-service training can be defined as "the processes that aim to equip teachers with knowledge, attitude, and habits that are necessary to provide the targeted qualifications in education and to equip teachers with occupational knowledge, attitude, and habits that are proved deficient in the light of scientific and socioeconomic facts" (Erdem & Simsek, 2013, p. 1). Considering the occupational competencies defined in this basic definition, the studies on in-service training focus on four dimensions: (1) investigating the cooperation types between administrators and teachers in planning and implementing in-service training, (2) investigating the educational change of the teacher as an individual, (3) investigating the areas in which in-service training is provided, and (4) analyzing the special needs of novice teachers (Neil, 1986). The educational change and innovative applications regarding this change constitutes an important dimension of in-service training. Arslan and Sahin (2013) proposed that in-service training contributes to teachers' professional development in three ways: learning new knowledge, occupational experience, and occupational sharing. Information and communication technology teachers need in-service training to learn updated content knowledge and methodological innovations.

Studies on programming language education are prominent in the computer science education literature. These studies can touch on new methodological suggestions as well as technological support tools. Shaw (2012) associated the learning process in programming language education with learning types and the profiles of participants in online environments. He proposed using hybrid learning environments during programming education and highlighted the effect of online exercises and interactions on programming performance. His results revealed that among the participant profiles, "adaptive" learners are more successful in the programming course than other groups. This study can be used as an example of the studies considering programming education in the methodological dimension. In the same vein, Pereira, Zebende, and Moret (2010) proposed a new methodology in programming education. Based on the Turing theory, which constitutes the basis of coding, they asserted that teaching programming with simple instructions is more effective than teaching with complex code blocks and verified their hypothesis with an experimental study. Ismail, Ngah, and Umar (2010) emphasized that the deficiencies in the skills of analytic thinking, problem-solving, and algorithm development negatively influence learning a programming language. Besides the methodological studies about programming education, there are studies on the efficiency of computer science education with tangible tools in the literature. Corral, Balcells, Estevez, Moreno & Ramos (2014) used the Sifteo cubes in programming education. Their study tried to increase the efficiency of the process in programming education based on pedagogical purposes. According to Corral et al. (2014), a student completes the process of learning a programming language by passing through four stages. First, the student defines the programming content needed in the active learning stage. Then, he tests the codes by typing them in the application, checks for bugs with reflective observation, and records new knowledge learned by experience as concepts in his mind. Physical objects and robot making are important for learners' motivation in younger age groups. Benitti (2012) made a systematic literature review on the use of robotic applications in education and revealed that such applications provide significant advantages in terms of studying analysis and students' motivation. Lego MindStorm robotic coding studies are among such educational tools. Software that supports computer literacy education is also important to provide motivation and visual learning. Kert, Kayak, Erkoc and Avincan (2014) in their study entitled Children Who Design Their Play made play design and coding activities with children and adopted an approach dealing with two dimensions. In this regard, they created educational modules and design documents. In the same vein, there are many studies on programming education with or without play for younger age groups (Bers, Flannery, Kazakoff & Sullivan, 2014; Corral et al., 2014; Fessakis, Gouli, & Mavroudi, 2013; Howland & Good, 2015; Kordaki, 2010; Lau & Yuen, 2011; Navarrete, 2013). In addition to the technology-supported environments in computer science education, there are environments in which some students cannot reach computers. In such environments, activities without computers are preferred. An international organization called "CS Unplugged" presents examples of such activities. The activities are listed according to subject and are presented with worksheets, activity stages, and exemplary videos. In-class and out-of-class programming language education activities can be found on the site (CSERG, 2017).

In the light of current computer science education studies in the literature, the Ministry of Education (MoE) updated the Information and Communication Technologies Curriculum in Turkey in 2017. The aims of the curriculum since September 2017 are as follows:

Information Technologies and Software curriculum teaches students to

1. Be individuals who better understand technological concepts, systems, and practices as technological citizens,
2. Use information technologies effectively and relevantly,
3. Reach, explore, and use internet-based services,
4. Create a general understanding and technical knowledge about computer science,
5. Acquire and develop skills of problem-solving and computational thinking,
6. Follow and evaluate reasoning processes,
7. Acquire the skills of cooperative studying as part of the learning process, benefit from social environments, and share what they learned,
8. Search for learning opportunities in the internet environment,
9. Develop an understanding regarding algorithm design and express it verbally and visually,
10. Choose and implement appropriate programming approaches to solve problems,
11. Create technical knowledge about programming,
12. Use at least one programming language at a good level,
13. Conduct studies in design and manage products,
14. Develop innovative and original projects for problems encountered in daily life (the problems that elderly and handicapped individuals encounter, etc.),

