Ilkogretim Online - Elementary Education Online, 2020; 19 (3): pp. 1781-1796
http://ilkogretim-online.org.tr
doi:10.17051/ilkonline.2020.735161

Teaching computing at secondary school level: Understanding teachers' experiences and challenges

Filiz Kalelioğlu, *Baskent University,* Turkey, *filizk@baskent.edu.tr* ORCID: 0000-0002-7729-5674
Yasemin Gülbahar, *Ankara University,* Turkey, *gulbahar@ankara.edu.tr* ORCID: 0000-0002-1726-3224
Serhat Bahadır Kert, *Yıldız Technical University,* Turkey, *sbkert@yildiz.edu.tr* ORCID: 0000-0002-1093-6326

Abstract. The purpose of the study is to evaluate the efficiency of the computer science teachers' guide prepared for the 5th Grade Computing Course in Turkey. Experiences of 111 teachers regarding the implementation process were gathered through a survey. The results revealed that the 5th Grade Teachers' Guide to Teach Computing' is a supportive material for the delivery of CS education. Most of the teachers found the activities and general framework of the lessons to be useful. Despite most of the opinions being positive, on some points, the teachers considerably criticized the activities contained within the guide. It can be said that the results of the study offer a roadmap to practitioners on how to develop an effective guide for CS teachers, and how to create effective learning activities and materials for students.

Keywords: Computer science education, instructional materials, instructional resources

Received: 30.04.2019	Accepted: 19.09.2019	Published: 15.06.2020

INTRODUCTION

In recent years, there has been a notable worldwide increase in individuals' interest in computer science (CS) education. Effective CS teaching strategies, technologies and pedagogies have been investigated by a number of researchers (Blikstein et al., 2016; Caspersen, 2018; Chen & Rea, 2018; Guzdial, 2016; Mason & Cooper, 2012; Shim, Kwon, & Lee, 2017). The origin of this trend can be attributed to two reasons. First, the needs of modern societies to produce their own technologies have increased based on the sheer demand to keep up with the rapid advances in information technologies. Hence, the new CS professionals such as programmers, network analysts, and web developers are in demand in many countries. This can be explained as an economic reason why global efforts are being made towards finding new ways to improve the efficiency of CS education at all levels, from kindergarten right through to university. However, the second and more important base to this trend relates to the interdisciplinary outcomes of learning to code. Software development requires the application of different thinking skills, from logical reasoning to abstraction.

In other words, learning programming languages supports students in improving their computational thinking (CT) skills, which can be used to solve problems faced in daily life, besides being part of the skillset required in the production of modern technological solutions and products. CT is, therefore, not only relevant for computer scientists as it includes the understanding of problem solving, systems design, and human behaviour through the application of computational concepts (Wing, 2006). With a pedagogical transformation from 'learning to code' to 'code for learning', it has been emphasised in the literature that coding is a key skill for all children to learn, and that students having CT skills are better able to reach solutions for problems in any discipline (Denning, 2017; Kafai & Burke, 2014). CT, as a new term, has played a crucial role in this pedagogical revolution. This new aspect has been the cause of radical changes to computing curricula at the K-12 level. Many countries have begun to develop curricula that cover current computer programming topics for the education of the next generation. The fundamental philosophy of this new trend is to focus on the more efficient usage of technology for problem solving and for product development.

Existing Computer Science Education Curricula

When the computing curricula of a number of countries (Turkey, Austria, Czech Republic, Denmark, Finland, France, Greece, Hungary, Italy, Lithuania, Poland, Portugal, and Switzerland) were examined, computer science topics were seen to be taught as either compulsory or elective courses, and from kindergarten right through to high school graduation (Bocconi, Chioccariello, Dettori, Ferrari, & Engelhardt, 2016). When examined in more depth, it was observed that 13 different countries have added computational thinking skills to their curricula in order to support logical thinking and problem-solving skills, to direct students towards the computer sciences, teach coding and programming skills, to increase students' subsequent employment in the area of information and communication technologies, as well as to support other key competencies.

When the curricula of different countries are examined, it is notable that the UK was the first country in this field, with computational thinking and programming topics taught in primary and secondary level classes since 2014. In September 2016, France starting updating curriculum studies, forming a structure for teaching computational thinking to all school age levels. The basic principles of algorithms and coding, the use of programming languages, as well as digital citizenship are included in French curricula. Similarly, in 2016, Finland began to study algorithmic thinking and programming within compulsory courses, ranging from interdisciplinary teaching approach to elementary school instruction. In Poland, computer science and informatics topics have been taught for a long time as compulsory courses for all ages. The computer science subject in the curriculum was updated to be more comprehensive, with pilot studies having begun in September 2016, and becoming a compulsory subject in September 2017. Danish information and communication technologies are taught through an interdisciplinary approach at first and middle level schools, and problem-solving and logical thinking skills are taught with a very narrow scope. The compulsory course is delivered to grades 10-11 as from 2017. In Norway, computational thinking and programming is taught according to an updated curriculum as an elective course, and is being piloted in 143 schools.

Similarly, Webb et al. (2017) analysed several curriculum specifications and found scope for Computer Science topics as programming/algorithms, data representation, digital infrastructure, digital applications, human factors, and related ethical issues. They found less agreement regarding the importance of more general intellectual practices and social competences such as cooperation, collaboration and communication (p. 60). Moreover, Webb et al. (2018) examined curriculum development of six countries (Australia, Israel, New Zealand [NZ], Poland, Slovakia, and the UK) and found that computer science education is seen as a core subject that all students should learn from elementary school upwards.

Turkey has integrated problem solving and programming concepts into its existing curricula since 2012. A standards-based curricula was developed for this purpose with four standards and various age-appropriate teaching levels.

