




Research Article

Use of Arithmetic Operation Skills in Block Based Programming Environments: A Comparative Case Study

Hayal YAVUZ MUMCU *¹  Suheda MUMCU ²  Ünal ÇAKIROĞLU³ 

¹Ordu University, Faculty of Education, Ordu, Turkey, hayalym52@gmail.com

²Trabzon University, Faculty of Education Trabzon, Turkey, suhedayildiz.b@gmail.com

³Trabzon University, Faculty of Education, Trabzon, Turkey, cakirogluunalts@gmail.com


* Corresponding Author: hayalym52@gmail.com

Article Info

Received: 18 March 2020

Accepted: 23 July 2020

Keywords: Arithmetic operation skill, block based programming, code.org, comparative case study

 DOI: 10.18009/jcer.705822

Publication Language: English

Abstract

The purpose of this study is to examine the 7th grade middle school students' processes of creating and calculating arithmetic expressions in the code.org block based coding environments. For this purpose, the reasons of students' behaviors in coding environments are tried to be revealed by comparing them with student behaviors in traditional environments. The participants of the study, which was conducted as a comparative case study, are composed of 10 students studying at a public school. *Arithmetic Operation Test* developed by the researchers, *Student Screen Records* taken from the code.org platform, *Student Opinion Form* consisting of three open-ended questions, *Unstructured Interviews* held with the students and *Field Notes* were used as data collection tools in this study. Descriptive analyzes were performed in order to compare the data obtained from different cases and the analysis results were expressed in different categories. In conclusion, it has been observed that the students were not as successful in coding environments as expected and use arithmetical operation skills more successfully in traditional environments than block based coding environments. In particular, it has been revealed that the use of blocks and the way in which code.org handles arithmetic operations cause the students cannot transfer their existing knowledge to the coding environments.



To cite this article: Yavuz-Mumcu, H., Mumcu, S., Çakıroğlu, Ü. (2020). Use of arithmetic operation skills in block based programming environments: A comparative case study. *Journal of Computer and Education Research*, 8 (16), 404-427. DOI: 10.18009/jcer.705822

Introduction

In parallel with the scientific and technological developments, the content and objectives of education are being renewed and the educational programs of countries are being revised for promoting the basic life skills. When it comes to teaching mathematics, the basic skills aimed to be taught today can be shown as mathematical reasoning, problem solving, creative thinking, modelling, critical thinking etc. (Ministry of National Education [MoNE], 2013; National Council of Teachers of Mathematics [NCTM], 2000). Numerous

opportunities offered by technology are benefited to teach all these skills, and the scope of technology-assisted mathematics teaching applications is being expanded day by day.

Computational thinking, in the most general sense, is defined as a way of thinking that a calculator, which can be human or computer, can be effectively used in framing or solving the problem (Wing, 2006). Although the idea of teaching computational thinking was first suggested by Seymour Papert (1980), the debates about the concept has become prevalent through the view of Wing (2006), stating that the computational thinking should be taught to every student in the 21st century as one of the basic areas such as reading, writing and arithmetic. Programming, which is one of the methods used to teach computational thinking, is now considered to be the skill of the 21st century and this is included in the curriculum of primary and secondary education in many countries (Mannila, Dagiene, Demo, Grgurina, Mirolo, Rolandsson & Settle, 2014). In our country, the lessons of Computer Sciences and Information Technologies and Software are compulsory in the 9th and 10th grades in secondary education and in the 5th and 6th grades in middle school respectively. Accordingly, it is stated that the aim of this lesson is to improve the skills of problem solving, reasoning, algorithmic thinking, generalization, making decision and computational thinking which is planned to be compulsorily in grades 1 to 8 (MoNE, 2018). Within the scope of problem solving and programming which is one of the five basic chapters of the teaching program, the students will be taught approaches to the problems encountered in daily life, how to solve these problems by using algorithms, and the features and uses of the current block-based programming tools. Emphasis is laid on improving the algorithmic thinking skills of students through block-based tools within programming. In this framework, basic programming structures are presented to the students in the form of scenarios and students' logical reasoning, mathematical modelling and generalization skills are expected to be improved through the use of appropriate algorithms to solve the problem situations created (National Research Council [NRC], 2008, Wing, 2008). At this point, the skills such as logical reasoning, algorithm, decomposition, abstraction, modelling, generalization and evaluation are among the basic skills required by the problem solving processes used in math lessons. Therefore, it can be concluded that the problems presented in block-based coding environments can be used directly or indirectly to teach the basic skills in math lessons (Fesakis & Serafeim, 2009). As a matter of fact, the number of the academic studies conducted for the areas where programming and coding processes can be used in

mathematics education (Genç & Karakuş, 2011) or the relationship of programming and coding processes with mathematics education (Larson, 2013) is increasing each passing day. From this point of view, this study tests the usability of block-based coding environments in improving the arithmetic operation skills as one of the specific mathematical skills.

Block-based Coding

Block-based coding environments are programming interfaces created with reference to the idea of simplifying visual programming environments in order to enable kids to study programming. These interfaces use block structures that minimize syntax errors and reduce cognitive barriers arising from the syntax compared to text-based programming interfaces. These types of environments are not meant to write typical application programs, but rather to develop problem-solving thinking; they are called as coding environments rather than programming environments (Yıldız, 2017). The platforms such as scratch (scratch.mit.edu), hackercan, code.org, codecademy and so on are examples of such encoding environments. Different types of blocks are used by the learners in accordance with the situation in these environments in order to be able to create the targeted mathematical situation or to produce a solution to a problem. In this study, the interface of code.org was preferred because of the advantages such as the Turkish language support which does not require installation, the online platform, the students' chance to see their own progress and the easy student follow-up thanks to the teacher interface.

Code.org

Code.org launched in 2013 is an online learning environment that aims to help students learn computer science through drag and drop activities. The courses available in the environment are suitable for students from kindergarten to the 8th grade and upper grades. The 20-hour courses, that can be accessed both online and offline, include the subjects such as programming concepts, loops, functions, functions with parameters, variables, operational thinking, analysis, models, abstraction and algorithms (Kalelioğlu, 2015). During these courses, the students are enabled to evaluate the accuracy of what they do thanks to the instant feedbacks and to learn individually. The situations including the basic structures of programming such as condition (if), loop, arithmetical operations, functions etc. are used as problem condition in the relevant environment and computational thinking skills are utilized in the solution processes.

