Informatics in Education, 2019, Vol. 18, No. 1, 131–150

DOI: 10.15388/infedu.2019.06

A Holistic Approach for Computer Science Education in Secondary Schools

Serhat Bahadır KERT[1], Filiz KALELİOĞLU[2], Yasemin GÜLBAHAR[3]
[1]*Yıldız Technical University, Faculty of Education, Computer Education & Instructional Technologies, Istanbul, Turkey*
[2]*Başkent University, Faculty of Education Baglıca Campus Ankara, Turkey*
[3]*Ankara University, Faculty of Educational Sciences Cebeci Campus Ankara, Turkey*
e-mail: sbkert@gmail.com, filizk@baskent.edu.tr, ysmnglbhr@gmail.com

Received: November 2018

Abstract. In this study, effectiveness of a computer science course at the secondary school level is investigated through a holistic approach addressing the dimensions of instructional content design, development, implementation and evaluation framed according to ADDIE instructional design model where evaluation part constituted the research process for the current study. The process has initiated when the computer science curriculum had major revisions in order to provide in-service teachers with necessary support and guidance. The study is carried through as a project, which lasted more than one year and both quantitative and qualitative measures were used through a sequential explanatory method approach. The intention was to investigate the whole process in detail in order to reveal the effectiveness of the process and the products. In this regard, not only teachers' perceptions but also students' developments in their perceptions of academic achievement and computational thinking, as well as correlations between the computational thinking sub-factors were investigated. The findings showed that the instructional materials and activities developed within the scope of the study, positively affected the computational thinking and academic achievement of students aged 10 and 12 years old. The teachers' weekly feedbacks regarding application structures and implementation processes were also supported the findings and revealed some more details that will be useful both for instructional designers and teachers.

Keywords: computer science education, computational thinking, content development

Introduction

Although it has been less than a century since the invention of computer technologies and the Internet, the infusion and influence of these technologies in daily life has been extraordinarily rapid. This innovation and change process has also had a significant effect on human life; having changed our ways of socialisation, learning, communication and entertainment. In addition, it is necessary to play constant catch-up as these technologies are also employed in producing and applying technology at speed. Thus, it is crucial to educate individuals who will be able to keep up with the change processes entailed in following the global innovations, and this process starts from scratch at school. Parallel to this, some countries have begun to develop curricula that cover current computer programming topics for the education of new generations. The fundamental philosophy of this new trend is focused on the more efficient use of technology, problem solving and product development (ISTE, 2017).

In today's age, it has become important to educate individuals who can reach the information that they require, interpret the information that they access, add to it, and then disseminate their knowledge to others. In other words, learning is a process that continues throughout life. Today, distance education programmes have emerged as new educational platforms that can contribute to lifelong learning, where people can reach information without limitation of time and space, and can continue to work outside of their place of employment or study. Rapid developments in information technologies have also triggered developments in mobile technology systems, which also affect mobile programming and application development. The new generation should not only be able to use these technologies adequately well, but they should also be trained as individuals who can design and develop both for and with these technologies. This can only be achieved through the provision of certain qualifications to young people.

As traditional learning methods and environments are largely inadequate for today's students, it has become inevitable for educational institutions to support traditional teaching programmes with technological innovations. The worldwide digital transformation and the realisation of economic growth, the welfare of citizens, and the creation of a digital economic strategy seem to be prerequisites to the development of digital skills. Therefore, a meaningful relationship has begun to be established between national development plans and education policies, and computer science education in particular.

Consequentially, some countries (Turkey, Austria, Czech Republic, Denmark, Finland, France, Greece, Hungary, Italy, Lithuania, Poland, Portugal and Switzerland) have started to offer computer science education from kindergarten right through until the end of high school, either electively or as part of students' compulsory education (Bocconi, Chioccariello, Dettori, Ferrari, & Engelhardt, 2016). According to the same report, it was revealed that there are two main reasons for countries to conduct studies on updating their curricula. The first is to help students to develop their knowledge of computational thinking skills in different ways, to express themselves using technology, and to solve real life problems from different angles. And the other reason is to support economic growth, to fulfil job openings in the field of information and communication technologies, and to acquire knowledge of computational thinking in order to prepare students for the professions of the future. When examined in more detail, it was observed that 13 different countries included in the report had added IT skills to their curricula in support of logical thinking and problem-solving skills, to direct students towards computer science, to teach computational thinking and programming skills, to increase employment in areas of information and communication technologies, and in support of other important competencies.

When the curricula of different countries are examined, it is noteworthy that the United Kingdom is first in this field. Informative thinking and programming topics have been taught at the primary and secondary school levels since 2014. In September 2016, France starting updating studies in the curriculum, forming a structure on teaching the knowledge of computational thinking across all age levels. The basic principles of algorithms and programming, the use of programming languages, as well as digital citizenship were all discussed, and algorithms and programming teaching viewed as language learning necessary for modern day communication. In 2016, Finland began to study algorithmic thinking and programming within compulsory courses from an interdisciplinary teaching approach to elementary school. In Poland, computer science and informatics topics have been taught for a considerable time within compulsory courses for all ages. The computer science subject in the Polish curriculum has been updated to be more comprehensive, with pilot studies in September 2016 followed by rollout in September 2017 as a compulsory subject. In Denmark, information and communication technologies are taught with an interdisciplinary approach at the first and middle level, with problem-solving and logical thinking skills taught within a very limited scope. The compulsory course was aimed to be offered to 10th and 11th grade students in 2017. In Norway, they also teach computational thinking and programming in an updated curriculum as an elective course, following being piloted in 143 schools.

In Turkey, the Information Technology and Software Course focused on 'problem solving and programming' when it was updated in 2012, and was also aimed to provide students with knowledge, skills and attitudes about literacy and digital citizenship. In other words, computer science topics have been included in the curriculum since 2012. In 2017, the curriculum was updated with new additions, and with more programming objectives. The course has been updated to be a two-hour compulsory course for middle school Grades 5 and 6, and as an optional course for Grades 7 and 8. In the high school curriculum, which was updated in 2016, ethics, problem solving and algorithms, programming (Python), robotics programming, and web and mobile programming were added to teaching at different levels (Gülbahar & Kalelioğlu, 2018).