1. Information and Communication Technologies (ICT) Literacy
2. Communication, Knowledge Sharing, and Self-reporting Through ICT Usage
3. Researching, Knowledge Construction, and Collaborative Working
4. Problem Solving, Programming, and Product Development (Gülbahar & Kalelioğlu, 2018).

The curricula was revised to a traditional curriculum in 2017, with slight revisions to the content and with precise learning outcomes added for each grade. The reason for the change was stated that the standards-based curriculum could not be adopted by CS teachers and that so many variations and levels in learning outcomes had been revealed. The revised curriculum is composed of five themes:

1. Information Technologies
2. Ethics and Security
3. Communication, Research, and Collaboration
4. Product Development
5. Problem Solving and Programming (Gülbahar & Kalelioğlu, 2018).

The first three themes are for the teaching of ICT concepts for effective usage, and the last two are for the teaching of coding and programming with a baseline of computational thinking in the early stages. With a cross-curricular review, the enhancement of learners' computational thinking skills is a common aim found within various educational processes. In the literature, it can be seen that indicators showing the growth of learners' CT skills have been investigated by researchers. Brennan and Resnick (2012) proposed a three-dimensional framework based on the concept knowledge, practice skills, and perspectives of learners in order to evaluate the CT skills improvement. From this point of view, Kong (2016) proposed the following principles towards designing effective CT curricula at the K-12 level:

- It is crucial to follow a top-down strategy in order to make it easier for learners to resolve complex computational problems.
- The efficiency of the project development process depends on the learners' comprehension of essential knowledge.
- Learning activities should be designed in accordance with learners' interests in order for creativity to be supported.
- The development of CT skills can be achieved through the designing of complex computational CT tasks.
- It is important to review each level of the curricula to ensure all essential CT knowledge is covered.
- Interest-driven task design, appropriate assessment criteria, and motivating learners by staging the products are three key factors of an efficient CT curricula.

These principles present a pathway for instructional designers preparing materials for CS education.

Instructional Materials for Computer Science Education

Pedagogical qualifies of Instructional materials are essential for providing the efficiency of an educational curriculum. Learners' performances, in different dimensions, mostly depend on the variety of the materials used in the educational process (Lin et al, 2014). If a material is not designed as attractive, goal-oriented, activity-based and easily understandable then it affects the educational motivation of the learner negatively. Concrete content like educational modules, lesson plans or the explanations about the activities is generally called as materials (D'Angelo et al., 2017). In this context; graphical design, structural design, activity design or scenario design can be listed as phases of an instructional material design process. In the literature, there can be seen different aspects of designing and using principles of them. Clark and Mayer (2016), in their multimedia design theory, proposes a framework for instructional content. Some items of it can be listed so: It's better using words and graphics together rather than using words alone, words and related graphics must be placed close to each other, redundant knowledge and unrelated content with the pedagogical content must be omitted from the course, complex content can be segmented into small parts to make it plain and clear. Additionally, providing the balance between the challenge and skills of the learners in educational activities is an important educational design process. In the flow theory, if skills of the individual exceed the presented challenge then boredom is inevitable, on the other hand, if the presented challenge exceeds the skills then anxiety would be seen (Nakamura & Csikszentmihalyi, 2009). If the target group is composed of the children then it is crucial to embed the "fun" in the content. Papert (2002), emphasizes this point by using "hard fun" as a new concept for educational environments. In order to create a learner-centered environment, the content must include connections to funny experiences from the life of the learner. In a comprehensive manner, Lashley (2019), recommends that instructional materials must be durable, safe, compatible with different individual features, easily understandable, integrated with the educational interactions and reshaped easily for different pedagogical targets.

In CS education, the constructionist perspective has an important role in the shape of educational design. It can be explained as a conscious knowledge building process which is done in a context, it can be a simple task or a complex one (Papert & Harel, 1991). The main idea is to present opportunities for learners to enhance their thinking skills by generating new ideas during the educational process. Hence, using rich interactions and activities are important to design instructional materials for CS education. Supporting learners to create their mental model as early as possible is the key factor of success in CS courses, therefore, using qualified educational tools is essential to reduce cognitive load (Margulieux, Guzdial & Catrambone, 2012). In most CS courses the focal point is productive problem solving, teaching specific coding content or creating the project (Pears' et al, 2007). In such a comprehensive process, the effectiveness of instructional material can be provided only with CS teachers' contributions as experienced practitioners. Their professional competencies directly affect the efficiency of their students' learning. Teachers are expected to have subject matter knowledge, content knowledge, pedagogical knowledge, and pedagogical content knowledge (Shulman, 1986). With the perspective of CS education, keeping up to date with innovative tools, content, and pedagogies is crucial to the success of CS teachers. The questions of 'Why will I teach programming?', 'What will I teach?', 'How will I teach?', and 'What are the learning difficulties?' must be addressed in order to achieve pedagogical content knowledge regarding programming education (Saeli, Perrenet, Jochems, & Zwaneveld, 2011). Educational material prepared for CS education must be composed of the activities improving the efficiency of the instruction and support teachers in finding solutions to problems related to the pedagogy of the process. Using semi-structured interviews, Yadav, Gretter, Hambrusch, and Sands (2016) investigated the challenges faced by CS teachers. One of their findings showed that teachers have problems meeting students' pedagogical and content needs. In another study, Cutts, Robertson, Donaldson, and O'Donnell (2017) recommended, based on the data obtained from a teacher training project, that it is important to encourage CS teachers to learn new teaching techniques and to address gaps in their conceptual knowledge.

In terms of the radical changes to CS curricula, it can be seen that teachers' guides play an important role as a supportive educational tool as they include technological knowledge, activities and pedagogical suggestions aimed at CS teachers. Within this context, the purpose of the current study is to evaluate the efficiency of the CS teachers' guide prepared for a 5th Grade Computing Course in Turkey.