Evaluation Blocks and Arithmetic Expressions (EBAE) being the second course of the interface of Computer Science in Algebra in code.org platform was used in this study. Within the scope of this course, the learners are expected to construct arithmetic expressions with the help of blocks and to code/model such operations. The course involves 15 different screens. In the first 10 screens, individuals are expected to code the arithmetic expressions provided by the program with blocks and to predict the correct answers of the related operations. The user receives feedback on incorrectly and correctly coded expressions, and s/he is provided to monitor the solution process thanks to the calculation of blocks which the individual has set up step by step by the program and to evaluate what s/he has done in this regard. On the 11th, 12th, 13th and 14th screens, the individuals are expected to express the arithmetic expressions coded (codified) by the program in writing. The 6th and 13th screenshots of the software are shown in Figure 1 respectively.

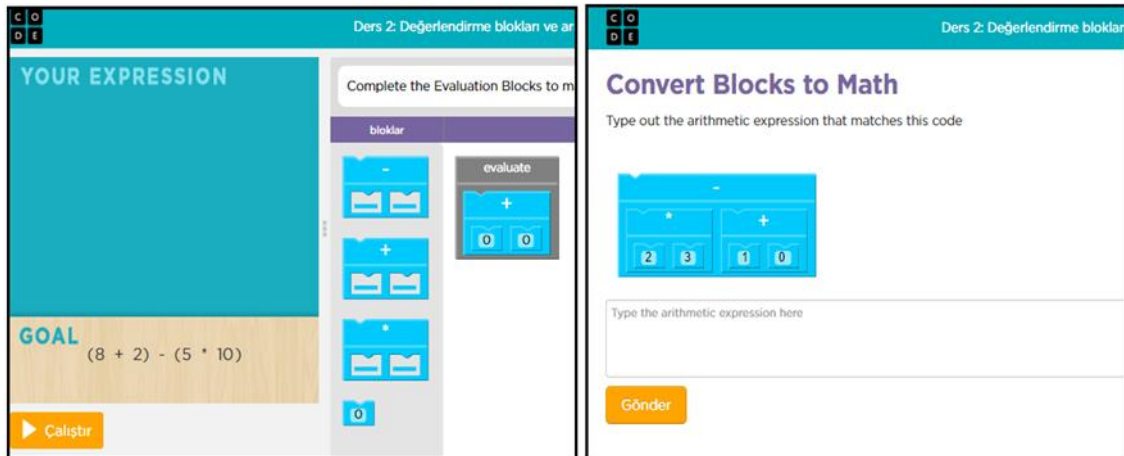


Figure 1. The 6th and 13th screenshots of the Code.org-EBAE course respectively

These screens provide similar feedbacks as described above. On the 15th and the last screen, a blank screen is presented to individuals and different arithmetic expressions to be created by themselves through blocks are calculated by the program. In this way, the learners will be able to compare an encoded expression with their algebraic form and the learning process will be completed by this means (Figure 2). It can be expressed that the learners will have the opportunity to improve their different knowledge and skills while the aim of the course is to teach the order of operations. Modelling of arithmetic expressions with the help of blocks in particular, has an important role in understanding the mathematical meanings of the operations in question. In addition, the formation of the loops involved in arithmetic expressions with the help of blocks is extremely favorable environments that can be used to improve the students' algorithmic thinking structure.

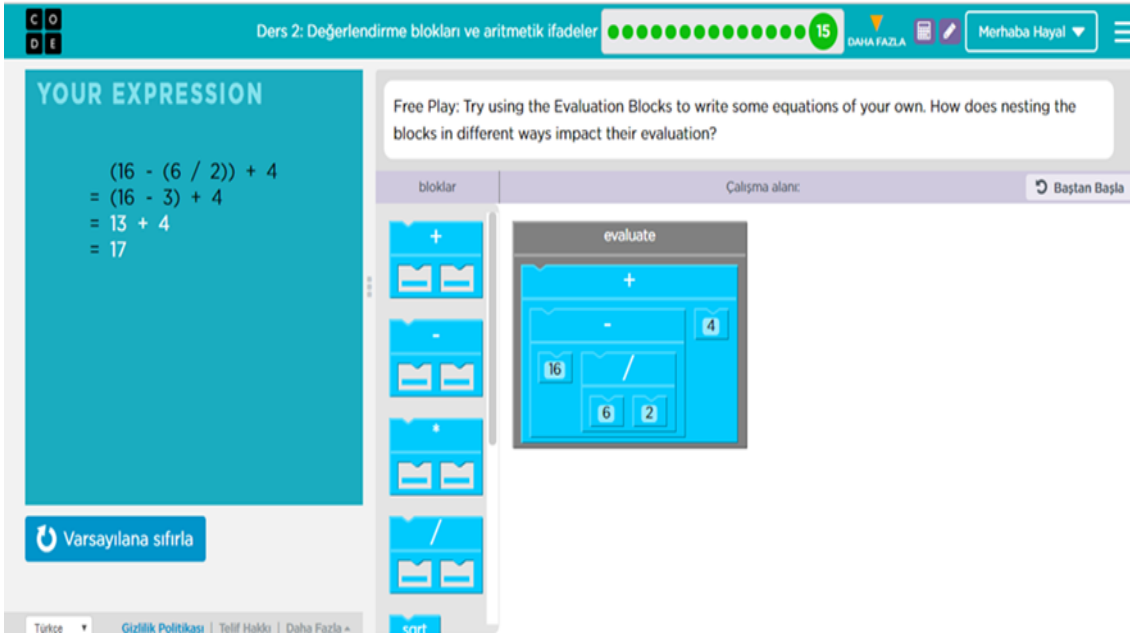


Figure 2. The 15th screenshot of the Code.org-EBAE course

From this point of view, this study has been conducted in order to compare the student performances in code.org aided block-based coding environments and traditional environments by examining the 7th grade middle school students' processes of creating and calculating arithmetic expressions. It is thought that the study will reveal the results and suggestions that will contribute to literature regarding the use of the code.org learning platform in teaching processes.

Method

Case study method which is used to identify and see the details of a situation, to develop possible explanations for a situation or evaluate a situation (Gall, Gall & Borg, 2007) was used in this study. Case studies address an event, such as a particular activity or a learning centered program (Hancock & Algozzine, 2006). In this study, this method was deemed appropriate since it was aimed to evaluate the usefulness of a learning environment in terms of student conceptions.

Besides, comparative case study, which is one of the different types of case studies, was used in the study. Comparative case studies compare multiple cases to understand a problem and aims to examine the common and individual characteristics of different situations (Baxter & Jack, 2008). Also these studies are often used to define the properties of different programs. This study was described as a comparative case study because of the

outcomes of different learning environments were examined comparatively and the results were interpreted based on the obtained data.