In recent years, there has been noteworthy growth in the efforts of the researchers on developing effective educational activities implemented in CS education. Especially at K-12 level, majority of these are practiced without using computers. The main objective of an unplugged CS activity is to make student noticed that computers are not the only content of CS and it is also required to use mathematical thinking skills (Taub, Armoni and Ben-Ari, 2012). Cortina (2015) states that unplugged activities provide students towards to make collaborative studies and support them to use their creativity and problem solving skills. Kineasthetic practices, role playing studies, puzzles, artistic activities, games and mysteries can be listed as different types of unplugged CS activities (Curzon, McOwen, DonoHue, Wright & Marsh, 2018). In the literature, there can be seen different studies on investigating the effectiveness of unplugged CS activities. In their studies, Jiang and Wong (2017), found that unplugged CS activities positively affect the motivation of primary school students on learning computational thinking. Similarly, Taub, Armoni and Ben-Ari (2012) found that unplugged CS activities improved the understanding of 5th grade students on the meaning of CS. Improving the computational thinking skills of the students is one of the important goals of today's CS curricula. Unplugged CS activities can also be effectively used to be able to reach this goal in education settings (Curzon, McOwen, Plant & Meagher, 2014). Given the perspectives presented in the literature, all of the teaching processes in this study were composed with unplugged CS activities. The effectiveness of instructional design and the CS teachers' opinions on the learning activities were investigated.

In summary, evolving with today's technological changes, computer science curricula have been changed and implemented effectively in order to graduate a sufficiently qualified and knowledgeable workforce for the 21st century in emerging dimensions of educational technology as mentioned by the Horizon report (Becker et al., 2017) since there will be a need of well-educated human resources who will work on the aforementioned jobs which requires different skills but computer science. However, just revising or developing the curriculum is not enough for effective implementation and spread of the new content. Most of the teachers are in need for professional development, showing a resistance to welcome new topics and innovative applications into their classrooms. Thus, providing teachers with the necessary instructional support, which will guide the teaching process in their classroom, plays an important role in the implementation of the curriculum. Owing to these facts, the aim of the current research study is to shed light on the process of implementing curricula changes by exploring the effect of developed materials and learning activities on students' performances and skills by considering the perspectives of both teachers and students as stakeholders.

Methodology

Research Design

This research study employs a mixed method approach in order to investigate the effectiveness of educational materials and activities used in computer science education for 5th grade students from a pedagogical perspective. The course is compulsory, lasts 18 weeks and has to be provided two hours a week. The first semester topics include information and communication technologies whereas second semester focuses on problem solving and programming. Mixed methods research designs use both quantitative and qualitative approaches within a single research project in order to gather or analyse data (Creswell, 2003; Creswell & Plano Clark, 2007; Miles & Huberman, 1994; Tashakkori & Teddlie, 2003). Using mixed methods is useful when the exploration of a phenomenon and testing of a new intervention is required. Mixed method research overcomes the inherent weaknesses of a single method design by complementing the qualitative and quantitative methods. Thus, sequential explanatory method was chosen for the current study since the phenomenon to be investigated requires a depth of detail in order to reveal the effectiveness of the whole learning process. In sequential design, one type of

Table 1
Method of the study

Phase I *Content Design & Development*	Phase II *Pre-Implementation*	Phase III *Implementation*	Phase IV *Post-Implementation*
Design and development process of educational materials and activities based on needs analysis results	***(Pre-Test)*** Achievement Test for Academic Performance (quantitative data)	***(Mid-Test)*** Academic Performance(quantitative data)	***(Post-Test)*** Achievement Test for Academic Performance (quantitative data)
	(Pre-Test) *Self-Efficacy Perception Scale for Computational ThinkingSkills* (quantitative data)	Teacher Diary – weekly (qualitative data)	***(Post-Test)*** *Self-Efficacy Perception Scale for Computational Thinking Skills* (quantitative data)

data provides the basis for the collection of another type of data (Mertens, 2005, p. 292). Hence for this current research study, qualitative results have been used to assist in explaining and interpreting the findings of a quantitative study. Each phase of the study are summarised in Table 1.

As can be seen in Table 1, in order to provide different perspectives to the phenomenon, two data sources, students and teachers, considered for this research. Students' performance was gathered through quantitative measures whereas teachers' perceptions were gathered through qualitative measures. Hence, current study is conducted in order to answer the following research questions:

1. What was the effect of educational materials and activities developed for computer science course in terms of students' academic performance and computational thinking skills? *(Quantitative data)*
2. What were the teachers' perceptions about learning activities and teaching process after implementation in class? *(Qualitative data)*

Instructional Design Process of the Project

Based on the recent revisions in computer science curriculum, Google and the Ministry of National Education (MoNE) of Turkey have signed a protocol for the development of educational materials and activities for students to be implemented by teachers, supporting the spreading of effective teaching of the computer science course. According to this protocol, a project team were in charge where three academicians and four teachers contributed as experts, besides two instructional, two graphic designers and a project coordinator.

The team members incorporated ADDIE (Analysis, Design, Development, Implementation, and Evaluation) model, which is a framework for almost all other Instructional System Design models, by incorporating the five steps of the model during the project (Aisami, 2009; Floyd & Shambaugh, 2017), where all elements of designed are covered (Morrison, Ross, Kalman, and Kemp, 2012). For the analysis phase, the team is focused on the learning outcomes, the features of the participants, delivery options for instructional content, and pedagogical aspects, together with the potential challenges. Learning objectives of the spring semester focusing on problem solving and programming were presented in Appendix A (`http://bit.do/Ap_A`).

Most of the time is spent on analysing the learners (Lasky, 2018); their age group, physical capabilities, and their cognitive pre-qualifications as well as the technical infrastructure and access opportunities to technology. Computer Science (CS) education course content was aimed to be prepared for secondary school students aged 10 to 12 years old. In the literature, children in this age group are named as 'early adolescents' (Rambaree, 2015). During this period, psychological, physical, social and emotional changes occur in the individuals' characteristics. Mobility, interaction, collaboration and meaningful learning can be listed among the key factors in the education of early adolescents. Given these developmental features of the student group, it was considered appropriate to go beyond the traditional instructional approach to CS, and hence, the content developed included gamification and unplugged activities (drama, worksheets, games etc.), attractive stories and daily life problems, collaborative design studies and productive computer science practices.