15. Raise an awareness regarding lifelong learning (MoE, 2017).

When these aims are examined, thinking skills, problem-solving, and the development of programming products are prominent headings. These subheadings constitute a very crucial part of the curriculum. Given these noteworthy developments in CS education, it looks as an important requirement to design effective in-service training activities towards CS teachers. Doukakis, Koilias, Adamopoulos, and Giannopoulou (2013) emphasize that CS teachers working at secondary school level need to develop their knowledge on integrating technology in their courses and teaching strategies for the algorithms. This kind of training can be given through a digital platform or face to face instruction approach. In their experimental study, Alkaria and Alhassan (2017) found that the teachers using electronic training platform had significantly higher academic scores than those having face to face in-service training. Within the scope of this project, an in-service face to face training model was designed, implemented, and evaluated for information and communication technology teachers, in light of the updated curriculum aims and the current studies in the literature.

Methodology

Research Design

The study was designed as a mixed method study, which had qualitative and quantitative data from three different groups for three weeks. Quantitative section was adopted to quasi-experimental design with the implementation of pretest and posttest. The training for each group lasted 5 days and covered one subheading per day. The analyses of the group data was conducted both before and after each group meeting.

Participants

Participants' applications were received through a pre-questionnaire form shared through announcements in the online environment. A total of 135 computer science teachers pre-applied for the training that was planned to reach 90 participants (30 per week). The answers given in the survey form indicated teachers' willingness to participate in the study and their previous knowledge. Final approval emails were sent according to the application priority. Eighty teachers stated their desire to participate in the 3-week training one week before the training began; however, the total number of participants was limited to 32. Although a very limited number of teachers participated in the study compared to the number of applicants, the rate of voluntary participation in a flexible educational course was satisfactory.

Instruments

Academic Achievement Test, Social Network Sharing, Rubric Measurements, Product Evaluations, and Questionnaire Data were used to assess the effectiveness of each training module. The data collection tools used in all evaluation stages are summarized in Table 1.

Table 1. The Data Collection Tools used in the Evaluation of the Project

Evaluation Stage	Data Collection Tool
Pre-evaluation Stage	Pre-Activity Expectation and Prequalification Questionnaire (A1) Academic Achievement Pretest including the methodology, software, and hardware that was used during the training (ABT1)
Process Investigation	The content analysis of social network platforms during the training.
Product Evaluation	Scenarios written during the training (U)
The stage of result evaluation	The perspectives survey on the process after the training (A2) Academic Achievement Posttest including the methodology, software, and hardware used during the training (ABT2) Objectives Check Rubric to evaluate if national course objectives were met, which was one of the aims of the project (KKR)

A1: A short survey was designed by the researcher in order to determine the participants' pre-competencies in the concepts and applications such as computational thinking. Computer science teachers were required to complete the survey online during the application.

ABT1: Experts and trainers developed an academic achievement test to assess knowledge of the content areas to be taught. The ideal question number on the test was 20 after conducting item analysis and reliability coefficient calculations. The KR-20 analysis was used for the reliability of the test. A reliability coefficient higher than .50 was reached (α=0.74; >0.50).

S1: In the implementation process, daily product examples were shared on social network platforms opened on behalf of the group. This was done to increase the widespread effect of the training. This approach will be used in the process analysis. The comments on the shared applications were examined and inferences regarding the training were derived.

U: Product evaluation was used to yield data that can support process analysis. In the activities, participants were asked to design application examples of training regarding computational thinking skills without computers through scenarios in which computer science is related to daily life. Participants' internalization of the targeted objectives was investigated in these products.

A2: A survey was developed by the researcher to examine whether the process of five-staged in-service training met participants' expectations. The perspectives obtained by the survey are presented in percentages/frequencies tables.

ABT2: The pretest was administered again as the posttest.

KKR: Participants were asked on a rubric whether the nine objectives related to the project's aims were met during the in-service training. The following nine objectives were included in the rubric and participants were asked to rate the applications from one to five regarding whether the applications provided contribute to realizing the objectives.

Students will be able to:

"Explain the advantages provided by object-based programming languages."

"Use variables in order to explain knowledge and make calculations with them."

"Notice that functions are created through constants and variables in order to perform a task."

"Explain the concept of object."

"Program objects in order to realize a series of actions."

"Creates a series of objects based on classes."

"Explains the concept of class."

"Uses loops in a repeating group within the program."

"Test and improve a program in order to reach the targeted outcome."

Procedure

Innovative methods, techniques, and strategies were integrated into the activities in the in-service training for information and communication technology teachers. The five main dimensions utilized in the training model are depicted in Figure 2.