Participants

In case studies, participants are usually identified as consists of people who interact with each other, share the same place and know each other such as students in a class or teachers working in the same branch (Büyüköztürk, Çakmak, Akgün, Karadeniz & Demirel, 2018, p.270). Besides, it is recommended to study on small samples in order to obtain broad and in-dept information in these studies. So, the participants of this study consisted of 10 students attending the same class in 7th grades of a public school in the academic year of 2017-2018.

Purposeful sampling methods is used to provide maximum level of information for logical generalizations and making inferences in qualitative studies generally (Patton, 1990). So, it was used to determine the participants of this study, as arithmetic processing skills were tried to be observed and evaluated in different environments accordance with the purpose of the study. Accordingly, the students who have arithmetic operation skills above the average level were determined as participants of the study.

Data Collection Tools

The data collection tools of the study are Arithmetic Operation Test (AOT) developed by the researches, Student Screen Records created on the code.org platform, Unstructured Interviews (UI) held with students, Student Opinion Form (SOF) prepared by the researchers to reveal the students' opinions about the block based coding environments and Field Notes (FN). These tools are described in detail below.

Arithmetic Operation Test (AOT): The Arithmetic Operation Test developed by the researchers consists of 10 questions and there are different arithmetic expressions with and without parentheses in which the addition, subtraction, multiplication and division are combined. The aim of these questions is to observe whether the students pay attention to the order of operations and how they use their arithmetic operation skills in this context.

In the creation of the questions, the attainments in the curriculum for the related grade level, the opinions of the teacher who participated in the mathematics lessons of the participating students group and the opinions of the field experts (two math teacher, two lecturers who are specialized in the fields of Computer Education and Instructional Technologies and Mathematics Education) were used. Besides item discrimination and item

difficulty indexes of the questions in the test were calculated and determined to be suitable for the validity of AOT. In order to ensure the reliability of the test, Kuder Richardson (KR-20) method was used and the reliability coefficient was calculated as .89 for the AOT used in the study.

Student Screen Records: The student screen records were created and stored on the code.org platform. These records were used in order to examine the answers of the students to the AOT questions in the block-based coding environment. HyperCam software, which is a free application, was used in order to follow the behaviors of the students during the process in the block-based coding environment and the process steps they applied. Before the student started working on Code.org, the interface of the HyperCam (screen recording) software and shortcut keys were introduced to the students. In this way, it was tried to minimize the negativities that may occur during screen recording during the application. Students who entered the Code.org platform should be able to run the HyperCam screen recording program before solving the problem on the stage. During the screen video recording, students were physically observed in the classroom and field notes were taken.

Unstructured Interviews (UI): The unstructured interviews were used in order to code student answers properly within the categories formed by evaluating the students' performances in the traditional and block-based environments. The interview processes were sustained until the students' ways of thinking were clarified; accordingly, the length of interviews ranged from 45 minutes to 1 hour and these processes were recorded. During the interview, the students were asked the reasons for their movements in the process by the researcher.

Field Notes (FN): Field notes are defined as the expression of what the researcher sees, hears, observes and experiences in qualitative research process (Bogdan & Biklen, 2003). The researcher teacher involved in this study, recorded the observations and experiences related to the students' behaviors on the code.org platform as field notes. These short notes were detailed in order to describe what happened in the coding environment and classified by repetitive readings and converted into systematic data. Field notes not only create a rich data environment for the research, but also enable researchers to review the implementation process.

Student Opinion Form (SOF): In this study, the SOF was applied to students following the block-based activities. The purpose of using this data collection tool is to reveal the opinions

of students about the activities in question. There are three open-ended questions in the SOF as follows:

- What do you think about the coding activities you have conducted within the scope of this course? What are the positive and negative aspects of these activities?
- What are the most challenging situations (questions) in the coding processes? Express with the reason.
- Do you want the teachers to use the coding activities in the math lessons? Do you think these activities can help you learn mathematics better? Express with the reason.

Implementation

The whole application phase of the study was carried out by the researcher teacher involved in the study. The AOT was applied to the 34 students who were planned to be included in the study group. According to the test results, only the students who answered 50% of the questions correctly in the test were included in the study, since the data to be obtained from the students who did not have sufficient proficiency in arithmetic operations was envisaged not to be sufficiently valid and reliable. The 10 students determined according to the AOT results were given guidance on the use of the code.org interface for 4 course hours under the supervision of the researcher teacher. The students were generally informed about the purpose, content and use of the code.org website in the first two hours while they were subject to online courses in the other two hours. The students tried to complete the sub-sections of the lesson in a correct way and were informed by the teacher about the reasons of the situations when they received negative feedback. In this way, the students experienced the block-based coding activities freely during the 2-hour course.

After the interactive course, the implementation process of the study was conducted under the guidance of the researcher teacher. Within this period, the 15th page of the Evaluation Blocks and Arithmetic Expressions section of Code.org platform was studied and all the student data were stored through screen records. In this process, all questions in the AOT were distributed to the students and they were asked to solve these questions in code.org coding platform this time. Thus, it was attempted to compare the performances of the students in different environments and to test the usefulness of the related interface in this context.

Data Analysis

Descriptive analysis method was used to analyze the data obtained from the study and the answers of students were primarily grouped under four different categories for the traditional environment (TE) and the block-based coding environment (BBCE). According to this, the answers that are correct both in TE and BBCE are encoded as Correct Answer / Correct Coding (CA/CC), the answers given correctly in TE and wrongly in BBCE are encoded as Correct Answer/Incorrect Coding (CA/IC), the answers that are wrong in TE and correct in BBCE are encoded as Incorrect Answer/Correct Coding (IA/CC) and the answers that are wrong both in TE and BBCE are encoded as Incorrect Answer/Incorrect Coding (IA/IC). After that, appropriate coding was performed for the second time for common causes of the students' incorrect answers in each category. In this process, unstructured interviews were conducted with the students on the selected situations in order to be able to express student behaviors with appropriate codes and to decide the appropriate coding. The interviews were conducted with each participant of the study. The answers of the students in different environments were used together in the interviews and the reasons of their answers were tried to be revealed. In addition, the thinking processes of the students in different environments were interrelatedly observed and the findings of the study were interpreted based on this data and field notes obtained. The interviews held with each student were kept as voice records and these records were used in the data analysis process. If there were more than one reason of any wrong answer given to the questions, the answer in question was included in the frequency of more than one code in the coding processes by using the interview records. In the process of data analysis, the emphasis was laid on the reasons for the wrong answers of the students. The percentage of inter-coders compliance was calculated for the reliability of the content analysis and the coding processes performed within the study (Miles & Huberman, 1994) and a consensus was tried to be reached among the coders for the data analysis processes.