As not all schools have the same standards in terms of student numbers, computer hardware, accessibility and Internet infrastructure, it was obvious that the country has diverse technological opportunities with some schools having a computer laboratory and Internet access, while some schools lacking of these opportunities and having only a smartboard in their classrooms. Thus, it is decided to deliver at least two alternative content and activities for these diverse groups. That is why, both unplugged activities in addition to digital materials were provided to students in order to maintain equity for all in learning computer science concepts. It is also decided to group instructional content and activities as 'Teacher Guide' since the overall mission of the project was to provide all computer science teachers effective content to teach in any circumstance.

The verification of expected outputs of instruction and determining the relevant teaching and assessment tools are also decided in the design phase (Branch, 2009). After reaching an agreement about all pedagogical concerns, content development started with continuous control and revision in the development phase. 'Content Development' process took about three months, where all educational materials and activities like presentations, posters, worksheets, unplugged activities, game-based activities, tools and technologies to be integrated to learning process were developed. The academicians contributed to the production process as both instructional designers and content experts, as well as supervising the computer science teachers by providing feedback during the design process. Secondary school students were the target group of the activities, therefore, the computer science teachers played a significant role because of their experience teaching children at early ages. The graphic designer drew informative graphics for use with the content, and prepared animations for the activities. The proofreading of the whole text was performed by the language expert as the final step before printing. Additionally, communication among the stakeholders and organisation of meetings was provided by the project manager. The content design and development process has been graphically presented as Fig. 1.

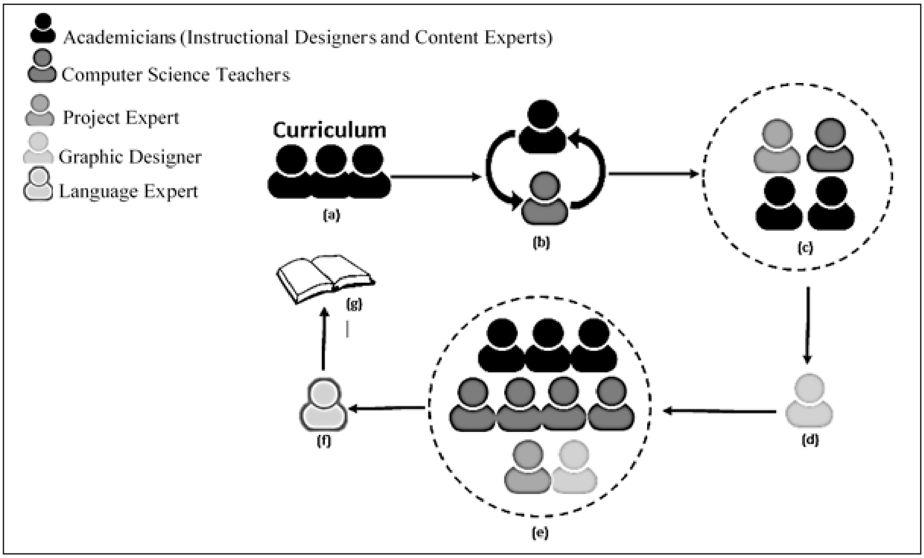

Fig. 1. Design and Development Process for Educational Materials and Activities.

The design and development process for each activity was conducted in seven steps. Firstly, objectives written in the curriculum were reviewed by the academicians (Fig. 1-a). Then, each computer science teacher designed content under the supervision of an academician (Fig. 1-b). This was the most extensive step with a period of design. The academician would provide feedback, and then revisions would be applied by the teacher – this process loop was continuous until a satisfactory product was agreed. The content that was separately designed by the academician was reviewed by teacher groups in local project meetings (Fig. 1-c), and then sent to the graphic designer for production of the required graphics (Fig. 1-d). After gaining approval of the local project group, the whole content was reviewed in a general meeting (Fig. 1-e). Then the language expert checked for spelling problems and the final version of the content was sent for printing (Fig. 1-f). A sample of the developed activities is presented in Fig. 2.

Each activity page begins with an introductory section (see Fig. 2-a), which includes the proposed duration to complete the activity, targeted objectives, keywords, and a list of required materials. Graphical representations are important components of the activities and are used to motivate students and to show details as to the required knowledge. As can be seen in the activity named ‘A Cake Recipe’ (see Fig. 2-b), the text and related images are used together in order to explain the structure of the flowchart and to create a connection between the theoretical information and a real-life practice. Additionally, knowledge base (see Fig. 2-c) were prepared to support the computer science teachers during the concept teaching process. All the tasks and activities agreed totally by the whole team in terms of meeting the related learning outcomes. In order to give detailed information about the material, sample pages from the activity book was presented in appendix F (`http://bit.do/Ap_F`).

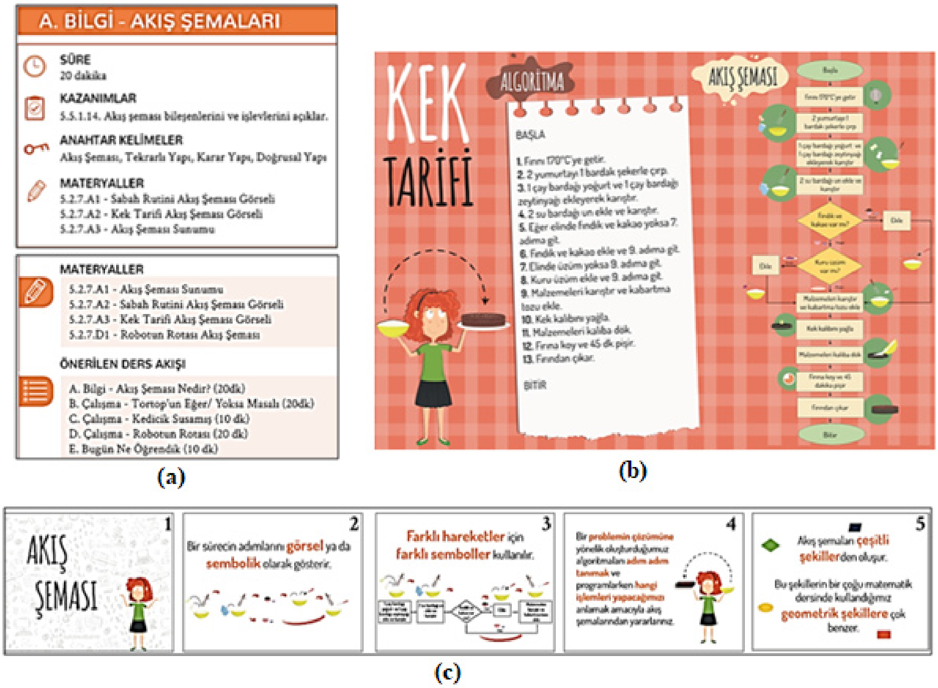

Fig. 2. Sample of Activity for Computer Science Education.