Day 1: The activities without computers in the programming language

Day 2: Basic computational thinking skills training

Day 3: The use of utility software

Day 4: The use of utility hardware

Day 5: The methodologies and practices based on pedagogical approaches

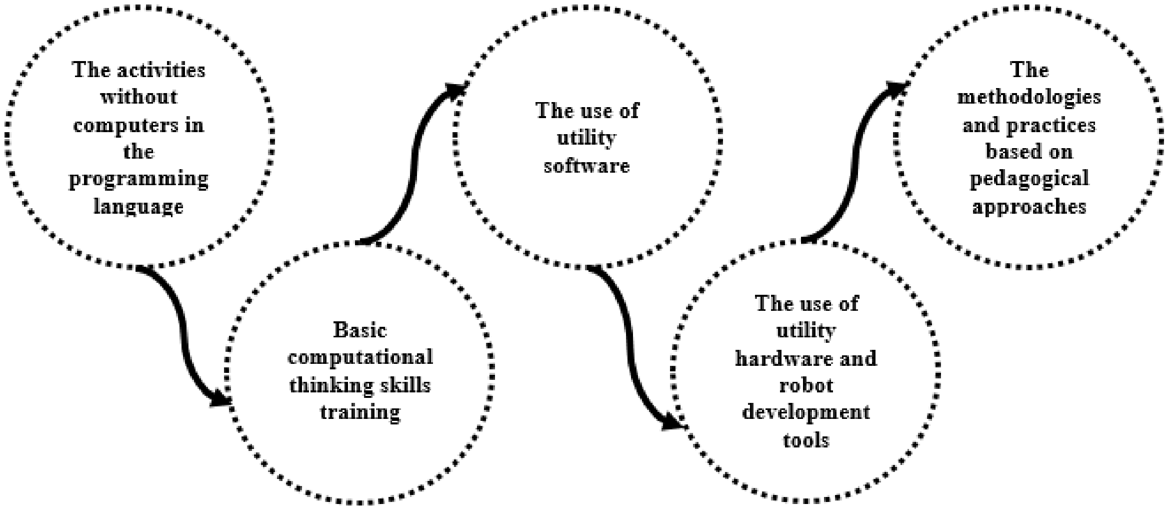
The activities without computers in the programming language
Basic computational thinking skills training
The use of utility software
The use of utility hardware and robot development tools
The methodologies and practices based on pedagogical approaches

Figure 2. The stages of one week in-service training for information and communication technology teachers

During the one-day training regarding the activities without computers in the programming language, the examples of how to conduct programming education without computers were shared. The main purpose of this stage was to support teachers who cannot perform programming education due to technical deficiencies. It also aimed to raise awareness that programming education is not just a process of writing code in front of computers, but also includes skills such as problem-solving, critical thinking, and multi-dimensional thinking.

Through Basic Computational Thinking Skills Training, an application-based learning environment was designed regarding how computer science education at the social level can be used to equip individuals with computational thinking skills.

The Use of Utility Software is a learning process to provide information about the properties of the utility software prominent in teaching computer science. It focused on how to use new software platforms effectively for younger age groups and applications were shared through the Kodu Game Lab example.

The Use of Utility Hardware was shared with teachers in order to raise awareness regarding the tangible tools used in computer science education. The applications of robot development were carried out and Lego MindStorm EV3 sets bought with TUBITAK support were used in the training.

The Methodologies and Practices Based on Pedagogical Approaches constituted an important part of the training. Generally, activities included in up-to-date technological software in programming language education are changing daily in software chaos. Teachers' tendencies to focus on specific software by moving away from the basic aim of programming education are emerging. Scratch, Kodu, Alice, etc. are focused on educational practices for children. However, the software fulfills pedagogical deficiencies; in other words, methodological infrastructure is ignored. From an integrated point of view, the pedagogical perspective was provided in this module. The in-service training, which was implemented over 18 hours with three different groups of teachers, was designed to increase the pedagogical and technological awareness of information and communication technology teachers regarding computer science education.

Announcements were made in order to promote the participation of middle school information and communication technology teachers. An online application and survey form was first created and published online. The brochure and the survey addresses were sent in an official letter to the Istanbul Provincial Directorate of National Education. In order to disseminate the brochure, an email to all colleges in Istanbul was sent and the announcements were made via the Turkish Informatics Teachers Association. The bilgiislemseldusunme@gmail.com email account was used to answer teachers' questions during the application process and the training. In order to support communication during the implementation and to disseminate activities, a Twitter account (https://twitter.com/kmptsynldsnm), an Instagram account (@kmptsynldsnm), and a Facebook account (Bilgi islemsel Dusunme) were also created. In order to solve the organizational problems during the implementation, three different meetings with the project team and a pilot study were conducted. Participants confirmed their participation through email one week before each activity week. For physical preparation of activities, the software in the university computer classroom was checked, all computers in the classroom were ensured to have the Kodu Game Lab software, the materials were finalized for the activities without computers, and final arrangements were done. In addition, the department funded the Lego Mindstorm parts needed for the robotic applications and contact with Lego Turkey was made. With these preliminary preparations, the efficiency of the training was addressed.