Findings

General Findings from the TE and the BBCE

The answers of the students in the traditional and block-based coding environments are given in Table 1.

Table 1. Distribution of the students' answers for different environments

Answers of the Students										
Q.No	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Q. 1	CA/CC	CA/CC	CA/CC	CA/CC	CA/CC	CA/IC	CA/CC	CA/CC	CA/CC	CA/CC
Q. 2	IA/CC	CA/CC	CA/CC	CA/CC	CA/CC	CA/CC	CA/IC	CA/IC	CA/CC	CA/IC
Q. 3	CA/CC	IA/IC	CA/CC	CA/CC	CA/IC	CA/IC	CA/IC	CA/IC	CA/CC	CA/CC
Q. 4	CA/CC	CA/IC	CA/CC	CA/CC	CA/CC	CA/IC	CA/CC	CA/IC	CA/CC	CA/CC
Q. 5	CA/CC	CA/IC	CA/CC	CA/CC	CA/CC	CA/IC	CA/CC	CA/IC	CA/CC	CA/CC
Q. 6	IA/IC	IA/CC	CA/CC	CA/CC	CA/CC	CA/IC	CA/IC	CA/CC	CA/IC	CA/IC
Q. 7	IA/IC	IA/IC	CA/IC	CA/CC	IA/CC	IA/IC	CA/CC	CA/IC	IA/CC	CA/IC
Q. 8	CA/IC	CA/IC	CA/CC	CA/IC	IA/CC	CA/IC	CA/IC	CA/IC	CA/CC	CA/IC
Q. 9	CA/CC	CA/IC	CA/CC	IA/CC	IA/IC	IA/IC	IA/CC	IA/CC	IA/CC	IA/CC
Q. 10	CA/CC	IA/IC	CA/CC	CA/IC	IA/IC	CA/IC	CA/CC	CA/CC	CA/CC	IA/IC

There are 100 (10x10) different cases related to 10 questions in the AOT for 10 students included in the study. According to the data in Table 1, the students gave correct answers for both environments in 49 of these cases. Therefore, it can be stated that the students can use their arithmetical operation skills moderately in block-based coding environments. It has been observed that the students have difficulty in using block-based environments rather than the traditional environments in 60.78% of the 51 remaining cases (cases with errors) and have difficulty in computing operations in the traditional environment rather than the block-based environment at a rate of 19.60% of these cases. Besides, the students have experienced difficulties in both of the environments in 19.60% of these situations. Based on the data given here, it can be stated that the students use arithmetical operation skills more successfully in traditional environments than in block-based environments and their mistakes are generally caused by coding processes.

Findings from the Category of CA / IC

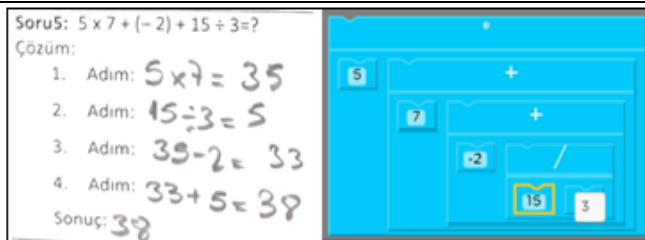
Regarding the answers in this category, the findings obtained from the analyses which were conducted to reveal the reasons of the incorrect answers in the BBCE are shown in Table 2.

Table 2. Distribution of the answers in the CA / IC category

Type of Mistake	Scope / Content of Mistake	f	tf
<i>Mistakes related to the Use of the Blocks (UB)</i>	Following the order in the question	14	29
	Inability to formulate nested operations with blocks	7	
	Ignoring any number	1	
	Disregarding the order of operations	5	
	Using a number twice	2	
<i>Mistakes related to Conceptual Knowledge (CK)</i>	Not using the order of operations	3	9
	Misinterpreting/Misusing arithmetic expression	3	
	Misusing the minus (-) sign	3	
<i>Mistakes related to Lack of Attention (Random mistakes) (LA)</i>	Using wrong operation	4	6
	Disregarding the order of operations	1	
	Using a number twice	1	

When the data in Table 2 are examined, it is seen that the category with the highest frequency is the *Mistakes Related to the Use of Blocks* (29). The following categories are the *Mistakes related to Conceptual Knowledge* (9) and the *Mistakes related to Lack of Attention* (6) respectively. Therefore, it can be said that the students generally have difficulty in the use of blocks in coding processes. The other reasons are the inadequacy of conceptual knowledge and lack of attention. Table 2 shows that the same codes are used in different categories. The reason for this is that the students made the same mistake because of different reasons. The data presented here are based on the reasons for the mistakes (reasons for doing so). In this part of the study, data analysis processes for different types of coding are presented as sample cases.

Sample Case 1 (UB- Following the order in the question): Interview Process with Student 8 (S8)



R: Why did you use the blocks here in this way?

Ss: I did it according to the order on the paper.

R: How?

Ss: So, 5 times 7, then 7 plus (-2) on the paper.

R: Which one do you think is correct?

Ss: The one on the paper.

R: What is wrong with the blocks?

Ss: I should have done the multiplication first. I did it without considering the order of operations.

R: Well, why do you think you did that?

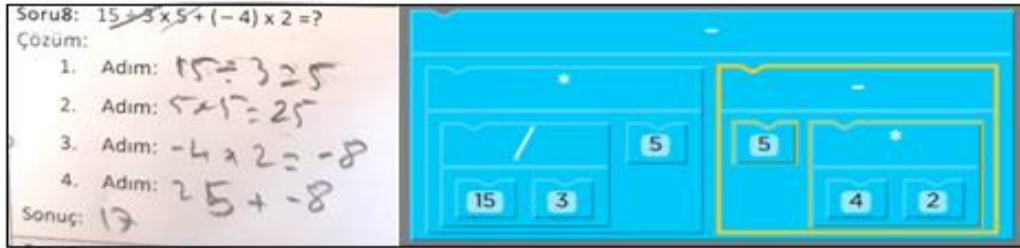
Ss: I think it's easier this way, that's why.

R: If I give these blocks to you now, can you line them correctly?

Ss: It's very complicated, there are three operations, and I don't understand whether to put the division first or to put the division in the most internal part. I mean I don't understand which one is more internal or external.

In the interview process with S₈, it is seen that the student has difficulty in using blocks in the coding environment although she calculates correctly the arithmetic expression given in the traditional environment. Depending on this situation, the student has formed the expression with blocks without considering the order of operations and parentheses.

Sample Case 2 (CI- Misinterpreting and misusing Arithmetic Expression): Interview Process with Student 10 (S₁₀)



R: What does this minus sign (the minus (-) between the multiplication and subtraction blocks is shown) mean?