After completion of all development work, it is required to implement the proposed instructional approach in order to reveal effectiveness of both teaching approaches and educational materials and activities. Implementation phase, included both formative evaluation so as to carry out possible revisions to the design, and a summative evaluation in order to obtain data on the efficiency of the instruction (Gustafson & Branch, 2002). Hence, this research study is focused on the implementation and evaluation processes of this project, where the data collection instruments were designed and developed in parallel with the instructional design process.

Sample of the Study

Volunteer teachers and their students from different cities implemented the learning activities, and the relevant data was collected as planned. A total of 162 secondary school students aged 10 and 12 years old participated in the quantitative part of the study. Taub, Armoni and Ben-Ari (2012) emphasize that the age group of middle-school students is crucial on CS education, children begin to have ideas about their future during these ages. Additionally, Hur, Andrzejewski and Marghitu (2017) indicate that girls prefer on a CS career in the future is highly-related their self-confidence in CS education. Therefore, in this study, it was wanted to give details about the participants in this study. All 162 students participated in the pre-test and post-test academic performance measurements

Table 2
Gender-age cross table of the participants

		Age			Total
		10	11	12	
Female	Count	5	50	1	56
	% within Gender	8.9%	89.3%	1.8%	100.0%
	% within Age	41.7%	62.5%	11.1%	55.4%
	% of Total	5.0%	49.5%	1.0%	55.4%
Male	Count	7	30	8	45
	% within Gender	15.6%	66.7%	17.8%	100.0%
	% within Age	58.3%	37.5%	88.9%	44.6%
	% of Total	6.9%	29.7%	7.9%	44.6%
Total	Count	12	80	9	101
	% within Gender	11.9%	79.2%	8.9%	100.0%
	% within Age	100.0%	100.0%	100.0%	100.0%
	% of Total	11.9%	79.2%	8.9%	100.0%

and the midterm exam, which was set once during the term. Pre-test–post-test data regarding the Computational Thinking Scale were obtained from a total of 101 students. The gender-age distribution of the students is presented in Table 2.

Weekly feedback regarding the applications developed was gathered from 15 secondary school computer science teachers and the contents obtained from these teachers were analysed by means of quantitative data analysis.

Research Design Process of the Project

Data Collection Tools

Data collection tools were selected in accordance with the pedagogical outcomes of the course. The quantitative data was mainly collected through application of an Achievement Test for Academic Performance (APPENDIX E – `http://bit.do/Ap_E`), a Self-Efficacy Perception Scale for Computational Thinking Skills (SPSCTS) (APPENDIX B – `http://bit.do/Ap_B`), and a Midterm Exam (APPENDIX C – `http://bit.do/Ap_C`). The Achievement Test for Academic Performance consisted of 18 questions that measured the participants' algorithmic thinking skills. Item and reliability analysis of the test was performed by Doğan and Kert (2016), with difficulties of the items ranging between .30 and .65 and an average discrimination value of .47. The test was prepared by the researchers in order to measure the students' growth in academic achievement and proposed as a sample assessment tool to computer science teachers. Data about the computational skills of the students were gathered through application of the SPSCTS scale, which was originally developed by the researchers. The Cronbach's Alpha (α) value of the scale was found to be .943 (Gülbahar, Kert, & Kalelioğlu, 2018), with the scale consisting of five factors and a total of 36 questions (see Table 3).

Table 3
Factors of the Self-Efficacy Perception Scale for Computational Thinking Skills (SPSCTS)

Factors	Number of Items	Cronbach's Alpha (α) Value
Efficacy for Algorithm Design	9	.930
Efficacy for Problem Solving	10	.880
Efficacy for Data Processing	7	.856
Efficacy for Basic Programming	5	.838
Efficacy for Self-Confidence	5	.762
Total	36	.943

As a reliable and valid scale for the 10–14 year age group, the SPSCTS scale was chosen for the current study in order to reveal students' perceived skills about computational thinking. A midterm exam was performed at the halfway point in the semester, and the obtained data was not used in the pre-test or post-test comparisons.

Furthermore, a weekly evaluation questionnaire (APPENDIX D – `http://bit.do/Ap_D`) was developed by the researchers. For the construct validity, the opinions of three experts in the field of Educational Technology were sought. Following each activity, the teachers completed a questionnaire consisting of nine closed items and seven open-ended questions. The closed item list consisted of three Likert-type questions that focused on the scope of the text, the activities' duration, flow plan, pre-implementation notes, effectiveness of the sections (e.g., introduction, development and conclusion), validity of the keywords used to depict the activities, visual design and the materials' appeal to students. The open-ended questions focused on the students' reactions to the lesson, problems encountered in the learning environment, their favourite and least-favoured aspects of the activities, any changes suggested to the application of the activities, and the level of attainment of the learning achievements. All data collection tools were added as appendices of the study.

Data Analysis

Descriptive statistics and factor analysis used to analyse the data collected from three tests, namely Achievement test for academic performance, Midterm test for academic performance, and Self-Efficacy perception scale for computational thinking skills as quantitative measures, whereas inductive coding were used to analyse weekly diaries of teachers as feedback for qualitative measures.

Findings

The Effect of Educational Materials and Activities Based on Academic Achievement and Computational Thinking Skills

As the first step, a pre-test and post-test comparison for the Algorithmic Thinking scores of the participants was investigated. Prior to the analysis, the normal distribution of the

Table 4
Algorithmic Thinking Scale Paired Sample *t*-Test results

Scale	*Groups*	*N*	$\overline{X}$	*SD*	*SE*	*t-test*		
						t	*df*	*p*
Algorithmic thinking	Pre-test	162	83.58	32.48	2.55	2.552	161	.012
	Post-test	162	90.06	31.45	2.47			

data was tested with the Shapiro-Wilk test ($p < .05$), and therefore, a Paired Sample *t*-Test as a parametric statistic was used to compare the pre-test and post-test scores. The summary of the statistics is shown in Table 4.