Data Collection

The changes that may have been caused by the education throughout the project and the opinions of participants regarding the training were evaluated using qualitative and quantitative data collection tools. The survey data collected before the project, social media shares throughout the training, scenario production processes, academic achievement test applied as pre- and posttests, participants' opinions after the implementation, objective-achievement test, and feedback collected through an objective control rubric were evaluated.

Findings

Demographics of the Participants

The information regarding the participants' demographic features and motivation was collected through a survey of the computer science teachers. The survey was composed of open-ended questions and the findings regarding the participant characteristics and the results of theme analysis are as follows:

- While it was found that the occupational experience of participants ranged from 1 to 12 years, the teachers in their first 5 years on the job were more interested in the in-service education training.
- The participants had basic programming language skills. Among the languages, C#, C++, Visual Basic, and Java were prominent.

- Most of the participants stated that their teaching experiences for children generally included programming education using SCRATCH block-based programming media.
- Data processing thinking was related to problem-solving and algorithmic thinking skills by the participants.
- Half of the participants stated that they participated in robotics education activities organized by other institutions.
- When asked why they participated in the in-service training program, the answers "I want to develop myself" and "I want to hear about the latest technologies regarding computer science education" were prominent.

The participant characteristics were composed of teachers who were open to innovation, interested about the developments in their occupations, and had technological proficiency.

Data Obtained from the Pretest and Posttest (ABT1-ABT2) Academic Achievement Tests

Throughout the three weeks of the project, the academic achievement in each group was monitored by pretest and posttest implementation of the academic achievement test. The data were analyzed for normal distribution and the results are given in Table 2.

Table 2. Normal Distribution Statistics Regarding Group Pretest Scores

		Shapiro-Wilk Values		
	Group	**Statistics**	**SD**	**p**
Pretest	1 (05-09 September 2016)	0.908	12	0.203
	2 (19-23 September 2016)	0.944	11	0.568
	3 (26-30 September 2016)	0.919	9	0.380

As it can be seen in Table 2, the pretest scores revealed a normal distribution (p>.05). After the pretest analysis, a normal distribution analysis of posttest scores was conducted. The data are shown in Table 3.

Table 3. Normal Distribution Statistics Regarding Group Posttest Scores

		Shapiro-Wilk		
	Group	**Statistics**	**SD**	**p**
Posttest	1 (05-09 September 2016)	0.920	12	0.286
	2 (19-23 September 2016)	0.938	11	0.498
	3 (26-30 September 2016)	0.794	9	0.017

When Table 3 was investigated, the posttest scores of Group 1 and Group 2 revealed normal distributions (p>.05); however, the posttest scores of Group 3 did not confirm the normal distribution hypothesis (p<.05). Therefore, parametric statistics were used for pretest-posttest comparisons in Group 1 and Group 2, whereas non-parametric statistical methods were used for Group 3. Group 1's paired samples t-test results regarding the development between pretest and posttest are given in Table 4.

Table 4. Comparison of Pretest and Posttest Academic Achievement Test and Paired Samples t-Test Regarding Group 1

	$\overline{X}$	N	SD	t	df	p
Pretest	8.6667	12	2.05971	-4.068	11	0.002
Posttest	11.9167	12	1.78164			

Table 4 shows a statistically significant difference between the pretest and posttest results of the participants who participated in the first week (p<.05). Thus, the training process revealed a statistically significant difference in the academic knowledge and skills of the participants. The changes in Group 2 were analyzed via a paired samples t-test to assess any changes and the results are given in Table 5.

Table 5. Comparison of Pretest and Posttest Academic Achievement Test and Paired Samples t-Test Regarding Group 2

	$\overline{X}$	N	SD	t	df	p
Pretest	8.5455	11	2.33939	-8.138	10	0.00
Posttest	13.8182	11	1.25045			

Table 5 shows a significant difference (p<.05) between the pretest and posttest scores of the participants in Group 2. Thus, the training process under the project significantly differentiated the academic knowledge and skill levels of participants in Group 2. Because the group data did not reveal a normal distribution, the significant difference analysis caused by the Group 3 training was analyzed by the non-parametric Wilcoxon signed ranks test and the results are shown in Table 6.

Table 6. Comparison of Academic Achievement Test and Wilcoxon X Signed Ranks Regarding Group 3

	$\bar{X}$	N	SD	Z	p
Pretest	7.7778	9	3.45607	2.67	0.008
Posttest	12.3333	9	1.32288		

Table 6 shows a statistically significant difference (p<.05) in the pre- and posttests of the participants regarding the academic achievement test scores. Therefore, the training procedure had a significant positive effect on the participants' academic skills and knowledge. According to the academic achievement test scores in general, the in-service training had a positive effect on participants.