S₁₀: The minus sign of -4.

R: How do you create the arithmetic expression in the coding environment?

S₁₀: According to the order of operation rules. Multiplication and division are done first according these rules, so multiplication and division are internal, addition and subtraction is done ultimately so I put the subtraction on the external part (on the top).

R: Well, which one of these blocks is calculated first by the computer? (showing multiplication and subtraction blocks)

S₁₀: The multiplication, the computer calculates this block first.

R: What is the minus next to (on the right side) this 5? (asking for the coding environment)

S₁₀: I do not know. I guess I made a mistake there.

R: Why did you make a mistake, where exactly did you have difficulty?

S₁₀: While positioning, I'm confused when I have presented the expression on the left with blocks.

R: Well, which of your answers in these two environments do you think are correct?

S₁₀: The one on the paper.

R: Why?

S₁₀: I used the number 5 twice here, and I used minuses twice. I did it wrong.

R: All right, how would you line these if we involved you in the coding process again?

S₁₀: I wouldn't use 5 twice. The rest sounds right.

R: Are you sure?

S₁₀: No, I am not.

The interview with S₁₀ shows that the student has performed the coding incorrectly as result of misinterpreting the arithmetic expressions. In general, the student is not sure about the accuracy of what he has done during the process of coding.

Sample Case 3 (LA-Using Wrong Operations): Interview Process with Student 2 (S₂)



R: Do you think the operations on the paper are the same with the ones in blocks?

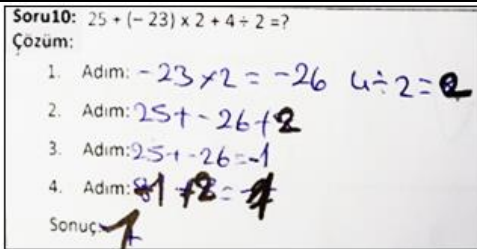

S2:	No.
R:	Why?
S2:	I multiplied -4 with 2 first and multiplied the result with 5. That's wrong. I should have added.
R:	Why did you do that?
S2:	It escaped my notice.
R:	Well, what else?
S2:	I used the division wrong.
R:	Why, then?
S2:	I don't know, I was confused I guess.
R:	Is it hard for you to use these blocks?
S2:	Yes, sometimes.
S:	Why?
S2:	I can get confused in the case of nested operations.

At the end of the interview with S2, it is seen that the student has used the wrong operation depending on the inability to fully understand the arithmetic expression given, and lack of attention. The fact that the student has used the wrong operation for the relevant question in the block-based coding environment is coded as using the wrong operation in the category of LA and as misinterpreting/misusing the arithmetic expression in the category of CK.

Findings from the Category of IA/CC

When the answers in this category are examined, it is seen that there are two kinds of reasons for wrong answers given in the traditional environment. These are *ignoring the order of operation rules -OOR* (4), and the *calculation errors-CE* (6). Therefore, it can be said that the mistakes students make when calculating the arithmetic operations given in the traditional environment are caused by lack of attention. The next reason is not to consider the order of operation rules.

Sample Case 4 (Calculation Errors): Interview Process with Student 4 (S4)

			
R:	Let's look at what you did on the paper and in the block-based environment.		
S4:	I multiplied -23 with 2 and found -26. The result should have been 46. I made a calculation error.		
A:	So, what about the blocks on the computer?		
S4:	I did it right. But no, there is -23 there but I used 23. That's wrong too. I should have written 25 plus -46.		
R:	So, what did you do?		
S4:	I formulated it as 25 minus 46.		
R:	Are they two different things?		

S ₄ :	If there are two minuses, they make each other plus (positive number), but if there is one minus, that number is taken negative, accordingly the operation is done. If I had taken 23 as -23 and used addition, it would have been more understandable.
R:	Well, are they two different things?
S ₄ :	No actually.
R:	Well, why do you think you did that?
S ₄ :	According to what's written on the paper I guess. I did it right in accordance with the order of operation rules but I made a mistake on the paper.

According to the interview with the student coded as S₄, it is clear that the student has given a wrong answer to the question in the traditional environment because of calculation error. In addition to this, it has been ascertained that the student does not have a complete and clear conceptual understanding of arithmetic operations.

Sample Case 5 (Ignoring the order of operations): Interview Process with Student 5 (S₅)

The image shows a handwritten solution on the left and a block-based calculator interface on the right. The handwritten solution is as follows:

Soru8: $(15 + 3) \times 5 + (-4) \times 2 = ?$
 Çözüm:
 1. Adım: $15 \div 3 = 5$
 2. Adım: $5 \times 5 = 25$
 3. Adım: $25 - 4 = 21$
 4. Adım: $21 \times 2 = 42$
 Sonuç: 42

The block-based calculator interface shows the expression $(15 + 3) \times 5 + (-4) \times 2$ being entered. The blocks for 15, 3, 5, 4, and 2 are visible, along with the operators +, *, and -. The result 42 is displayed at the bottom.

R:	In which environment you did is right do you think?
S ₅ :	In the computer environment.
R:	Why?
S ₅ :	Because I should have multiplied -4 with 2 first according to the precedence rules. I didn't do that, I did it wrong.
R:	Well, do you think your answer in the block-based environment is correct?
S ₅ :	Yes, because 4 and 2 must be multiplied first and subtracted from the first part. But I added it with 4, so I did it wrong.

It has been established that the student is able to distinguish the reason of the mistake s/he has made in the traditional environment during the interview process with the S₅ coded student. However, the student has answered the same operation considering the order of operation rules in the coding environment.

Findings from the Category of IA/ IC

The answers in this category are given in Table 3 for both the traditional and block-based coding environments depending on the type of the mistake.

Table 3. Distribution of the answers in the IA/IC category

	Type of Mistake	Scope/Content of Mistake	f	tf
BBCE	<i>Mistakes related to Conceptual Knowledge - CK</i>	Misinterpreting the arithmetic expression	6	11
		Using the minus (-) sign wrong	4	
		Not using the order of operation rules	1	
	<i>Mistakes related to the Use of the Blocks- UB</i>	Inability to create nested operations with blocks	3	6
		Following the order of operations in the question	2	
		Using a number twice	1	
	<i>Mistakes related Lack of Attention - LA</i>	Using the wrong operation	3	4
Ignoring any number		1		
TE	<i>Disregarding the order of operation rules -OOR</i>			6
	<i>Calculation errors-CE</i>			5

When the data in Table 3 are investigated, it can be seen that the frequency of the student mistakes originated from not following the order of operation rules in the traditional environment is 6 and the frequency of the calculation errors is 5 for the IA/IC category. Therefore, the students can be said to give incorrect answers for similar and common reasons in general. The reasons for the incorrect answers given in block-based coding environments in the IA/IC category can be ranked as the *mistakes due to conceptual knowledge-CK* (11), the *mistakes related to use of blocks-UB* (6) and the *mistakes related to lack of attention - LA* (4). The mistakes made in the coding environment in this category can be stated to arise from the lack of conceptual knowledge and the difficulties in the use of blocks respectively. A sample case is given below regarding the coding in Table 3 for the IA/IC category.