Since the *p*-value was .012 ($p < .05$) and the mean of the post-test scores was greater than the mean of pre-test scores, it can be stated that there was a significant difference in favour of the post-test algorithmic thinking test scores. This shows that the activities prepared by the project group positively affected the algorithmic thinking skills of the students. In order to examine the effect of the educational process on the computational thinking skills of the participants, the Self-Efficacy Perception Scale for Computational Thinking Skills (SPSCTS) was applied. The assumption of the normality was confirmed ($p < .05$) and Paired Sample *t*-Test statistics were used to compare the pre-test and post-test SPSCTS scores for all sub-factors. The results are presented in Table 5.

As can be seen in Table 5, the results showed a positive improvement in total and sub-factor SPSCTS scores of the participants ($p < .05$). Additionally, for the intra-group comparison of pre-test–post-test data, the variance among the overall scale and sub-factor scores were investigated using Pearson correlation analysis. The analysis results are presented in Table 6.

Table 5
SPSCTS Paired Sample *t*-Test results

Factors	*Tests*	*N*	$\overline{X}$	*SD*	*SE*	*t-test*		
						t	*df*	*p*
Whole SCALE	*Pre-Test*	101	75.75	14.66	1.45	-11.79	100	.000
	Post –Test	101	94.20	12.58	1.25			
Efficacy for Algorithm Design *(Factor 1)*	*Pre-Test*	101	13.22	4.95	.49	-17.36	100	.000
	Post-Test	101	23.17	4.01	.39			
Efficacy for Problem Solving *(Factor 2)*	*Pre-Test*	101	24.05	4.95	.49	-4.83	100	.000
	Post-Test	101	26.55	3.71	.36			
Efficacy for Data Processing *(Factor 3)*	*Pre-Test*	101	16.72	3.67	.36	-4.94	100	.000
	Post-Test	101	18.57	2.52	.25			
Efficacy for Basic Programming *(Factor 4)*	*Pre-Test*	101	9.06	3.14	.31	-11.12	100	.000
	Post-Test	101	12.65	2.41	.24			
Efficacy for Self-Confidence *(Factor 5)*	*Pre-Test*	101	12.67	2.38	.23	-2.21	100	.010
	Post-Test	101	13.24	1.93	.19			

Table 6

Correlation among different improvement levels of computational thinking perception factors

Variable	SPSCTS Total	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
SPSCTS Total	1					
Factor 1	.697**	1				
Factor 2	.842**	.343**	1			
Factor 3	.834**	.352**	.743**	1		
Factor 4	.769**	.416**	.589**	.632**	1	
Factor 5	.658**	.289**	.530**	.546**	.394**	1

** Correlation significant at the .05 level (2-tailed)

As can be seen in Table 6, it is notable that there was a significant relationship between each factor score and the overall scale score. A high-level significant relationship was found between the variances of Factors 2 (Problem-solving skill self-efficacy perception), Factor 3 (Data processing skill self-efficacy perception), and Factor 4 (Basic programming skill self-efficacy perception) and the variance of the overall scale score (.842: $p < .05$; .834: $p < .05$; .769: $p < .05$).

In addition to the purposeful measurement tools used in the comparison analyses, a midterm exam was developed as an evaluation tool for computer science teachers. The midterm exam consisted of 12 multiple-choice questions from different skill areas considering the objectives targeted through the activities designed. Each of the proposed questions were designed with a scenario from real-life problems in an attempt to create a connection between the content of computer science education and the physical world in the students' minds.

The questions were designed to evaluate different sub-skills of computational thinking. Three different field experts' opinions were consulted in order to determine the question structures and the sub-skill areas of each question. The percentages of distribution of the sub-skills are shown in Fig. 3.

The exam results were only used to gain an idea as to the educational growth of the students during the activities, and to propose an assessment tool for computer science education at the secondary school level. The students' academic success was distributed as shown in Fig. 4.

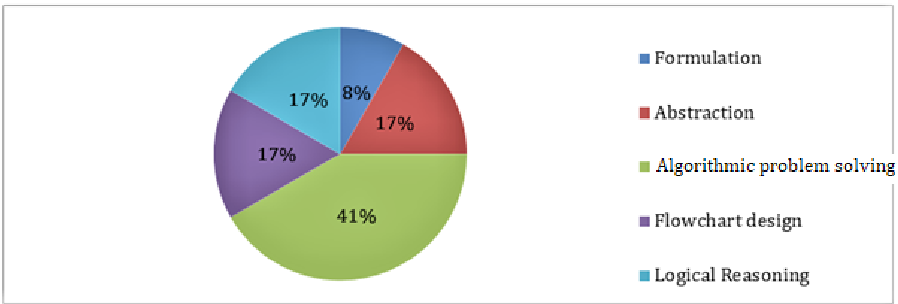

Fig. 3. Distribution of skills targeted by midterm exam questions.

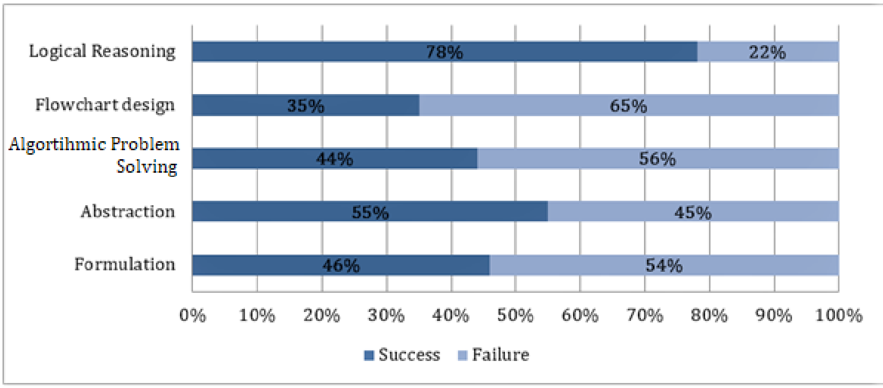

Fig. 5. Distribution of academic achievement of students in the midterm exam.

