Journal of Learning and Teaching in Digital Age, 2019, **6**(1), 72-80
https://dergipark.org.tr/en/pub/joltida
ISSN: 2458-8350 (online)

Research Paper

The Effect of Programming with Scratch Course on Reflective Thinking Skills of Students Towards Problem Solving

Esra Çoban Budak[a], Aynur Kolburan Geçer[b], Arzu Deveci Topal[*c]

[a](ORCID ID: 0000-0003-4408-8836), Kocaeli University, TURKEY, esra@kocaeli.edu.tr
[b](ORCID ID: 0000-0002-2000-9526), Kocaeli University, TURKEY, aynurgecer@hotmail.com
[c](ORCID ID: 0000-0001-5090-8592), Kocaeli University, TURKEY, arzudevecit@gmail.com
[*]Corresponding author

ARTICLE INFO
Received: 11 August 2020
Revised: 14 October 2020
Accepted: 09 December 2020

Keywords:
Scratch program
Reflective thinking skills towards problem solving
Basic Computer Sciences course

ABSTRACT
With the development of information and communication technologies, access to information is becoming easier every day. Consequently, individuals' abilities to interpret information that especially exists in the learning processes, to produce new knowledge and to use the information produced have gained more importance day by day. The aim of this study is to determine the effect of digital story design made by students working on project in groups using Scratch program on their reflective thinking skills towards problem solving. The sample of the study consisted of 82 students who attended Basic Computer Sciences (Social) course and studied in the verbal departments such as History, Archaeology and Philosophy of Faculty of Arts and Sciences of Kocaeli University in the 2018-2019 academic year. In this study, single-group pretest-posttest experimental design was used and data were collected by "Reflective Thinking Skill Towards Problem Solving Scale". According to the findings, it was determined that teaching of Scratch programming has a positive effect on reflective thinking skills of the students studying in the departments of Archelogy, Philosophy and History and is mostly effective in sub-dimension of questioning. In future studies, students' reflective thinking skills towards problem solving can be measured through project-based, observational applications and achievement tests.

INTRODUCTION

With the advancing technology, access to information gets increasingly easier. This may decrease the ability of individuals to interpret information, to produce new knowledge and to use the information produced in their educational lives (Karabulut, 2015; İTÜ Rehberlik , 2012; Karar and Vatan Newspapers, 2018). Employers state that learning ability, problem solving skills and teamwork are important (OECD, 2001). Problem-solving ability is a prerequisite for success in most professions and in everyday life. In the 21st century, new approaches have differently proposed that instead of only learning the information in their lives, individuals should rather be able to use and question it, produce new information, transfer it and find solutions to the problems they face (Lai & Land, 2009). Reflective thinking skills play an important role in the development of problem solving skills of the students (Kapranos, 2007).

The concept of reflective thinking can be defined as individuals' thinking about what and how they have learned by looking at what they experienced during and after learning process in order to gain new knowledge, and thinking internally to decide for new knowledge to learn by evaluating themselves (Strampel & Oliver, 2007; Uygun & Çetin, 2014). The purpose of reflective thinking is to understand a case or a problem, to involve solving the problem better, and to enable increase of the learning experience by maintaining the students' interests (Epstein, 2003; Kızılkaya and Aşkar, 2009; Baş, 2013).

Phan (2009) and Al-Tarawneh (2015) stated that reflective thinking can be considered as one of the most important style of thinking that has contributed to solve future problem and at data analysis, it helps students experience too much information needed for problems and situations, so student success increases. Hsieh and Chen (2012) demonstrated that, in the process of software design, reflective thinking strategy has a positive impact on the improvement of learning performance of the students especially who are less successful. There are many studies focusing on the relationship between programming processes, problem solving and reflective thinking (Fessakis, Gouli, & Mavroudi, 2013; Liao & Bright, 1991).