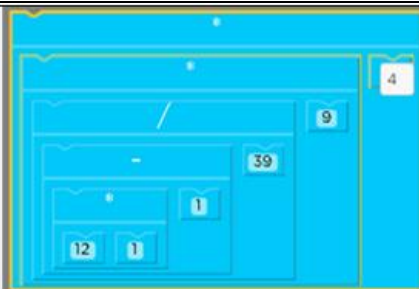
Sample Case 6 (Question 7): Interview Process with Student 6(S6) for the IA/IC category

Soru7: $12 \times 1 + (-39) \div (9 + 4) = ?$

Çözüm:

1. Adım:
2. Adım: $12 \times 1 + 1 = 13$
3. Adım:
4. Adım:

Sonuç:



R: Can you compare your solutions, which one do you think is right?

S₆: My solution on the paper is wrong.

R: Why?

S₆: It should have been -3, I didn't take into account the minus.

R: What do you think about what you did with the blocks? Why did you do that so?

S₆: I did it according to the order on the paper.

R: According to the order of operations?

S₆: Yes.

R: All right, does the minus here (-) (for BBC-block-based coding) belong to 39?

S₆: No. It belongs to 1.

R: You mean -1. Is there -1 in this operation? (showing the AOT)

S₆: No, there is not. I did it wrong.

R:	Okay, what is this operation after this 9?
S ₆ :	Multiplication.
R:	Why is there a multiplication sign here?
S ₆ :	Because of the order of the question... (the student thinks) Aah, I did this wrong either... There should have been plus sign here.
R:	Why do you think you have been surprised and made these mistakes? Have you not been able to fully comprehend the use of blocks?
S ₆ :	Yes. I can achieve it for short operations but I get confused in long operations.

During the interview process with S₆-coded student, it has been observed that the student gives a wrong answer to the current question due to calculation/operation error in the traditional environment. In the coding environment, the student has tried to create an arithmetic expression in a block-based environment according to the operation order on the paper, and given a wrong answer. The reason for the incorrect answer of the student in the block-based environment is coded as *misinterpreting /misusing arithmetic expression (CK)*, *using a number twice (UB)* and *using the wrong operation (LA)*.

Findings from the SOF (Student Opinion Form)

The data obtained from the SOF are given in Table 4. According to the data obtained from Table 4, it has been determined that the students have negative opinions about the coding processes as well as positive opinions.

Table 4. Distribution of the answers in the student opinion form

Question	Positive Aspects	f	Negative Aspects	f
<i>What are the positive and negative aspects of the coding activities you have conducted within the scope of this course?</i>	It is fun/wonderful as a positive aspect.	4	There's nothing negative about it.	4
	We are learning the order of operation rules.	2	A negative side is that our answers turn out wrong after a lot of effort.	1
	It is instructive.	2	Our calculations on paper get worse.	1
	Our mathematics is progressing in this way and we can perceive more easily.	1	I have a little difficulty as a negative aspect.	1
	It is a good application since it teaches how to code.	1	I've had a bit of trouble since I am used to doing operations on paper.	1
	Calculations on paper are made by the computer.	1		
	The combination of mathematics and informatics.	1		
<i>Do you want the use of the coding activities by the teacher in the math lessons? Do you think these activities can help you learn mathematics</i>	Yes, I do.	f	No, I don't.	f
	Yes, I do. It makes contribution/I understand the lesson better.	4	No, I don't. I have had great difficulty in coding. It's easier to learn mathematics on the board or on the paper.	1
	Yes, I do. It is easier while solving math/difficult	2	No, I don't, because it might not contribute to learning for	1

<i>better?</i>	questions.	everyone.
<i>Express with the reason.</i>	Yes, I do. We have the chance to see that we use mathematics on the computer.	1
	Yes, I do. The system warns me when my answer is wrong, I understand my mistake.	1
<i>What are the most challenging situations (questions) in the coding processes? Express with the reason.</i>	I've had difficulty in long/multi-operation questions, because they were so complicated.	3
	No/they were all easy.	3
	I've had difficulty in the last questions because it was difficult to sort them out by the order of operation rules.	2
	I've had difficulty in the questions with nested operations.	1

The frequency of the students' positive opinions about the coding activities has been calculated as (12 + 4) 16, and the frequency of the negative opinions has been calculated as 4. When the positive opinions are analyzed, the coding activities are evaluated as fun (4) and instructive (5). In addition to this, the students have found the coding activities useful because mathematics and computer are combined, the students are able to make long calculations practically and have learned how to code.

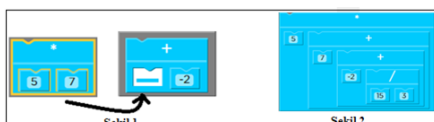
While the students have stated that they have difficulty mostly in the too long and nested operations within the coding activities (6), 3 students have remarked not to have any difficulty in the activities. Therefore, it can be said that students generally adopt the idea of using coding activities in math lessons. When the answers in this category are considered, the students utter that they will understand the lesson better through these activities (4), they can solve the long and difficult questions more easily (2) and they can correct their mistakes by noticing them (1).

Discussion and Conclusions

Within the framework of this study, the purpose is to compare the performance of 7th grade students in middle school by analyzing the students' creating and calculating arithmetic expressions in block based coding environments and traditional environments. In this way, the attempt was to test the usefulness of block-based environments. The study has shown that the answers of the students for arithmetic expressions in different environments differ. The study has also revealed that the students can use their arithmetic operation skills more successfully in traditional environments compared to block-based environments. This finding can be explained in relation to the students' having no experience with block-based

coding environments and their habits of thinking and solving operations. On the other hand, the way in which Code.org handles arithmetical operators and the students' problems of how to transfer especially the order of operations to Code.org environment requires a short-term preliminary instruction for the use of such environments to develop skills. The participant students of this study have carried out only 4-hour free activities in the BBCE. Similarly, Özmen and Altun (2014) state that one of the most important reasons for the failure of the students in the programming environment is the lack of practice. However, different studies in this regard indicate that difficulties in teaching programming are experienced especially by beginners because of the reasons such as inability to understand the concepts of programming language (Bayman & Mayer, 1983), and not apprehending the syntax of programming language and the standard structures used in programming (DuBoulay, 1986), and the lack of the skills required to write a program. The study results are discussed in relation to these results in the following sections.