METHODOLOGY

Based on the recent curriculum of a computing course, Google and the Ministry of National Education of Turkey signed a protocol for the development of instructional materials and activities for students to be implemented by teachers. According to the protocol, a working group was established which includes three academicians and four computer education teachers as experts, plus two instructional designers, two graphic designers, and one project coordinator. The group members have experience in terms of other countries' curricula, digital materials for teaching computing and coding concepts, and pedagogical approaches that may be incorporated in delivering the content.

The aim of the current study is to evaluate the end product from the aforementioned working group, namely the instructional materials and activities for the teaching of coding, which was published in the form of a 'Teachers' Guide to Teach Computing'. In order to evaluate the end-to-end instructional process, including the materials and activities, a 'survey approach' was chosen for the study. The single guiding research question for the study is, 'To what extent is the Teachers' Guide to Teach Computing' effective based on the opinions of teachers?'

Pedagogical Design of Instructional Activities

Student-centred instruction was adopted for the design process of the learning activities. In-class interactions, active participation of the students, and gamification are some of the key features of the teaching. In order to improve the conceptual knowledge of students, presentations were prepared as supportive educational materials. In this context, the following types of the activities can be listed:

- Game-based activities
- Paper-based activities
- In-class discussions

Unplugged CS Activities

- Block-based programming activities
- Other computer-based activities (word processing etc.)

CS Activities

The activities are divided as two types, Unplugged CS Activities and CS Activities. Considering the differences among the schools' technical infrastructures, each activity was prepared with an alternative unplugged/CS variant of that activity. Some schools do not have computer laboratories but have interactive smartboards in each classroom. Therefore, alternative activities were also generated for classrooms having only interactive boards. The 'File Management' activity, can be used as a sample of this approach (see Figure 1).

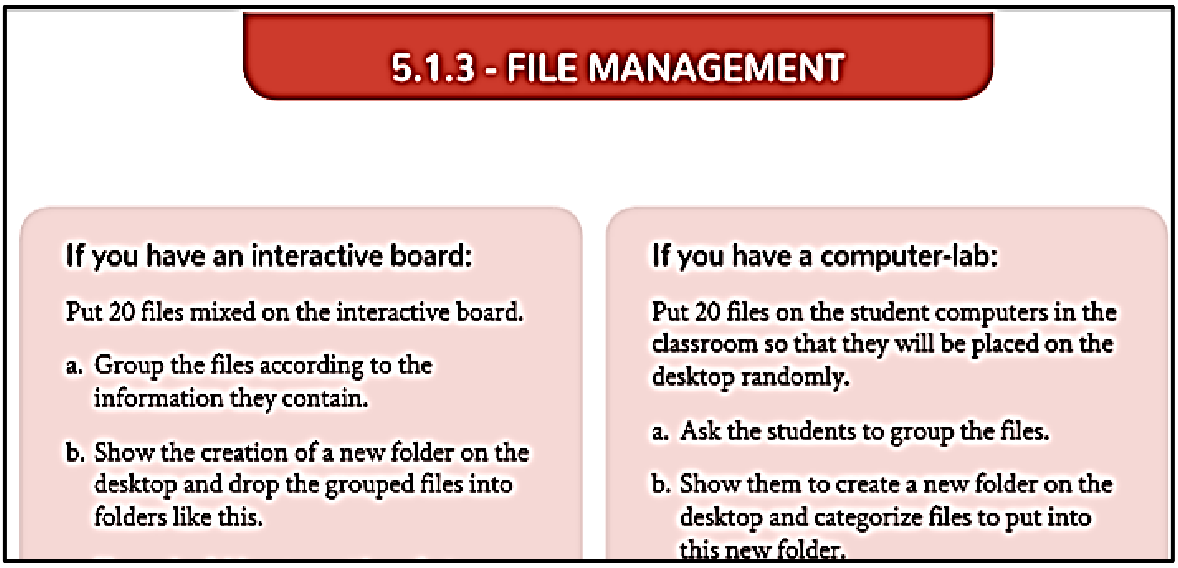

FIGURE 1. *Alternative activities for different infrastructures*

Game-based activities were especially used for teaching abstract knowledge related to information technologies in a humorous way. The 'Giant for Input, Dwarf for Output! Game Scenario' can be used as a sample of such an activity (see Figure 2).

PRACTICE

Giant for Input, Dwarf for Output! Game Scenario

1. Explain that some of the tools are used for data input, some for data output and some for both tasks. Tell them that you will play a Giant-Dwarf game in which they will identify which tool is to be used for what purpose by considering the jobs they carry out.

Game rules will be as follows:

Open the "Data I / O Game" presentation and display the tools in this section respectively; if the displayed tool is an input tool, ask the students to stand up and be a Giant like in Figure 1.

Figure 1

Figure 2

If the displayed tool is a hardware for **output**, ask the students to sit down and be a Dwarf like in Figure 2.

If a device such as memory USB that can be used both for **input and output** appears on the screen, ask the students to lay over the desk as in Figure 3.

Those students who fail to act in accordance with the tool displayed on the screen, is out of the game.. When the last device is shown on the screen, you can cheer for the students who won the game and proceed to the introduction of the input / output units.

FIGURE 2. *Game-based activity*

The working mechanism of the computers was taught by using paper-based activities (see Figure 3). In-class discussions are used and play an important role in creating self-consciousness among students with regards to abstract concepts such as ethical values.