Data Collected from the Social Network Shares throughout the Training Period

During the 3-week implementation period, social network accounts were opened to share the activities conducted. The daily activities were shared with the participating and non-participating users. Additionally, the views of the computer science teachers participating in the training were investigated through their written and visual comments on social networks. Figure 3 includes samples of the shares through social media accounts.

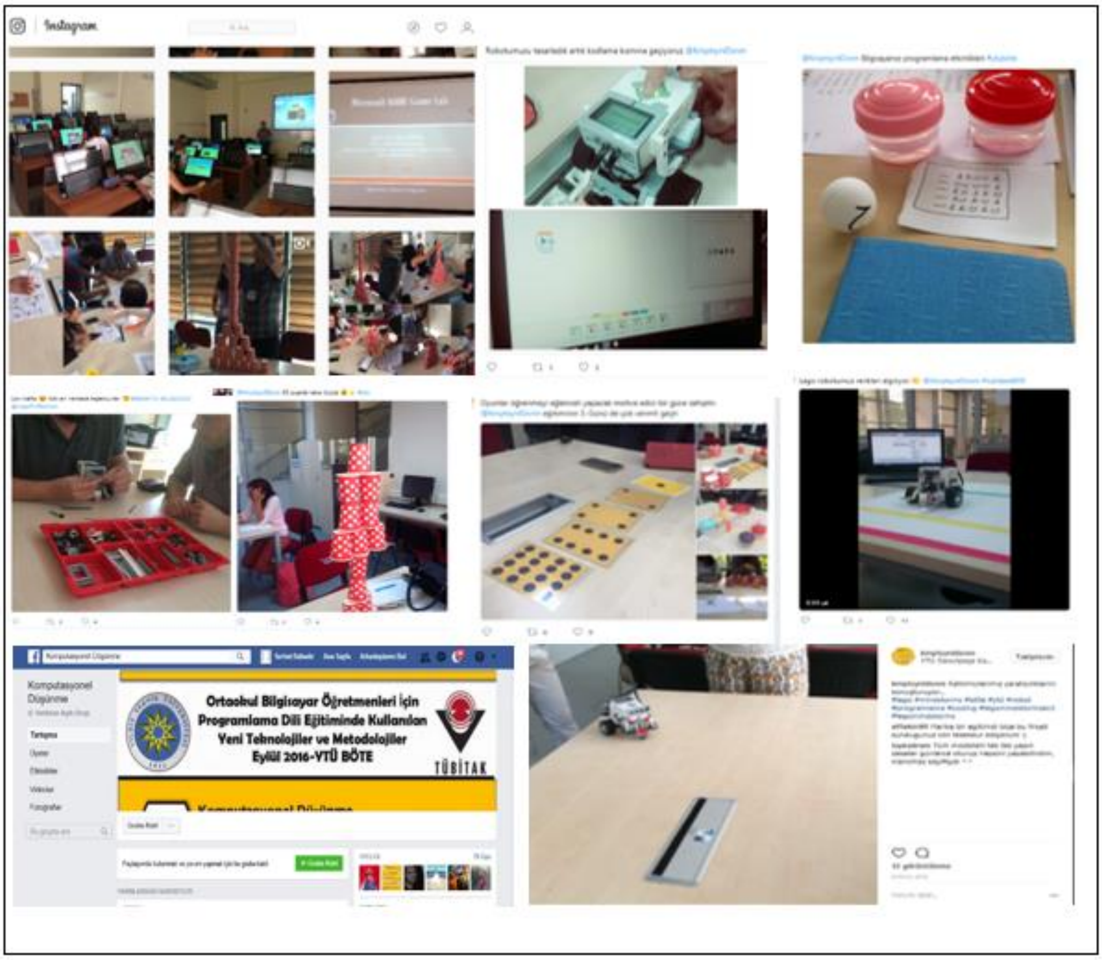
Instagram
Kompütasyonel Düşünme
Ortaokul Bilgisayar Öğretmenleri için Programlama Dili Eğitiminde Kullanılan Yeni Teknolojiler ve Metodolojiler Eylül 2016-YTÜ BÖTE
TÜBİTAK

Figure 3. Sample visuals of social media shares throughout the in-service training

When the shared visuals were investigated, programming without a computer, computational thinking skills, and robot development studies were prominent. Three activities among the computational thinking and programming without computer activities were shared most often.

1. *Paper cup tower:* In this activity, the participants were expected to form groups of 3 people and make the longest and strongest tower with the 25 paper cups and 2 rulers given to them. The tower making activity was recommended to develop decomposition, abstraction, and organization skills, which are included in computational thinking skills.