When the reasons for the incorrect answers of students in the block based coding environment are investigated, these reasons have been determined to arise mainly from the difficulties in using blocks (UB). The students have arranged largely blocks based on the order of operations within the question. The interviews conducted with the students have shown that this situation is caused by the fact that the students do not know how to formulate the existing expression by using blocks. The students have had difficulty in using blocks in the nested operations. This situation is supported by the data obtained from the opinion form filled in by the students. Some of the other difficulties the students have experienced in using blocks are the inability to create nested operations with blocks and ignoring the order of operation rules. The relationship between the difficulties expressed here can be explained as follows: e.g., the student can easily create the operation of 5×7 by using blocks for the expression of $5 \times 7 + (-2) + 15 \div 3$, but s/he has difficulty in nesting blocks while adding (-2) to this multiplication ($5 \times 7 + (-2)$); because (5×7) must be placed in the addition block as the first component of a new block. For this, the student must first display the addition block and then place (5×7) as the first component of this operation, and (-2) as the second component in the block, s/he must move reversely in a way (Figure 1).



The students used the blocks according to the order of operations in the question since they could not achieve this sequence (Figure 2 / S8-Question 5). Ignoring the precedence rules in this category is entirely relevant to this situation. During the interviews held with the students, it was found that the students knew the precedence rules and followed these rules in the traditional environment but they had to ignore the precedence rules since they couldn't achieve the sequence in the block-based environment. The other mistakes related to the use of blocks were to use a number twice and to ignore any number. The reasons for these situations can be explained by the difficulties experienced by the students in the process of working with blocks and the loss of their awareness. The students, who had difficulties in constructing the given arithmetic expression by using blocks, used a number a twice due to the binary feature (consisting of two numbers) of blocks or ignored any number. For instance, the student knows that the multiplication must be done first in the arithmetic expression of $25 + (-23) \times 2 + 4 \div 2$ but s/he can't keep the number of 25 out, so s/he cannot place $[(-23) \times 2]$ as second component into the addition block. Since it is easier to place the first two numbers in a block for the student, s/he uses $25 + (-23)$ together with the addition block. This situation causes the student to use a number twice and to code the given expression in the format of $25 - 23 - [(23 \times 2) + (4 \div 2)]$. This situation was encountered in some of the students. Depending on the situation, the students can also ignore any number. The cases of using a number twice or ignoring any number occur in this way in the category of inability to use blocks correctly. Apart from this, another reason of these situations is that the students lose their awareness due to the difficulty in nesting blocks. Therefore, it is seen that the main reason of the situations in the category of the mistakes related to the use of blocks is that the students cannot comprehend the logic in the use of blocks (codes). This result is also expressed in different forms in literature. Pea and Kurland (1983), Özmen and Altun (2014) state that the most basic reason of the challenges in learning programming is "the failure to constitute the logic of programming". Besides, Esteves and Mendes (2004), Ozoran, Çağıltay and Topallı (2012) indicate the complexity of the concepts of the programming language as one of the relevant reasons alike this study. Likewise, it has been observed that the block-based coding environments are complex in accordance with the data obtained from the interviews held with the students and from the student opinion forms.

Another type of the incorrect answers given by the students in the block-based environment is the mistakes resulting from the conceptual knowledge (CK). In this category,

the cases of ignoring the precedence rules, misinterpreting/misusing the arithmetic expression and misusing the minus sign according to the order of frequency have been observed. These cases can be seen as the reasons based on the conceptual knowledge of the students' incorrect answers in the block-based environment. The interviews held with the students have clarified that the students obey the precedence rules in the traditional environment but they can't achieve this in the coding environment. It can be said that this is because the students use the precedence rules only in traditional environments and they have not fully understood what these rules actually mean and why they are used. The interviews conducted within the scope of the study indicate that the students behave more freely as they can move the arithmetic operations given to them in the block-based environments without following the order on the paper, and they tend more to ignore or not to use the precedence rules in these environments. The misinterpretation/misuse of the arithmetic expression, another type of mistakes in the CI category, can also be explained in relation to the aforementioned situation. The fact that the student (S2) has coded the expression of $12 \times 1 + (-39) \div (9+4)$ (Question 7) as $[(12 \times 1) \div (9+4)] - 39$ can be considered as an indication that s/he could not interpret the algorithm of the expression given. In the case of the mistakes related to misusing the minus sign, it has been observed that the students use interchangeably the signs of the numbers and the arithmetic expressions in the questions. This situation can be considered as an indicator that the students have insufficient conceptual knowledge about arithmetic operations. The interview held with the student (S2) who has expressed $12 + 4 \times 2 + (-1)$ (Question 4) as $1 - [(4 \times 2) + 12]$ in the coding environment has showed that the student's mistake is based on the lack of conceptual knowledge besides the inability to use the blocks correctly. Therefore, the general reasons of the mistakes in the CK category can be expressed as the lack of knowledge of the students about the meanings of arithmetic expressions and operations. It is stated in different studies (Hayes & Stacey, 1990; Van De Walle, Karp & Bay-Williams, 2012) that students do not have a complete learning for arithmetic operations. In parallel to the results obtained from this study, Yenilmez and Bağdat (2014), Bostan (2010) point out in their studies that students cannot distinguish between the minus sign "-", which is the direction of a negative number, and the symbol of the subtraction "-".

It has been ascertained in this study that the reasons for the incorrect answers of the students in the traditional environment within the IA/CC category are making mistakes in

operations/calculations or not taking into account the precedence rules. The students who made the aforesaid mistakes could accurately calculate the result since they could correctly code the expressions in the block-based environment. The unstructured interviews held with the students have demonstrated that the general reasons of the mistakes are lack of attention for the IA/CC cases that constitute 19.60% of all cases observed within this study. The predominant reasons for the incorrect answers in the IA/IC category have been discovered to depend on the lack of conceptual knowledge. In this category, the students have been found not be able to fully understand the arithmetic expressions with regard to their mistakes in the traditional and block-based environment in most of the interviews held with the students. Therefore, these results can be related to the discussion in the CK category.

Despite a number of limitations encountered in the process of computing arithmetic operations, the positive assessments of the students for the block-based environment in general provides significant clues regarding the use of such environments in math lessons. According to the data obtained from the student opinion form used in this study, the students have been detected to find the block-based coding environments entertaining and instructive, to have positive opinions on the use these environments in math lessons but to have difficulty in expressing long and complicated questions with blocks. Different studies in literature (Calder, 2010; Ruf, Mühling & Hubwieser, 2014) have reached similar results, and these studies point out that visual programming environments increase students' motivation and attitude to learn programming.