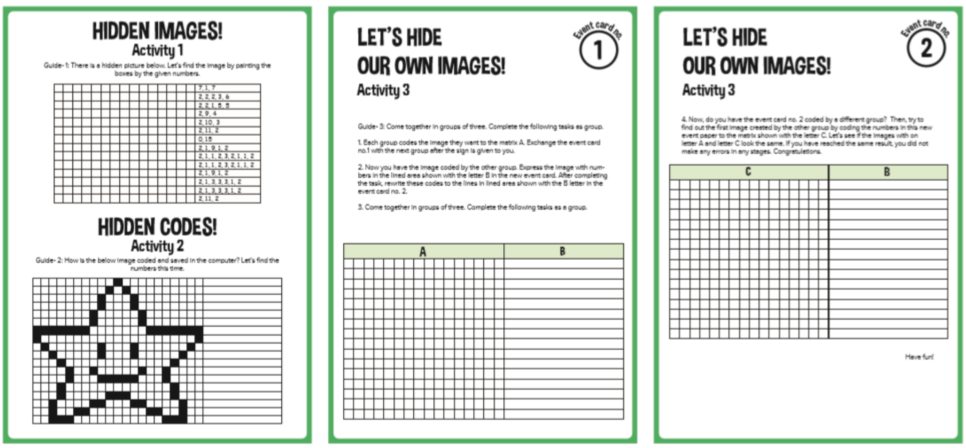

FIGURE 3. *Paper-based activities*

Additionally, Scratch and Blockly games were the two platforms used as block-based programming environments. In order to support the teachers towards providing efficient activities, some 'Notes' giving clues about the implementation were added next to the text. Time was an important factor for the courses. At the secondary education level in Turkey, each course hour lasts 40 minutes; therefore, all of the activities were prepared as to be completed by students within 40 minutes.

Instructional Design Process

The working group, as a first step, investigated the instructional objectives and decided upon the duration of each objective within the weekly two-hour computing course. As a second step, a face-to-face brainstorming session was conducted as a two-day workshop in order to reveal and discuss possible teaching scenarios and activities, where each day the instructional objectives regarding the whole semester were covered. Then, the teachers were given appropriate time to write their lesson plans and to draft student handouts and activities. Monitoring and giving feedback during this process was communicated online through the Google Docs environment. After approval of the ideas and the flow of the lesson plans, the graphic designers worked on preparation of the digital materials. The working group held a final two-day workshop in order to finalise and make recommendations for changes or corrections as deemed necessary for each case. The final product was sent to the Turkey's Ministry of National Education for approval, and the necessary revisions were applied based on their feedback prior to publication of the 'Teachers' Guide to Teach Computing' on the official website of the Ministry of National Education.

The teachers' guide was prepared within a project group that included the authors of the current study. All of the activities designed in the guide are appropriate to outcomes of the new secondary school CS curriculum in Turkey. In terms of instructional design, the phases of the ADDIE (analysis, design, development, implementation and evaluation) model were followed during the design process. ADDIE is based on the core characteristics of an instructional system design process. Most of the instructional design models have been built on the structure of ADDIE. In the analysis phase, all components of the instruction such as learners, content, goals and objectives, and resources were analysed and the problem(s) of the process identified. Moreover, the working group members searched all the possible existing digital materials available on the Internet, and also checked the curricula and implementations of other countries. Hodell (2010) explained these phases as a process of answering the questions of 'what, who, where, by whom, why and when' (p. 3).

The definition of the content and delivery mediums were undertaken during the design phase (Lior, 2013). In the design phase, after the brainstorming session and agreeing on the definition of the content and delivery mediums (Lior, 2013), the teachers created the lesson plans and the academicians controlled and provided feedback prior to the final version being agreed. In the development phase, the graphic designers worked on the ideas and prepared the handouts, posters, infographics, presentations, and worksheets. All of the phases were carried out with the contribution of all working group members, which were led by the academicians. The learning resources were generated in accordance with the previously completed phases, and the validation of the content was reviewed during the development phase (Branch, 2009).

A pilot study was carried out during the implementation phase. During the pilot study, teachers voluntarily contributed to all the activities and learning scenarios as suggested during the 18 weeks spent with their students. Prior to the implementation of the study, students were administered the 'Self-Perceived Computational Thinking Skills Questionnaire' and 'Academic Achievement Test' as a pre-test. The teachers were requested to evaluate and submit their experiences on a weekly basis. The students were graded on their progress during the semester in addition to a midterm and final exam. The same data collection tools used for the students' pre-assessment were also used at the end of the semester as a post-test. The pilot was carried out in order to reveal the learning progress of the students and to investigate the effect of the instructional materials and activities. Based on the findings of the pilot study, "it was seen that the content developed within the scope of the study, positively affected the computational thinking and academic achievement of the students between the ages of 11 and 12" (Kert, Kalelioğlu, & Gülbahar, 2019, p. 131). Thus, scientifically positive findings proved that the instructional materials and activities were effective in delivering coding skills to students.

Following the pilot implementation, the instructional materials and activities were assessed as part of the evaluation phase. The materials and activities were revised according to the findings, and republished as an open source. Hence, as a follow-up after one year of the pilot study, a general survey, 'Computing Course Evaluation Survey', was administered to teachers on a voluntary basis.

Participants

A total of 111 teachers participated in the survey study in order to evaluate the 'Teachers' Guide to Teach Computing' product, which included evaluation of the instructional materials and activities contained in the guide. Of those 111 teachers, 44 (39.6%) were female and 67 (60.4%) were male. The number of teachers with one to ten years of teaching experience was 59 (53.2%), whereas the number of teachers with 10 to 20 years of teaching experience was 52 (46.8%), which implies a mostly young generation of teachers.

Data Collection Tools

A survey was developed specifically for evaluating the instructional materials and activities by focusing on various aspects of the instructional process of the course for teaching of computing skills. The survey, named the 'Computing Course Evaluation Survey', is composed of 37 Likert-type questions with options ranging from disagree to agree. In addition, the survey included demographic data questions and one open-ended question that asked the participants to comment on other topics not covered in the main questions. The questions were prepared in order to reveal the teachers' insight as to the alignment of the content with the instructional objectives, timing, instructional materials and activities, visual design, pedagogical appropriateness, assessment opportunities, and class management.