2. *Messaging with the Binary Coding Cards:* As an example of basic programming features and using computer architecture in education without computers, a messaging application with binary coding cards was presented. The participants were separated into two groups and tried to transfer messages with binary coding cards.

3. *Ranking with scales:* The ranking with scales activity was given as an example to teach the ranking algorithms included in programming education. In this activity, the participants tried to rank water containers, which were filled

with different amounts of water, according to their weight. They verified their predictions used scales they made with a ruler and a base point.

Along with the visuals, the written comments of the participants in the training also revealed a positive attitude. A few of the participants' comments include

> *"It was a perfect training, thank you for giving us such a chance."*
>
> *"I could even make all of the models for days, it was incredibly enjoyable."*
>
> *"I think the contents of the training were quite beneficial, thank you for everything."*

The social network shares were also positive after the training. Additionally, the comments of the visitors on the social media accounts suggested that more training should be provided.

Considerations Regarding the Products and Views After Training

First, the sample scenarios and application studies given by the participants were investigated and the following dimensions were observed:

- The complete emergence of the scenario as a whole
- Developed within the framework of objectives
- Originality
- The cooperation studies in the process of development
- Adaptability features

In this respect, observations were made regarding the participants' development and active involvement in the process of sample scenario development. All participants conducted effective implementation of the mentioned dimensions. Some examples were given by the participants during the training on computational thinking without computers are summarized below:

> *"Guessing 5 different geometrical shapes to form a given shape or vice versa; forming patterns/objects out of given geometrical shapes."*
>
> *"Trying to escape from a restricted area with limited movement, the car park escape games in computers."*
>
> *"Story examples in which the next step changes according to the answer given."*
>
> *"Binary number systems can be taught easier with the help of enjoyable visuals, burning candles, open bridges and so on."*
>
> *"Many approaches and problems in graph theory can be gamified with 10-people groups (forming a minimum spanning tree, the shortest/longest way algorithms and so on.)."*
>
> *"Nim game is a game that teaches single and double parity control."*

In addition to affectivity analysis during the scenario production phase, the participants' views were gathered with a survey form. The frequency values obtained with the survey form are given in Table 7.

Table 7. Percentages and Frequencies Regarding the Participants' Evaluation of Training

Complementary expression	**1**		**2**		**3**		**4**		**5**	
	%	**f**	**%**	**f**	**%**	**f**	**%**	**f**	**%**	**f**
It enabled the computational thinking concept to become clear in my mind	0	0	3	1	6	2	22	7	69	22
It positively affected my point of view to the computational thinking applications	0	0	0	0	6	2	19	6	75	24
It included many important clues regarding the scientists in the field and their applications	0	0	3	1	6	2	25	8	66	21
It contributed to the increase in my level of knowledge about basic software to be used in computer education for children	0	0	0	0	9	3	25	8	66	21
It supported gaining new points of view regarding programming language education without a computer	0	0	0	0	9	3	19	6	72	23
It helped me learn the basics about the robotic programming application	0	0	0	0	6	2	19	6	75	24
It was a beneficial training process that will help me maintain the computer teaching profession more effectively.	0	0	0	0	3	1	16	5	81	26

Table 7 indicates the participants stated the training content was positive. In addition to the multiple choice questions in the survey, open-ended questions were also included to evaluate the training. Sample participant opinions are given below:

> *"The level of my knowledge has increased, the training process could be kept longer, especially to see a little more detail of the software activities. Integration of the applications into a sample lesson plan could be experienced."*
>
> *"I would like to thank everyone who contributed to this training, the contents of this training can easily be implemented on all levels except for the secondary school level. Meetings can be set as discussion groups, different scenarios can be discussed, and they can be developed through brainstorming activities. I will be happy and proud of being a part of this."*
>
> *"The activities not only contributed in terms of computational thinking but also changed my view towards my profession, thank you."*
>
> *"It was a quite successful, beneficial, and interesting training."*
>
> *"In general, the computational thinking was explained well on the first day. In the second day, two programs were taught, but the time was limited so it was fast. The activities without computers on the third day were for early age groups rather than secondary school levels. It would be more enjoyable if it targeted all age groups. Robotics had many contributions. The pedagogical part on the final day was good in terms of integration.*
>
> *"The training was quite successful, it broadened my horizons, and I gained new information. I am quite satisfied. However, placing the two software trainings on the same day made it difficult to adapt."*
>
> *"It was quite successful in terms of involving both activities out of the ICT classroom and computer-based activities."*
>
> *"The software used, and the applications were things that I saw for the first time, the program gave me ideas. In my institution, I will use the activities without computers frequently, in this respect, my horizon was broadened."*

According to the opinions presented, participants viewed the training contents positively, the expectations regarding broadening available techniques were met, and enhancing computer science activities without computers was achieved.