The study showed that the students find studying in block based coding environments fun and instructive but they experience some difficulties in these environments. The most important reasons for these difficulties have been specified as the inability to fully adapt to the block-based coding environment, to predict how the blocks function and to express nested operations by using blocks. In order to prevent these types of situations, the students are recommended to spend more time and to gain experience in block-based coding environments, to be under the supervision of teachers during coding processes and to receive feedback from them, and the situations observed as a result of coding activities are suggested to be discussed in class environment. It is thought that students will be able to eliminate their errors related to both the use of blocks and the meaning of mathematical concepts within the teaching process, and will be able to be more successful in the relevant processes.

The study is limited in some certain respects. First; we considered the scenarios presented in code.org and we followed the code.org construction of arithmetic operation skills and the quantitative evidences were presented for only limited topics. Second; the selection and size of the study subjects may also limit the generalization of the study findings. We collected the data from small group of students; for generalization, future studies to investigate similar constructs with a broader population in different block based learning environments and within extended skills are warranted. We hope that this study sheds light for using block based code environments and helps mathematics educators in the course design and implementation side.

Acknowledgement

The data used in this study was confirmed by the researchers that it belongs to the years before 2020.

Authorship Contribution Statement

Hayal YAVUZ-MUMCU: *Conceptualization, design of the work, literature search, data analysis, data interpretation, writing - review and editing.*

Suheda MUMCU: *Conceptualization, data collection, preliminary analyses, manuscript draft, writing, manuscript revision*

Ünal ÇAKIROĞLU: *Conceptualization, design of the work, literature search, manuscript revision.*

References

- Bayman, P., & Mayer, R. E. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. *Communications of the ACM*, 26(9), 677-679. DOI: 10.1145/358172.358408.
- Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. *The Qualitative Report*, 13(4), 544-559.
- Bostan, M. I. (2010). Negatif sayılara ilişkin zorluklar, kavram yanlışları ve bu yanlışların giderilmesine yönelik öneriler [Difficulties and misconceptions regarding negative numbers and suggestions for overcoming these misconceptions]. In E. Bingolbali & M. F. Özmantar (Eds.), *İlköğretimde karşılaşılan matematiksel zorluklar ve çözüm önerileri [Mathematical difficulties encountered in primary education and solution suggestions]*, pp. 155-186, Ankara: Pegem.
- Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2018). *Eğitimde bilimsel araştırma yöntemleri [Scientific research methods (24th ed)]*. Ankara: Pegem.
- Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom (APMC)*, 15 (4), 9-14.

- Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57-73. DOI: 10.2190/3LFX-9RRF-67T8-UVK9.
- Esteves, M., & Mendes, A. (2004). A simulation tool to help learning of object oriented programming basics. In Proceedings of the 34th ASEE/IEEE Frontiers in Education Conference, Savannah, GA, USA. Retrieved from <https://ieeexplore.ieee.org/document/1408649>.
- Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. In *ACM SIGCSE Bulletin*, 41 (3), 258-262. New York: ACM. DOI: 10.1145/1595496.1562957.
- Gall, M. D., Gall, J. P., & Borg, W. R. (2007). *Education research: An introduction (8th ed.)*. Boston: Pearson Education.
- Genç, Z., & Karakuş, S. (2011). *Learning through design: Using scratch in instructional computer games design*. In 5th International Computer & Instructional Technologies Symposium (ICITS), pp. 22-24, Elazığ, Turkey.
- Hancock, D. R., & Algozzine, B. (2006). *Doing case study research: A practical guide for beginning researchers*. Columbia University, NY: Teachers College.
- Hayes, B., & Stacey, K. (1990). Teaching negative number using integer tiles. Unpublished doctoral thesis, University of Melbourne, USA.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200-210. DOI: 10.1016/j.chb.2015.05.047.
- Larson, E. (2013). Coding the curriculum: How high schools are reprogramming their classes. Retrieved from <http://mashable.com/2013/09/22/coding-curriculum>.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference* (pp. 1-29). New York: ACM. DOI: 10.1145/2713609.2713610.
- McMillan, J. H. (2000). *Educational research: Fundamentals for the consumer*. New York: Longman.
- Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: A Sourcebook of new methods* (2d Edition). Beverly Hills, CA: Sage Publications.
- Ministry of National Education (2013). *Ortaokul matematik dersi (5-6-7-8. sınıflar) öğretim programı [Secondary school mathematics (5-6-7-8. grades) curriculum]*. Ankara: MEB.
- Ministry of National Education (2018). *Bilişim teknolojileri yazılım dersi öğretim programı [Information technologies and software course curriculum]*. Ankara: MEB.
- National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics*. Reston, VA: National Council of Teachers of Mathematics.
- National Research Council (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington, D.C.: The National Academies Press.
- Ozoran, D., Çağltay, N. E., & Topallı, D. (2012). Using scratch in introduction to programming course for engineering students. *Educational Technologies & Distance Education in Engineering*, 2, 125-133.
- Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: Difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 9-27.
- Patton, M. Q. (1990). *Qualitative evaluation and research methods*. US: Sage Publications.
- Pea, R., & Kurland M. (1983). *On the cognitive prerequisites of learning computer programming*. (Technical Report No. 18). New York: Bank Street College of Education.

- Ruf, A., Muhling, A., & Hubwieser, P. (2014). Scratch vs. Karel: Impact on learning outcomes and motivation. Paper presented at the *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, Berlin, Germany. DOI: 10.1145/2670757.2670772.
- Van de Walle, J.A., Karp, K.S., & Bay-Williams, J.M. (2010). *Elementary and middle school mathematics: Teaching developmentally* (7th ed.). Boston, MA: Allyn & Bacon.
- Wing, J. (2008). Computational thinking and thinking about computing. *Phil. Trans. R. Soc. A*, 366, 3717-3725. DOI: 10.1098/rsta.2008.0118.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-5. DOI: 10.1145/1118178.1118215.
- Yenilmez, K., & Bagdat, O. (2014). Learning difficulties of year seven students on whole numbers problems. In *Abstracts of International EJER Congress 1st Eurasian Educational Research Congress*, 631-632.

Copyright © JCER

JCER's Publication Ethics and Publication Malpractice Statement are based, in large part, on the guidelines and standards developed by the Committee on Publication Ethics (COPE). This article is available under Creative Commons CC-BY 4.0 license (<https://creativecommons.org/licenses/by/4.0/>)