Some research studies regarding that reflective thinking skills are used in the programming process and affect the success are available in the literature. In one of these studies, Havenga, De Villiers and Mentz (2011) reported that high-performing programmers used various mental activities and supportive strategies by applying various cognitive, reflective and psychological processes and activities while writing their programs. According to Mohd Rum and Ismail (2017), development of metacognitive skills, including the conceptual framework provided by socio-cognitive theories that provide reflective thinking, such as active monitoring, evaluation and regulation of a thought, is important for the new learners of programming. Deng and friends stated that

in the sense that block-based programming tools had positive effects on students' programming attitude and that they could also promote the cultivation of computational thinking, reflective thinking and problem-solving ability.

Steps for the solution of the problem in the programming process for learners are as follows: understanding the problem, analysing the problem, developing solutions, implementing the way of solution by coding and evaluation (Polya, 1985; Brown and Kusiak, 2002). In order to carry out these steps easier, it will be quite helpful to include practical processes and concrete examples in which the eyesight and sense of touch are at forefront in teaching programming languages. Digital story design is important in solving the problems via visualizing step by step while giving concrete examples.

Digital stories can be described as an artistic activity created by combining traditional storytelling composed by individual or collaborative group work with multi-media components such as text, visual, audio narration, music and video (Sarıtepeci, 2017). According to Robin (2016), digital story design allows students to reflect what they have learnt creatively and critically. Individuals develop communication skills by learning to organize ideas, asking questions, expressing their ideas and developing narratives while creating a digital story; when they share their work with their peers, they can also increase their emotional intelligence, collaboration and social learning gains because they criticize their own work and the work of other students. In addition, Ohler (2005) proposed that digital stories should be used to strengthen students' critical thinking, report writing and media literacy.

Scratch program which is one of the visual programming languages and addresses every age group is based on digital story design. Programs such as Scratch, StarLogo TNG, which also allow for collaborative work, contribute to the success of the students in creating self-efficacy for developing their problem-solving skills, increasing their meaningful expression repertoire creatively and personally, and helping to eliminate the limitations in their language skills (Deveci Topal, Çoban Budak and Kolburan Geçer , 2017; Millner, Huang and Corbett, 2013).

Based on this information, it is thought that the programming course learned by digital story design and cooperative learning method can contribute positively to reflective thinking. According to Durak, Yılmaz and Yılmaz (2019) there was a positive and moderate relationship between computational thinking, programming self-efficacy and reflective thinking aimed at problem solving. The related literature has studies examining the effect of Scratch programming tool on programming teaching (Federici, 2011; Su et al., 2015). However, it is seen that studies which investigate whether reflective thinking which provides understanding of a situation or a problem and solving the problem better develops by using Scratch program are very limited in the literature.

Taylor, Harlow and Forret (2010) emphasized that programming education is effective on the teaching of mathematical subjects, development of problem solving strategies, collaborative, systematic and creative thinking, and Kert and Uğraş (2009) stated that algorithm and programming skills have contributed a lot to one's thinking and problem solving skills. It is also important for students studying in such departments as philosophy, history, archaeology of universities to develop their reflective thinking skills, to know how to extract information from resources, how to question objects and works, to be informed about history and historical processes, and to build knowledge. Therefore, reflective thinking skill is necessary for individuals to keep up with the current age regardless of their professions. This sample group (history and archaeology departments) has not taken any programming lessons before. Scratch, which is a visual programming language, was used in order to learn simply, in this study.

In this study, it is aimed to determine the effect of digital story design implemented by project work in groups and with Scratch program during programming course on reflective thinking skills towards problem solving with students in the verbal departments of Faculty of Arts and Sciences. The sub-problems of the study are as follows:

1. Does Scratch program have any effect on their reflective thinking skills towards problem solving?
2. Based on the final test scores, what is the perception level of students' reflective thinking skills towards problem solving?
3. Do students show significant difference in developing reflective thinking skills towards problem solving according to the demographic characteristics (such as gender, department and duration of computer use)?

METHOD

In this part of the study, research model, study group, data collection tools and data analysis have been included.

Research Model

In this study, pre-test / post-test single-group semi-experimental research model was used. Measurements of quantitative and qualitative variables of the participants are obtained by using pre-test before study and post-test after study using the same participants and measurement tools.