Findings

The opinions of the teachers about the activities are presented in Table 1. When the mean scores of these items were evaluated from high to low, it was seen that 90% of the teachers want a similar guidebook to be prepared for different age groups ($\bar{X}$ = 4.45), 92% of the teachers stated that they are motivated to perform the activities in the classroom ($\bar{X}$ = 4.39), and that they have the knowledge and skills to perform these activities ($\bar{X}$ = 4.37).

92% of the teachers liked the presence of activities that can be used interchangeably ($\bar{X}$ = 4.24), whilst 87% reported positive opinions about the ease of reading the images in the activities ($\bar{X}$ = 4.19). 87% of the teachers found having the identity of the course activities, presenting teachers with a structure of the lesson that directs them to the course with an introduction, development, and conclusion components as positive aspects ($\bar{X}$ = 4.18). Parallel to this, the proposed key concepts for activities were considered appropriate by 89% of the teachers ($\bar{X}$ = 4.18). 88% of the teachers thought that the weekly lesson plans were helpful ($\bar{X}$ = 4.16), 86% stated that the activities were student-centred ($\bar{X}$ = 4.16), 84% considered that the skills to be gained on the course would support other courses ($\bar{X}$ = 4.14), and 80% of the teachers stated that the materials had an effective visual design ($\bar{X}$ = 4.12).

While 87% of the teachers considered that learning with the activities was fun ($\bar{X}$ = 4.12), 86% thought that the images in the activities were in an easily usable format ($\bar{X}$ = 4.12), 87% agreed that the teaching materials facilitated the students' learning ($\bar{X}$ = 4.07), 85% agreed that teaching materials were varied ($\bar{X}$ = 4.05), 83% stated that the activities facilitated students' learning ($\bar{X}$ = 4.03), and 82% of the teachers reported that their students showed a positive attitude towards the activities ($\bar{X}$ = 4.02).

Table 1. *Analysis results for teachers' opinions about the activities*

ITEMS	*f*	%	*f*	%	*f*	%	*f*	%	*f*	%	
	I strongly disagree		I do not agree		undecided		I agree		I strongly agree		x
1. The scope of the texts given for the lecture is sufficient.	4	3.6	18	16.2	9	8.1	60	54.1	20	18	3.67
2. The time allocated for the activities is sufficient.	21	18.9	34	30.6	15	13.5	28	25.2	13	11.7	2.80
3. The proposed course flow was well planned.	5	4.5	15	13.5	18	16.2	52	46.8	21	18.9	3.62
4. The pre-application notes section is very important in terms of preparation for the course.	3	2.7	6	5.4	6	5.4	53	47.7	43	38.7	4.14
5. Introduction, development and consequent planning of lesson plans made it easy to practice.	3	2.7	3	2.7	8	7.2	54	48.6	43	38.7	4.18
6. Identifying keywords for the activities provided convenience in teaching the concept.	4	3.6	3	2.7	5	4.5	56	50.5	43	38.7	4.18
7. The materials are of interest to students.	3	2.7	6	5.4	17	15.3	51	45.9	34	30.6	3.96
8. Materials are effective for visual design.	3	2.7	4	3.6	15	13.5	44	39.6	45	40.5	4.12
9. Weekly lesson plans helped to explain the lessons.	3	2.7	3	2.7	7	6.3	58	52.3	40	36.0	4.16
10. The weekly distribution of subjects is appropriate.	8	7.2	20	18.0	29	26.1	31	27.9	23	20.7	3.37
11. Activities prepared according to age level.	4	3.6	11	9.9	18	16.2	49	44.1	29	26.1	3.79
12. Pedagogically correct objectives are determined.	5	4.5	8	7.2	18	16.2	46	41.4	34	30.6	3.86
13. It was useful to have alternative activities.	3	2.7	4	3.6	2	1.8	56	50.5	46	41.4	4.24
14. Drama activities are compatible with the objectives.	3	2.7	7	6.3	17	15.3	58	52.3	26	23.4	3.87

ITEMS	*f*	%	*f*	%	*f*	%	*f*	%	*f*	%	
	I strongly disagree		I do not agree		undecided		I agree		I strongly agree		x
15. Recommended evaluation approaches are sufficient.	3	2.7	7	6.3	17	15.3	58	52.3	26	23.4	3.74
16. Activities are usually student-centred.	3	2.7	3	2.7	9	8.1	54	48.6	42	37.8	4.16
17. Generally, fun learning is targeted.	3	2.7	5	4.5	7	6.3	58	52.3	38	34.2	4.12
18. Activities help in achieving objectives.	3	2.7	6	5.4	17	15.3	53	47.7	32	28.8	3.95
19. The ranking of learning outcomes is appropriate.	4	3.6	10	9.0	19	17.1	56	50.5	22	19.8	3.74
20. The teaching methods used were diversified.	3	2.7	3	2.7	11	9.9	63	56.8	31	27.9	4.05
21. Images are designed to be easy to read.	3	2.7	5	4.5	6	5.4	51	45.9	46	41.4	4.19
22. Images are presented in an easy-to-use format.	4	3.6	7	6.3	5	4.5	51	45.9	44	39.6	4.12
23. The explanations for the teacher are sufficient.	5	4.5	11	9.9	8	7.2	52	46.8	35	31.5	3.91
24. The interest of students with different genders has changed.	8	7.2	11	9.9	24	21.6	45	40.5	23	20.7	3.58
25. The attitude of the students towards the activities is positive.	3	2.7	4	3.6	13	11.7	59	53.2	32	28.8	4.02
26. Teaching materials facilitated learning from the students' point of view.	3	2.7	5	4.5	7	6.3	62	55.9	34	30.6	4.07
27. Teaching materials facilitated teaching from the teachers' point of view.	4	3.6	7	6.3	10	9.0	55	49.5	35	31.5	3.99
28. Activity identities helped manage the teaching process.	3	2.7	4	3.6	12	10.8	60	54.1	32	28.8	4.03
29. The class can be easily managed while implementing the activities.	7	6.3	14	12.6	27	24.3	42	37.8	21	18.9	3.50
30. The proposed alternative activities are consistent with the objectives.	3	2.7	6	5.4	15	13.5	60	54.1	27	24.3	3.92