When the midterm exam results were investigated, it was determined that the students were particularly successful in the questions on logical reasoning. However, the general classroom success was found to be low in the flowcharts topic. The data obtained from the midterm exam were not used for comparative analyses since the exam was only carried out once. The data was used to conduct a pilot study on the suitability of the evaluation tool for the target age group, and to obtain an understanding as to the midterm level of the students' academic development.

Teachers' Perceptions about Learning Activities and Teaching Process

Teachers are requested to fill diaries weekly after each course in order to get timely feedback. The findings revealed a positive insight about both the process and the product. Almost all of the average of items was quite good (Table 7). The time allocated for the activities was sufficient ($X = 2.37$, $SD = 56$). Another noteworthy aspect of the teacher evaluations made during the 14 week implementation was that the teachers liked the pre-

Table 7

Weekly feedback provided by the teachers

Items	Mean	SD
1. Scope of the text given for oral expression was sufficient	2.91	.14
2. Time allocated for activities was sufficient	2.37	.56
3. Suggested course flow was well planned	2.75	.32
4. Pre-application notes are very important in terms of course preparation	2.97	.12
5. Lesson planning with introduction, development and consequences facilitated the application	2.96	.12
6. Determination of keywords for the activity facilitated concepts teaching	2.80	.27
7. Materials would appeal to students	2.90	.14
8. Materials have no visual design problems	2.96	.09
9. The weekly lesson plan helped me to describe the course as smooth	2.86	.24

application notes (X = 2.97, SD = 0.12). In parallel, the attractiveness of the materials (X = 2.96, SD = .09) and the planning of the lesson (X = .96, SD = .12) were the other highest averages.

Students' Reaction for the Activities

The majority of teachers stated that the students were excited and willing in the events, but that some students showed a low interest because they did not understand the topic. One teacher who gave an opinion on this subject said, 'The students listened to the entry with interest. They were very excited about the activities, but they said that most of the class did not understand the questions'. Another teacher stated that, 'The students had a lot of fun in the wolf-lamb and Hanoi tower activities, and the motivation and attendance in the group work were very positive'.

Favourite Aspects for Lesson Plans

Teachers generally appreciated and liked the activities, that the materials were cuttable and feasible, that the visuals were fun, and that the course flow was led by the teacher. In this regard, one teacher said,

> *I liked the activities that led students to think. When we look at the classes in general, I think that such activities do not take place very much. I believe that it will be useful for students to gain different perspectives and to improve their thinking skills.*

Another teacher said, 'The examples were really beautiful. While directing and talking about example questions, they realised that they could solve many problems in their lives without realising it'.

Least-favoured Aspects of the Lesson Plans

In general, the long duration of some activities emerged as a least-favoured aspect as not all the activities could be completed. In this regard, one teacher said that, 'Working with the working paper in the first phase distributed the interest of the children'. Another teacher expressed that, 'The number of questions could be reduced or those that are most influential could be dealt with', and another stated that, 'My students had difficulties to solve them and wasted a lot of time. I needed to direct each group individually'. Lastly, related to the duration of the activities, one teacher stated that, 'There was no least-favoured aspect of the lesson plans, I just felt sorry that I could not manage to complete all the activities'.

Changes Made while Applying Activities

In general, it was observed that the teachers were trying to do all the recommended activities in the course flow. However, they stated that they were unable to implement all of the activities because there was insufficient time to complete some of the activities. In this regard, one teacher said that, 'I applied all the activities, but the students did not understand the problem-solving steps, so I've shortened the time when I gave them the

Hanoi towers activity'. Some teachers stated that they tried to use the time effectively by lecturing themselves in general using the interactive board instead of doing hands-on activities. Some teachers stated that they extended the period for the activities and some teachers continued with activities not completed from the previous week.

Level of Contribution to Learning by Students
Often, the teachers thought that the students achieved the learning outcomes. Teachers who stated that they experienced problems in this matter said that some students did not understand some of the concepts, and they explained that they needed more time in this regard. Some issues were not seen to be in line with the level of 5th Grade students; however, there was no problem with this issue reported for 6th Grade students.

Suggestions for Changes to the Activities
In this context, while the teachers expressed that the activities were sufficient in general, some teachers suggested that the activities could be simplified a little bit, that the time for completion of the activities should be changed, and that there should be black and white pictures for ease of printing. The addition of images or videos for some subjects was also put forward as a suggested change.

Discussion and Conclusion

As previously mentioned, many countries have or are in the process of updating their curricula for computer science education, and many reports have stressed the importance of the need for a larger number of individuals who are experienced in IT and software/hardware in the coming years. In this context, computer science education was included in the curriculum in Turkey in 2011, and revised in 2016. However, the inclusion of the computer science curriculum has not always led to the effective implementation of this curriculum by all teachers, and has not ensured that all students receive adequate and effective computer science training. Therefore, it is important to support the implementation of curriculum by delivering necessary guidance through delivery of instructional materials and activities.

Some teachers feel that they cannot effectively provide such computer science training because of their perceived inadequacy of knowledge, or due to lacking infrastructure of some schools. However, one of the most important components in making this process effective is teaching methods, activities and materials. With this awareness, as a result of a protocol agreement between the Turkish Ministry of National Education and Google, different computer science education activities have been designed based on the curriculum objectives in order to teach both the objectives and computational thinking sub-components to students. This protocol concluded the creation of teacher guides for grades 5 and 6, along with guidelines on how to apply the activities and materials which is currently continuing with the grades 1–4. Therefore, in order to assess the effectiveness of these activities, a pilot evaluation of computer science training was conducted. The activities related to the computer science training were evaluated with different teachers on the basis of voluntary participation at different public and private schools. In general, it has been seen that the activities contributed to the academic success of students and to the development of their computational thinking skills.