Sample of The Study

The sample of the study consisted of 82 students who attended Basic Computer Sciences (Social) course and studied in the first grade of History and Archaeology departments and in the second grade of Philosophy department of Faculty of Arts and Sciences at Kocaeli University in the 2018-2019 academic year. The demographic characteristics of the students participating in the study are given in Table 1.

Table 1. Demographic characteristics of the students participated in the research

Gender	N	**%**
Female	55	67
Male	27	33
Department	**N**	**%**
Archaeology	25	30,5
Philosophy	20	24,4
History	37	45,1
Computer ownership	**N**	**%**
Yes	60	73,2
No	22	26,8
Duration of daily computer use	**N**	**%**
Less than 1 hour	40	48,8
More than 1 hour	42	51,2
Knowledge of programming	**N**	**%**
Those having attended a programming course before	8	10
Those having tried to write a program before	5	6
Those not having programming knowledge	69	84

Data Collection Tools

The "Reflective Thinking Skill towards Problem Solving Scale "which was developed by Kızılkaya and Aşkar (2009), used in this study consists of 5 likert type 14 items and includes three sub-dimensions which are questioning (5 items), causation (4 items) and evaluation (5 items). The KMO value of the scale was calculated as 0.872 and Bartlett's Test of Sphericity value was 1084.329 ($p < 0.01$) and the reliability coefficient for all of the scale items was calculated as Cronbach's Alpha = 0.83. Cronbach alpha values have been examined for the reliability estimates of the dimensions. According to the analysis result, for the "questioning" dimension it has been found as 0.73, for the "reasoning" dimension it has been found as 0.71, and for the "evaluation" dimension it has been found as 0.69.

Action frequencies that items include are arranged at levels of Always (5), Most of the time (4), Sometimes (3) Rarely (2), Never (1). Total score of the scale was formed as the sum of the answers given to 14 items in terms of these scores. The height degree of the total score is interpreted as the degree of having reflective thinking skills.

Experimental Process

In programming, visualizing narration of algorithm steps written in verbal language with digital story designs eases understanding of the problem concretely. In Basic Computer Science (Social) course, which is instructed to students who study in verbal departments, students are taught the steps of problem solving and gain the necessary skills to perform simple practices that can be used in daily life and to detect and correct errors in program codes. Learning coding logic in programming is facilitated by use of user-friendly and visual programming tools (e.g. Scratch, Code.org, Code monkey, AppInventor) (Gezgin, Özcan, Ergün, Köse and Emir, 2017). In order to explain the programming logic by visualization, programming with Scratch was taught, the students were asked to prepare project work in groups of four, so they first designed the digital story on paper and coded it in scratch program. The students presented the story part of the project they prepared to the instructors and their classmates in the midst of the study and they presented the coding part in the last week of the study. The study lasted for 12 weeks, four hours per week. In this study, it is aimed to determine how the teaching of Scratch programming affects reflective thinking skills of students towards problem solving. The distribution of the subjects on a weekly basis is given in Table 2.

Table 2. Distribution of subjects by weeks

Week	**Subject**
1	Identifying the problem and developing solutions. Definition and types of the algorithm. Operators and examples used in algorithms and examples. Definition and examples of flow diagrams.
2	Process Structure in Algorithms: Definition and types of variable. Concepts of assignment-transfer of variable, increase-decrease. Visualisation of algorithm and flow diagrams with digital story telling.
3	Introduction of Scratch program interface. Transfer of digital story designs (algorithms) into Scratch program. Presentation of "Clicked" command "Start" and "Stop" command for "Finish" in order to run the program in algorithm. Recognition of the code blocks and presentation of the result of the output of the algorithm steps as a screen output.
4	The importance of user interface design in programming. Description of the interface design principles of Jakop Nielsen. Organizing individual or groups for project work. Creation of digital story designs.
5	Explanation of commands in objects (decor / stage), character (sprite) in programming. Adding or designing decor /character in Scratch program. Displaying definition of variable in Scratch code block.