ITEMS	*f*	%	*f*	%	*f*	%	*f*	%	*f*	%	
	I strongly disagree		I do not agree		undecided		I agree		I strongly agree		x
31. I think I have the knowledge and skills to perform the activities.	3	2.7	4	3.6	2	1.8	42	37.8	60	54.1	4.37
32. I think I have the motivation to perform the activities.	4	3.6	4	3.6	1	0.9	38	34.2	64	57.7	4.39
33. The content of the Teachers' Guide is applicable to classes of different characteristics.	6	5.4	8	7.2	18	16.2	42	37.8	37	33.3	3.86
34. I think that the skills gained in this course will support other courses.	6	5.4	4	3.6	8	7.2	44	39.6	49	44.1	4.14
35. This material is sufficient for teaching my lessons.	17	15.3	26	23.4	29	26.1	18	16.2	21	18.9	3.00
36. The measurement tools provided are sufficient.	7	6.3	18	16.2	18	16.2	47	42.3	21	18.9	3.51
37. I think that the similarities of this guide should be prepared for different age groups.	4	3.6	3	2.7	4	3.6	28	25.2	72	64.9	4.45

81% of the teachers thought that the materials were helpful while teaching ($\overline{X}$ = 3.99), 77% thought that the materials were deemed attractive by the students ($\overline{X}$ = 3.96), 77% agreed that the activities helped in achieving the objectives ($\overline{X}$ = 3.95), 78% found that the activities were compatible with the objectives ($\overline{X}$ = 3.92), and 78% were positive about the explanations given to the teachers ($\overline{X}$ = 3.91).

Drama activities were also designed within the structure of the course. 76% of the teachers thought that these activities were compatible with the achievements, although 15% were hesitant about this. 72% of the teachers were positive and 16% were unsure about the pedagogical appropriateness of the objectives ($\overline{X}$ = 3.86). 71% of the teachers thought that the guidebook was suited to students with different characteristics and from different class dynamics, and 16% stated that they were undecided ($\overline{X}$ = 3.86). 72% of the teachers were satisfied with the scope of the presentations and texts, while 20% did not agree with this opinion ($\overline{X}$ = 3.67). 66% of the teachers believed the course flow to be well-planned, while 18% did not agree with this view, and 16% remained undecided ($\overline{X}$ = 3.62). While 61% of the teachers thought positively that the content was of interest to students of both genders, 17% did not agree, and 22% remained undecided. 61% of the teachers were positive, 16% were undecided and 23% were negative about the assessment tools ($\overline{X}$ = 3.51).

In addition, 50% of the teachers thought that the recommended duration for the activities was insufficient ($\overline{X}$ = 2.80). With regards to the teachers' thoughts that the material was sufficient on its own; 40% were negative, 26% were undecided, and 35% were positive about this item ($\overline{X}$ = 3.00). The same situation emerged in terms of the appropriateness of the weekly topic distribution; with 47% of the teachers positive about the issue, 26% were undecided, and 25% did not agree with this view ($\overline{X}$ = 3.37). As for the ease of classroom management while implementing activities, half of the teachers agreed with this view, while 24% were undecided, and 19% did not agree ($\overline{X}$ = 3.50).

Answers to Open-Ended Questions

As for the optional open-ended question at the end of the questionnaire, not many of the teachers provided a written answer. Of the opinions that were offered by the respondents, they were grouped under three main headings as positive opinions, criticisms, and suggestions. As to the positive opinions, seven of the teachers expressed their appreciation for the Teachers' Guide. One teacher on this subject stated that 'The teachers' guide and activities are simply great. During the period, I always thought, "I wish there was a guide written for different age groups, too"'.

For the responses grouped as criticisms, different views emerged. Among these, the most expressed opinion was that the activities were considered time-consuming (see Table 2). Also, the teachers stated that there were excessive stationery expenses, that the activities related to programming were at the level of knowledge and comprehension, the activities were difficult for 5th grade students, the activities were difficult to apply in crowded groups, and that there was difficulty in adjusting the level to the students. The teachers also stated that Scratch was deemed difficult for the students. On the subject of time management, one teacher stated that 'There are too many group activities, so classroom management –at the 5th grade level– was a problem', while another teacher said, 'A lot of time has been devoted to activities for schools with a computer class, but group activities are not always possible to implement'. Another teacher said, 'The lack of student materials in printed form makes it difficult to implement certain activities. The most common adverse event that I encountered was that the 80-minute period was insufficient to implement the given activities'.