For the quantitative results, there was a considerable increase in the algorithmic thinking test scores of the students. Moreover, there was a significant relationship found between Problem-solving skill self-efficacy perceptions, this finding supports the Data processing skill self-efficacy perceptions and Basic programming skill self-efficacy perceptions. Activities focusing on teaching programming with developed CS activities positively affected the students' algorithmic thinking skills and their self-efficacy related to problem-solving, data processing and basic programming skill. This result can be explained with the motivational improvement of the students towards CS education. With this regard, it can be mentioned that the findings support the findings of Jian and Wong (2017) on the positive motivational effects of unplugged CS activities. The use of teaching methods such as drama, guided discovery, hands-on practicing, group projects, discussion, questioning and the computer science activities prepared to focus on programming and algorithm presented to the students helped them in gaining these skills. Supporting this fact, Salomon and Perkins (1987) confirmed that programming education can develop the cognitive skills of students when instruction is well-designed. Prepared in a holistic manner with material and methodological support, this current study's training series presents an authentic guide for teachers and may serve as leverage in the acquisition of such skills. Supporting this, Hromkovič, Kohn, Komm, and Serafini (2016) stated that, 'programming education is a great opportunity to teach important core concepts of computer science on various levels and to establish algorithmic thinking as part of a broad and general education' (p. 122).

When the exam results were evaluated, it was seen that the students mostly succeeded in questions related to logical reasoning, abstraction, and formulation. This may have arisen because students came across these issues for the first time and were unaccustomed to such evaluation questions. However, in the coming years, the comparison of these results with other examinations they take may provide a better interpretation. The fact that the students were successful in logical inquiry, abstraction, and formulating is seen as a significant measure of success. This finding of the study provides the perspective of Cortina (2015) on the potential of unplugged CS activities towards to support the improvement of creativity and problem solving skills of the students. Additionally, it can be said that developed unplugged activities can be used to build up computational thinking skills. Curzon, McOwen, Plant and Meagher, (2014) emphasized the same point in their research. Similarly, Taub, Armoni and Ben-Ari (2012) found that unplugged Cs activities could be used to support students to be able to understand the meaning of CS. This development at the conceptual level may indicate that the students experience problems at the point of transferring to the algorithm. It could be considered that there is a need for more practice on these issues. Considering the success seen in these skills and the positive change in perceptions, it could be anticipated that they will be more successful if they gain some further experience in designing algorithms.

When the results of the teachers' evaluations of their weekly feedback were examined, it showed that the teachers liked the activities and were benefited from them. Except for the time allocated for activities, the evaluation of the teachers regarding the scope of the activities, pre-class preparation notes, lesson plans and the quality of the materials were the most positive. This result supports the change of perception seen in the quantitative findings that the activities were liked by the students, and were especially excited about the activities. Having visual support in the form of the materials, presentation of course contents in a fun way, and ease of use of the activities made it easy for the teachers and the students to have welcomed them. As teaching a class is a living, dynamic process, it is a very natural to encounter problems when applying new activities within the classroom. The students' level of readiness for the subject may have varied and the difficulty of the subject for the students may also have affected their issues with the time required in reality for each activity. The occurrence of classroom management problems could also be another factor that affected follow-up and completion of the activities. Finding that the students achieved the course objectives and were successful in their learning process was a very satisfactory result. The fact that some subjects were perceived as difficult for 5th grade students should be seen as a potential reason behind such a situation, and should therefore be taken into account in the evaluation of the results of the examination.

Other problems observed in the current were technical infrastructure issues and teachers' in-service training needs. Following this research, a platform was created and presented to the teachers with an online digital lesson on teaching computational thinking that was originally developed by Google as the Computational Thinking Course and translated into Turkish. For meeting the future needs of future computer science teachers, the curriculum of education faculties was also reviewed and revised according to the expected knowledge and skills from the teachers. However, it seems that despite all these works and efforts, it may take considerable time for the results in Turkey to increase significantly and spread across the country. Hence, studies, regulations, reviews and revisions should continue throughout this process.

These steps taken and the practices offered in the current study are considered to still be very new to Turkish education. The country is still at the beginning of computer science education; yet, research should continue in many different dimensions such as how efforts affect student performance, and what contributions will facilitate to gaining computational thinking skills. It is therefore necessary to investigate whether or not computer science should be taught as a separate course, at what age should it be taught, and whether or not it should be taught along with disciplines such as mathematics in order to reveal which approach is more beneficiary for students in terms of reaching expected learning goals of the curriculum. Therefore, it is important that researchers construct and share their own practices in light of the findings contained in the current study. At this point, teachers also need to undertake research in their own classrooms and to share the results at academic conferences as the burden of this process should not fall solely to academia.

Although this research study is limited with the volunteer teachers and their students participated in the study, the overall process together with the developed products contributed to the computer science education in many ways. Teachers became aware of the fact that computational thinking and computer science concepts can be taught without use of computers via various approaches. They found the opportunity of application of different teaching methods and gained valuable experience. Hence, given all the findings obtained from the quantitative and qualitative analyses, it could be stated that the project presented a detailed roadmap for effective and efficient implementation of computer science education. It is believed that the results encourage educators, especially at the secondary education level, to be able to create multidirectional and meaningful computer science educational environments and to contribute to computer science education literacy towards emerging well-structured educational efforts.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Acknowledgment

The authors would like to thank GOOGLE TR for supporting computer science education in Turkey.

References

Aisami, R.S. (2009). Planning and teaching online courses. In: C. Howard, J. Boettecher, L. Justice, K. Schenk, P. Rogers, G. Berg (Eds), *Encyclopedia of Distance Learning (2nd ed.)*. Hershey, PA/London, UK: IGI Global, 1628–1638.

Becker, S.A., Cummins, M., Davis, A., Freeman, A., Hall, C.G., Ananthanarayanan, V. (2017). *NMC Horizon Report: 2017 Higher Education Edition.* Austin, Texas: The New Media Consortium.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education-Implications for policy and practice (No. JRC104188)*. Seville, Spain: European Commission Joint Research Centre.

Branch, R.M. (2009). *Instructional Design: The ADDIE Approach* (Vol. 722). Springer Science & Business Media.

Cortina, T.J. (2015). Reaching a broader population of students through unplugged activities. *Communications of the ACM*, 58(3), 25–27.

Creswell, J. (2003). *Research Design: Qualitative, Quantitative and Mixed Methods Approaches*. Thousand Oaks CA: Sage.

Creswell, J., Plano Clark, V. (2007). *Designing and Conducting Mixed Methods Research*. Thousand Oaks CA: Sage.

Curzon, P., McOwan, P.W., Plant, N., Meagher, L.R. (2014, November). Introducing teachers to computational thinking using unplugged storytelling. In: *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*. ACM, 89–92.