The "Information and Communication Technologies" course objectives' relationship with the training program using an objective rubric was examined. The relationship of the training with the objectives was scored between 1 *("What We Have Learned has No Effect") - 5 ("What We Have Learned is Very Effective").* The collected data is presented in Table 8.

Table 8. The Participants' Opinions Regarding the Relationship of Secondary School Information and Communication Course Objectives with the Training Program Provided.

Objective Expression	**1**		**2**		**3**		**4**		**5**	
	%	**f**	**%**	**f**	**%**	**f**	**%**	**f**	**%**	**f**
The students will be able to explain the ease provided by object-based programming languages.	0	0	0	0	13	4	44	14	44	14
The students will be able to use variables to express the information and operate on them.	0	0	0	0	16	5	34	11	47	15
The students will be able to understand that in order to carry out a task, functions are formed using variables and constants.	0	0	3	1	6	2	41	13	50	16
The students will be able to explain the concept of object.	0	0	3	1	6	2	47	15	47	15
The students will be able to program objects to fulfill a series of actions.	0	0	0	0	0	0	53	17	47	15
The students will be able to form a series of objects based on the classes.	0	0	0	0	6	2	53	17	41	13
The students will be able to explain the concept of class.	0	0	0	0	9	3	53	17	38	12
The students will be able to use the cycles in the repeating group in the program.	0	0	0	0	9	3	31	10	59	19
The students will be able to test the program to obtain the targeted outcome.	0	0	0	0	6	2	22	7	72	23

Table 8 shows participants think that the training contents are largely related to the "Secondary School Information and Communication Technologies Course" programming unit objectives. This finding is as an important outcome towards achieving the targeted course goals and participant proficiency.

Discussion

Given the results of the study, first, it can be mentioned that the computer science teachers' feedback on the importance of integrating activities improving computational thinking skills of the individuals into the CS education process was in parallel with Barger's (2008) computer literacy definition's writing emphasis. Additionally, it has been seen that Neil (1986)'s in-service training approach based on 4 dimensions detailed in the literature section can be used to design effective in-service training activities towards CS teacher. The results support the findings of Doukakis et al. (2013) on CS teachers' need to develop their knowledge of integrating technology in their courses and teaching strategies for the algorithms. The satisfaction and positive academic growth of the participants showed that face to face training approach can be used effectively, an e-leaning platform with the same content can be developed in order to examine the effectiveness of two different approaches as seen in the study of Alkaria and Alhassan (2017). On the other hand, in accordance with Papert's (1980) computational thinking approach, the positive change in the opinions of computer science teachers who participated in product-oriented robot development studies towards their courses were observed in their statements. The observations and data gathered from the evaluations of the teachers showed that in line with the inferences in Benitti's (2012) study, robotic applications' positively affected the participant's motivation. With the dimension of the qualification of the training process, the computer science teachers who participated in the study confirmed the content compatibility with MoE's (2017) programming languages objective in the curriculum. The study, which was started and conducted with the aim of supporting secondary school teachers in programming education, received positive feedback due to its structure integrating technological and pedagogical dimensions. The Harel and Papert's (1991) and Kafai and Resnick's (1996) studies complement placement of product design and development processes in the forefront in computer science education. This approach was adopted in the in-service education of computer science teachers. The use of social media platforms created under the scope of this study helped disseminate the positive outcomes achieved through in-service education. Through the project study conducted, the importance of researching pedagogical and method based aspects of the computer science education processes, which are mostly considered in their technological dimensions, was emphasized. The in-service education model created and the findings will contribute to enhancing the effectiveness of in-service education for computer science teachers.

Conclusion

In the current study, computer science education was investigated with a holistic approach. The in-service training model for computer science teachers was implemented with three different groups. During the in-service training, the views and evaluations in the literature towards the effectiveness of computer science education were considered to develop the contents of the training. In general, participants indicated that this training will positively affect their own personal development. In the future studies, the comparisons can be carried out among the effectiveness of e-learning and face to face training strategies including the same content.

References

Ackermann, E. K. (2006, September). Piaget's constructivism, Papert's constructionism: What's the difference? Retrieved from http://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf

Alkaria, A., & Alhassan, R. (2017). The effect of in-service training of computer science teachers on scratch programming language skills using an electronic learning platform on programming skills and the attitudes towards teaching programming. *Journal of Education and Training Studies*, *5*(11), 1-12. doi: 10.11114/jets.v5i11.2608

Arslan, H., & Sahin, I. (2013). Bilisim teknolojileri ögretmenlerinin hizmetici egitim kurslarına yonelik gorusleri [Information technology teachers' opinions about in-service training courses]. Middle Eastern & African Journal of Educational *Research*, (5), 56-66.

Barger, R. N. (2008). *Computer ethics: A case-based approach*. New York, NY: Cambridge University Press.