Table 2. *Emerging themes for open-ended questions*

Response category/subcategory	*n*
Criticisms	
Activities without computers and group activities are time-consuming.	5
Stationery expenses are excessive.	2
5th grade algorithms and problem-solving subjects remain at the level of knowledge and understanding.	1
The 5th grade activities are more suited to 6th grade students.	1
Difficulty in application within larger groups.	1
Very difficult to adjust and balance the level.	1
Platforms such as Scratch are difficult.	1
Suggestions	
Guidebooks are needed for other grades.	6
There should be a students' guidebook.	5
The number of students and the duration of activities should be considered when diversifying the materials.	2
Guidebooks should be distributed to schools.	2
There should be a coding platform specific to our country [Turkey].	2
The number of activities to be implemented on the computer should increase.	1
There should be an online pool of activities, and it should be expanded by the teachers.	1
Improvements in terms of measurement and evaluation (sample questions in the guidebook for students are useful for exams).	1
Paper activities should be kept to a minimum.	1
There should be computer-aided design in the curriculum.	1
Website access to materials should be better structured.	1

In the category of suggestions, the teachers proposed that similar books should be created for other grades, that a students' book should be also written, that the number of students and class duration should be taken into consideration more in designing the guidebooks, printed books should be distributed, a programming platform to support the Turkish language should be developed, that there should be an increase in the activities that can be applied on the computer, there should be online an activity pool, that measuring and evaluation opportunities should be improved, paper activities should be reduced, having computer-aided design issues in the curriculum, and that the website structure could be made to be more useful. One of the teachers stated the following with regards to time management:

The activities are difficult to complete in a single classroom session; especially in a class below that of the middle level it became impossible to complete. Particularly, the 5th grade guidebook seemed to be really compressed in the fall semester; I think that the time allocated to the gains could be increased slightly. Another teacher said, 'However, there is a serious problem in preparing for the written examinations as the students cannot use the resources they work with. It would be a much more productive year if the students had their own workbook'.

DISCUSSION and CONCLUSION

The results of the study revealed that the 5th grade 'Teachers' Guide to Teach Computing' is a supportive material for CS education and that similar guides should be developed for different age groups. This general inference can be argued within two dimensions as teacher-related and student-related results.

The teacher-related results are based on the teachers' thoughts on the benefit of the guide to instructional practices. First of all, it was seen that most of the teachers found the activities and general framework of the lessons to be useful. As Park and Sung (2013) stated that computer science curriculum reform is not easy and can be challenging for teachers; however in the literature, it can be seen that there is a significant lack of supportive educational materials developed for CS teachers. As two research samples, Cutts et al. (2017) and Yadav et al. (2016) found, CS teachers can experience problems meeting the pedagogical and conceptual knowledge aims of their classroom instruction. Hence, it is crucial to develop pedagogical guidance tools in order to improve the efficiency of CS education.

The positive attitudes of the teachers towards the items of 'The scope of the texts given for the lecture is sufficient', 'Recommended evaluation approaches are sufficient', 'Activities help in achieving objectives', 'The explanations for the teacher are sufficient', 'Activity identities helped manage the teaching process' and 'This material is sufficient for teaching my lesson' are consistent with this aim. Simplicity is an important feature of effective educational materials. The teachers perceived that they have the knowledge, skills and motivation to perform the activities detailed in the guide. This can be seen as a notable indicator of the clear structure of the content and perceived as a compatible finding with the challenge skill balance pointed out by Nakamura and Csikszentmihalyi (2009). Additionally, when considering the findings, it can be said that the pedagogical questions of 'why', 'what', and 'how', as indicated by Saeli et al. (2011), were generally answered in the guide so as to enable CS teachers to gain the level of pedagogical content knowledge they need regarding programming education.

The student-related results were based on the teachers' observations of their students' attitudes towards the guide and the efficiency of the learning process. There was a common view among the teachers that the activities are appropriate to the intended age group (5th graders) and consistent with the course objectives. Fun was one of the key features of the guide and the teachers generally confirmed that students had fun during the implementation of the activities.

In addition, the teachers felt that the attitudes of the students towards the activities were positive and that the teaching materials facilitated learning from the students' point of view. These opinions were consistent with the propositions of Prensky (2001), expressed that having fun during the learning process supports students in being able to understand the content more easily and motivate them in their education. Furthermore, in his study, Papert (2002, p. 1) stated that "What I mean was brought up by a teacher who objected to the idea that children should be allowed to write about what they liked... Of course, we should teach children the skill of self-control needed to carry out orders" (2002, p. 1).

Consistent with Papert's (2002, p. 1) perspective, a student-centred approach was adopted in the design of the lessons, and it was seen that the teachers generally agreed to the item of 'Activities are usually student-centred'. Appropriate assessment criteria, interest-driven task design, and covering essential CT knowledge areas are among the key features of effective CT curricula (Kong, 2016). With a comparison of instructional material design principles from the literature, it can be said that the structural design features of developed content are consistent with the recommendations of Lashley (2019). They include effective guidance with sufficient instructions. On the other hand, as emphasized by Margulieux, Guzdial, and Catrambone (2012) reducing cognitive load is crucial in CS education. The plain visual design features of the materials would help learners to create their own mental models easily and reduce the cognitive load of the process.

The findings obtained from the current study showed that the 5th grade 'Teachers' Guide to Teach Computing' contains sufficient evaluation approaches, measurement tools, and materials which are of interest to the students. Despite most of the teachers' opinions being positive, on some points the teachers levelled considerable criticism at the activities contained in the guide. They mostly expressed that the time allocated for the activities was insufficient. It is thought that the structure of the unplugged CS activities and the overcrowded classrooms seen in some schools were significant to such criticism. The researchers believe that such criticism is of considerable importance in order to design more effective activities in CS education. As Falkner and Vivian (2015), stated "comprehensive development of curriculum resources that can assist teacher communities in both developing necessary skills, and informing their classroom practices" (p.

423). On the other hand, the teachers stated that they required similar guidebooks for the other grades that they teach, and that there must be adequate student resources to support the efficiency of the course.

A range of instructional materials composing of unplugged and plugged activities will not only help teachers for teaching computing for different contexts and circumstances but also manage computing classes effectively by supporting students who have different levels of knowledge and practices in terms of computing. Thus, it can be said that the results of the current study offered important recommendations to practitioners on how they should develop effective classroom course guides for CS teachers. Besides providing a roadmap for instructional designers, this research is useful for CS teachers that they can also design learning activities or modify existing materials according to their own needs. CS teachers should also be flexible enough to adapt existing materials and activities into their infrastructure. Moreover, the importance of activity-based learning and knowledge construction is underlined once more based on positive feedback from teachers.