Curzon, P., McOwan, P.W., Donohue, J., Wright, S., Marsh, W. (2018). Teaching of Concepts. *Computer Science Education: Perspectives on Teaching and Learning in School*, 91.

Doğan, U., Kert, S.B. (2016). Bilgisayar Oyunu Geliştirme Sürecinin, Ortaokul Öğrencilerinin Eleştirel Düşünme Becerilerine ve Algoritma Başarılarına Etkisi. *Boğaziçi Üniversitesi Eğitim Dergisi,* 33(2), 21–42.

Floyd, K.K., Shambaugh, N. (2017). Instructional Design for Simulations in Special Education Virtual Learning Spaces. In: T. Kidd & L.R. Morris, Jr. (Eds.), *Handbook of Research on Instructional Systems and Educational Technology*. IGI Global, 202–215.

Gustafson, K.L., Branch, R.M. (2002). What is instructional design. In R.A. Reiser, J.V. Dempsey (Eds.), *Trends and Issues in Instructional Design and Technology*. Merrill/Prentice Hall, 16–25.

Gülbahar, Y., Kalelioğlu, F. (2018). Bilişim Teknolojileri ve Bilgisayar Bilimi: Öğretim Programı Güncelleme Süreci. *Millî Eğitim Dergisi,* 47(217), 5–23. Retrieved from `https://dhgm.meb.gov.tr/yayimlar/dergiler/Milli_Egitim_Dergisi/217.pdf`

Gülbahar, Y., Kert, S.B., Kalelioğlu, F. (2018). The Self-Efficacy Perception Scale for Computational Thinking Skill (SPSCTS): Validity and reliability stud*y. Turkish Journal of Computer and Mathematics Education*. Advance online publication. doi:10.16949/turkbilmat.385097

Hromkovič, J., Kohn, T., Komm, D., Serafini, G. (2016). Examples of algorithmic thinking in programming education. *Olympiads in Informatics*, 10(1–2), 111–124.

Hur, J.W., Andrzejewski, C.E., Marghitu, D. (2017). Girls and computer science: experiences, perceptions, and career aspirations. *Computer Science Education*, 27(2), 100–120.

International Society for Technology in Education [ISTE]. (2017). *ISTE Standarts for Educators*. Retrieved January 11th, 2018 from `https://www.iste.org/standards/for-educators`

Jiang, S., Wong, G.K. (2017). Assessing primary school students' intrinsic motivation of computational thinking. In: *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. IEEE, 469–474.

Lasky, J. (2018). *ADDIE Model.* Salem Press Encyclopedia. Retrieved from `http://search.ebscohost.com/login.aspx?direct=true&db=ers&AN=129815287&lang=tr&site=eds-li`

Mertens, D. (2005). *Research and Evaluation in Education and Psychology: Integrating Diversity with Quantitative, Qualitative, and Mixed Methods* (2nd ed.). Boston: Sage.

Miles, M., Huberman, A. (1994). *Qualitative Data Analysis: An Expanded Sourcebook* (2nd ed.). Thousand Oaks CA: Sage.

Morrison, G.R., Ross, S.M., Kalman, H.K., Kemp, J.E. (2012). *Designing Effective Instruction.* Wiley.

Rambaree, K. (2015). The Ecology of Early Adolescents' Internet Mediated Sexual Interactions. In: D.B.A. Mehdi Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology* (3rd ed.). IGI Global, 6781–6790.

Salomon, G., Perkins, D.N. (1987). Transfer of cognitive skills from programming: When and how? *Journal of Educational Computing Research*, 3(2), 149–169.

Taub, R., Armoni, M., Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions on Computing Education (TOCE)*, 12(2), 8.

Tashakkori, A., Teddlie, C. (Eds.). (2003). *Handbook of Mixed Methods in Social & Behavioral Research*. Thousand Oaks CA: Sage.

S.B. Kert is Associate Professor and the chair of the department of Computer Education and Instructional Technology (CEIT) at Yildiz Technical University. His research interests are computer science education for children, methodologies in programming language education, computational thinking, game-based education and development of e-learning environments. His PhD is in CEIT from Anadolu University. During his post-doctoral study at 2011–2012, he had a chance to study with Dr. Jianwei ZHANG who is known, in international scientific community, with his remarkable studies on pedagogical basis of knowledge building. That was an NSF project titled "Fostering Collective Progress in Online Discourse for Sustained Knowledge Building". He contributed to the process by preparing story boards of "Idea Thread Map (ITM)" which has been used for analysing discourses on a timeline. He teaches course at both undergraduate and graduate levels and goes on to conduct national and international research on the pedagogical features of computer science education.

F. Kalelioglu is an Associate Professor of Computer Education and Instructional Technology at the Education Faculty, Baskent University. She graduated with a PhD in Computer Education and Instructional Technology from Ankara University in 2011. Her academic interest areas are e-learning, social media in education, instructional design, technology integration and computer science. She serves as a reviewer for several journals in the field of educational technology. She has published many national and international articles and book chapters.

Y. Gülbahar has got her BS degree from Department of Mathematics of Science Faculty at Middle East Technical University (METU) in 1992. Same year she started working as a programmer at METU Computer Center. Then, in 1998, she became a research assistant to the Department of Computer Education and Instructional Technologies (CEIT) in the Faculty of Education, METU while studying her MS degree at the same department. She earned his MS degree in the field of Science Education at METU Graduate School of Science in 1999 and she received her PhD in Department of Computer Education and Instructional Technologies (CEIT) from Graduate School of Sciences in 2002. After, she worked for Başkent University Faculty of Education Department of Computer Education and Instructional Technologies (CEIT) for about 9 years. Between years 2011– 2017 she worked in Ankara University as the head of Department of Informatics and meanwhile served as the Chairman of Distance Education Center. Since 2018 she is a faculty member of Faculty of Educational Sciences in the Department of Computer Education and Instructional Technologies (CEIT). Dr. Gulbahar has got her Associate Professor degree in 2009 and full Professor Degree in 2014.

Yasemin Gulbahar has lectured on many topics such as programming languages, problem solving and algorithms, instructional technologies, instructional design, material design and development, distance learning, web design, measurement and evaluation, research methods, teaching methods, software development, technology integration and planning both in undergraduate and graduate level. She has also many national and international publications as books, book chapters, journal articles and proceeding papers.