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education, 58*(3), 978-988. doi: 10.1016/j.compedu.2011.10.006

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum, *Computers & Education*, *72*, 145-157. doi: 10.1016/j.compedu.2013.10.020

Brennan, K. (2015). Beyond technocentrism: Supporting constructionism in the classroom. *Constructivist Foundations*, *10*(3), 289-296.

Corral, J. M. R., Balcells, A. C., Estevez, A. M., Moreno, G. J., & Ramos, M. J. F. (2014). A game-based approach to the teaching of object-oriented programming languages. *Computers & Education, 73*, 83-92. doi: 10.1016/j.compedu.2013.12.013

Computer Science Education Research Group (CSERG). (2017, August 10). CS unplugged activities. Retrieved from http://csunplugged.org/activities/

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, *60*(6), 33-39. doi: 10.1145/2998438

Doukakis S., Koilias C., Adamopoulos N., Giannopoulou P. (2013) Computer Science Teachers' In-service Training Needs and Their Technological Pedagogical Content Knowledge. In M. D. Lytras, D. Ruan, R. D.Tennyson, P. Ordonez De Pablos, F. J. García Penalvo, & L. Rusu (Eds.), *Information Systems, E-learning, and Knowledge Management Research* (pp.311-316). Berlin, Germany: Springer.

Erdem, A. R., & Simsek, S. (2013). Ogretmenlere ve okul yoneticilerine verilen hizmet ici eğitimlerin irdelenmesi [Investigating in-service training given to teachers and school heads]. *Usak University Journal of Social Sciences,* (16), 94-108.

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

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38-43. doi: 10.3102/0013189X12463051

Harel, I. E., & Papert, S. E. (1991). *Constructionism*. Santa Barbara, CA: Praeger.

Howland, K., & Good, J. (2015). Learning to communicate computationally with Flip: A bi-modal programming language for game creation, *Computers & Education, 80*, 224-240, doi: 10.1016/j.compedu.2014.08.014

ICILS (2013). *The international computer and information literacy study*. Retrieved from http://ec.europa.eu/education/library/study/2014/ec-icils_en.pdf

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: a need assessment analyses. *TOJET: The Turkish Online Journal of Educational Technology*, *9*(2), 125-131.

Kafai, Y. B., & Resnick, M. (Eds.) (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Abingdo, UK: Routledge.

Kert, S. B. , Kayak, S., Erkoc, M. F., & Avincan, K. (2014). *Kodu ile Kendi Oyununu Gelistiren Cocuklar* [Children who develop their own game with code]. Paper presented at the 8th International Computer & Instructional Technologies Symposium, Edirne, Turkey.

Kordaki, M. (2010). A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation. *Computers & Education*, *54*(1), 69-87. doi: 10.1016/j.compedu.2009.07.012

Lau, W. W. F., & Yuen, A. H. K. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, *57*(1), 1202-1213. doi: 10.1016/j.compedu.2011.01.002

MoE (Ministry of Education) (2012, January 17). *Ortaokul ve imam hatip ortaokulu bilisim teknolojileri ve yazilim dersi (5, 6, 7 ve 8. Sınıflar) ogretim programi* [Secondary school and imam hatip secondary school information technologies and software course (Grades 5, 6, 7 and 8)]. Retrieved from http://ttkb.meb.gov.tr/program2.aspx?islem=2&kno=196

Navarrete, C. C. (2013). Creative thinking in digital game design and development: A case study, *Computers & Education*, *69*, 320-331. doi: 10.1016/j.compedu.2013.07.025.

Neil, R. (1986). Current models and approaches to in-service teacher education. *Journal of In-Service Education*, *12*(2), 58-67. doi: 10.1080/0305763860120202

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Pereira, H. B. D. B., Zebende, G. F., & Moret, M. A. (2010). Learning computer programming: Implementing a fractal in a Turing Machine. *Computers & Education*, *55(2),* 767-776. doi: 10.1016/j.compedu.2010.03.009.

Raud, N., & Orehhova, O. (2017). In-service training of teachers of English as a foreign language in Estonia: Mapping of trends and opportunities. *Problems of Education in the 21st Century*, *75*(2), 194-203.

Shaw, R. S. (2012). A study of the relationships among learning styles, participation types, and performance in programming language learning supported by online forums. *Computers & Education*, *58(1),* 111-120. doi: 10.1016/j.compedu.2011.08.013.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. doi: 10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A, 366*(1881), 3717–3725. doi: 10.1098/rsta.2008.0118

Wing, J. M. (2010). Research notebook: Computational thinking—What and why? *The Link.* Retrieved from https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Wing, J. M. (2016). Computational thinking 10 years later. *Microsoft Research Blog.* Retrieved from
https://blogs.msdn.microsoft.com/msr_er/2016/03/23/computational-thinking-10-years-later/