REFERENCES

Blikstein, P., Sipitakiat, A., Goldstein, J., Wilbert, J., Johnson, M., Vranakis, S., Pedersen, Z. & Carey, W. (2016). Project Bloks: designing a development platform for tangible programming for children. Position paper. Retrieved from https://projectbloks.withgoogle.com/static/Project_Bloks_position_paper_June_2016.pdf

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*. doi:10.2791/792158. Luxembourg: Publications Office of the European Union. Retrieved from http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf.

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

Brennan, K., & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking.* Paper presented at the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

Caspersen, M. E. (2018). Teaching Programming. In S. Sentance, E. Barendsen, & C. Schulte (Eds), *Computer Science Education: Perspectives on Teaching and learning in school.* (s. 109-130). Bloomsbury Publishing.

Chen, K. & Rea, A. (2018). Do Pair Programming Approaches Transcend Coding? Measuring Agile Attitudes in Diverse Information Systems Courses. *Journal of Information Systems Education*, 29(2), 53-64.

Clark, R. C., & Mayer, R. E. (2016). *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. John Wiley & Sons.

Cutts, Q., Robertson, J., Donaldson, P., & O'Donnell, L. (2017). An evaluation of a professional learning network for computer science teachers. *Computer Science Education, 27*(1), 30-53. DOI: 10.1080/08993408.2017.1315958.

D'Angelo, C. M., Harris, C. J., Lundh, P., House, A., Leones, T., & Llorente, C. (2017). Examining the Types, Features, and Use of Instructional Materials in Afterschool Science. *School Science and Mathematics*, *117*(6), 269-281.

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

Falkner, K., & Vivian, R. (2015). A review of computer science resources for learning and teaching with K-12 computing curricula: An Australian case study. Computer Science Education, 25(4), 390-429.

Gamson, D. A., Eckert, S. A., & Anderson, J. (2019). Standards, instructional objectives and curriculum design: A complex relationship. *Phi Delta Kappan*, *100*(6), 8-12.

Guzdial, M. (2016, June 14). *Five Principles For Programming Languages For Learners.* Communications of ACM: Retrieved from https://cacm.acm.org/blogs/blog-cacm/203554-five-principles-for-programming-languages-for-learners/fulltext.

Gülbahar, Y. & Kalelioğlu, F. (2018). Bilişim Teknolojileri ve Bilgisayar Bilimi: Öğretim Programı Güncelleme Süreci. *Millî Eğitim dergisi, 47* (217), 5-23.

Hodell, C. (2010). *The Basics of ISD Revisited (Vol. 1010).* Alexandria, VA: American Society for Training and Development.

Kafai, Y. B., & Burke, Q. (2014). *Connected Code: Why Children Need to Learn Programming*. MacArthur Foundation Series on Digital Media and Learning. Cambridge, MA: MIT Press.

Kert, S. B., Kalelioğlu, F., & Gülbahar, Y. (2019). A Holistic Approach for Computer Science Education in Secondary Schools. *Informatics in Education*, 131-150.

Kong, S.-C. (2016). A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education, 3*(4), 377-394.

Lashley, L. (2019). A Reflective Analysis of the Selection and Production of Instructional Material for Curriculum Delivery at the Primary Level in Postcolonial Guyana. *SAGE Open*, *9*(2), 2158244019858445.

Lin, C. F., Hung, Y. H., Chang, R. I., & Hung, S. H. (2014). Developing a problem-solving learning system to assess the effects of different materials on learning performance and attitudes. *Computers & Education*, *77*, 50-66.

Lior, L. N. (2013). *Writing for interaction: crafting the information experience for web and software apps*. Burlington, MA: Morgan Kaufmann.

Margulieux, L. E., Guzdial, M., & Catrambone, R. (2012, September). Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Proceedings of the ninth annual international conference on International computing education research* (pp. 71-78). ACM.

Mason, R., & Cooper, G. (2012, January). Why the bottom 10% just can't do it: mental effort measures and implication for introductory programming courses. In *Proceedings of the Fourteenth Australasian Computing Education Conference-Volume 123* (pp. 187-196). Australian Computer Society, Inc..

Nakamura, J., & Csikszentmihalyi, M. (2009). Flow theory and research. *Handbook of positive psychology*, 195-206.

Papert, S. (2002). *Hard fun*. Bangor Daily News. Retrieved From http://www.papert.org/articles/HardFun.html.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*(2), 1-11.

Park, M., & Sung, Y.-K. (2013). Teachers' perceptions of the recent curriculum reforms and their implementation: What can we learn from the case of Korean elementary teachers? *Asia Pacific Journal of Education, 33*, 15–33. doi:10.1080/02188791.2012.756391

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

Prensky, M. (2001). Fun, play and games: What makes games engaging? *Digital game-based learning, 5*(1), 5-31.

Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in Secondary school: A pedagogical content knowledge perspective. *Informatics in Education, 10*(1), 73-88.

Shim, J., Kwon, D., & Lee, W. (2017). The effects of a robot game environment on computer programming education for elementary school students. *IEEE Transactions on Education*, *60*(2), 164-172.

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational researcher*, *15*(2), 4-14.

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 21st century: Why, what and when? *Education and Information Technologies, 22*(2), 445-468.

Webb, M. E., Bell, T., Davis, N., Katz, Y. J., Fluck, A., Sysło, M. M., & Brinda, T. (2018). Tensions in specifying computing curricula for K-12: Towards a principled approach for objectives. *IT-Information Technology, 60*(2), 59-68.

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

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235-254.