	Examples of Verbal / Quantitative Problems with Algorithmic Design and Scratch (Examples such as loop concept by moving of character with commands of repeat until/ forever , deceleration / acceleration by adding seconds, stage / character change, character speech)
6	Transfer of project works, of which digital story designs have been prepared, into Scratch program. Designing of stage (decor) and character (sprite) in their works by project groups, change and motion of the stage / character. Control of project designs.
7	Description of condition commands (IF / Option - Case) in the program. Display of algorithm and flow diagram examples of verbal / numerical problems that should be solved logically via Scratch program.
8	Stage change according to conditions / character's asking question via using variable, moving with the keyboard arrow keys, adding / dropping points. When any key on the keyboard (such as space) is pressed, character (sprite), that is, when the object is clicked and the decor is ready, presentation of the commands with examples. Example of labyrinth with points.
9	Drawing geometric drawings with loop commands. Logic of sub-programs with commands of "broadcast" and "when I receive" in the programming. Drawing shapes with commands of "when clicked" and "broadcast/when I receive". Making calculation applications such as four operations, average, maximum with examples.
10	Control of project work. Detection of algorithm / flow diagram errors towards problem solving and troubleshooting with feedback.
11	Doing examples with general application questions for revising.
12	Last control of individual/group project work.

In order to ensure that the algorithm steps, which are the basis of programming, can be written accurately and clearly by verbal expressions, lecturing is supported by sample algorithms which include Scratch program codes. Some examples done in computer laboratory are given below. Firstly, the students were given to expression of problem verbally and then how to write algorithm steps by verbal language and drawing of the flow diagram are shown so as to solve the problem via programming (Figure 1). The commands written in Scratch Program were executed and the results (outputs) were displayed on the screen.

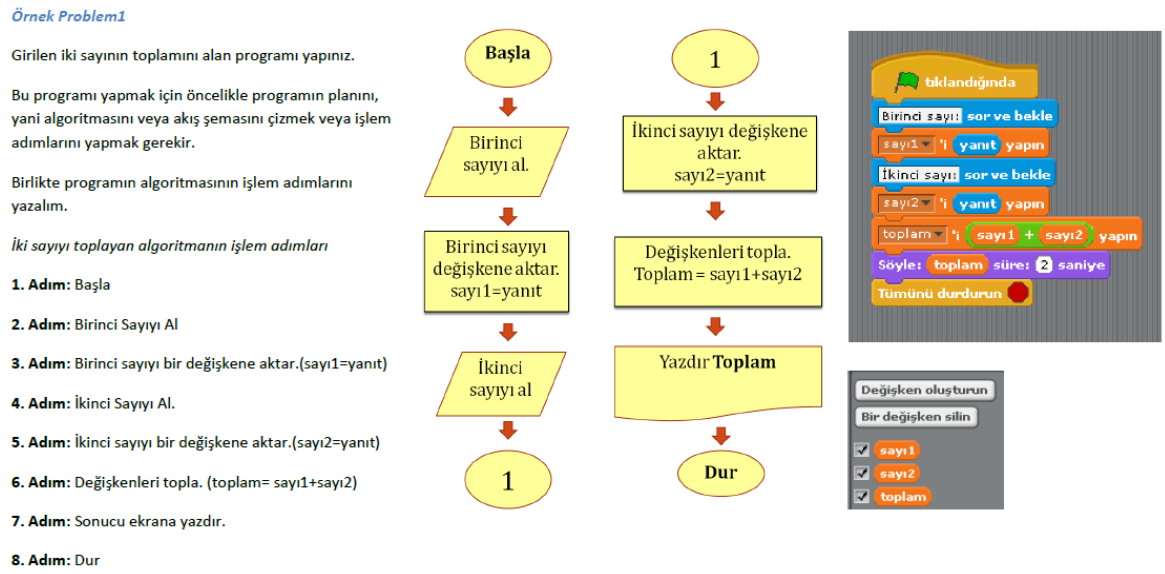

Figure 1. Algorithm of addition, flow diagram drawing and Scratch program codes

In order to improve programming logic of the students by using problem solving method, the algorithm steps of the problem or the command statements in Scratch program were left blank in some sample applications (Figure 2). With similar logic, places of algorithm steps were written in wrong order, and the students were asked to put them in correct order.

Soru: Scratch programında kare ve üçgen şekilleri çizdirmek için programda boş bırakılan yerlere gelecek olan komutları yazınız?

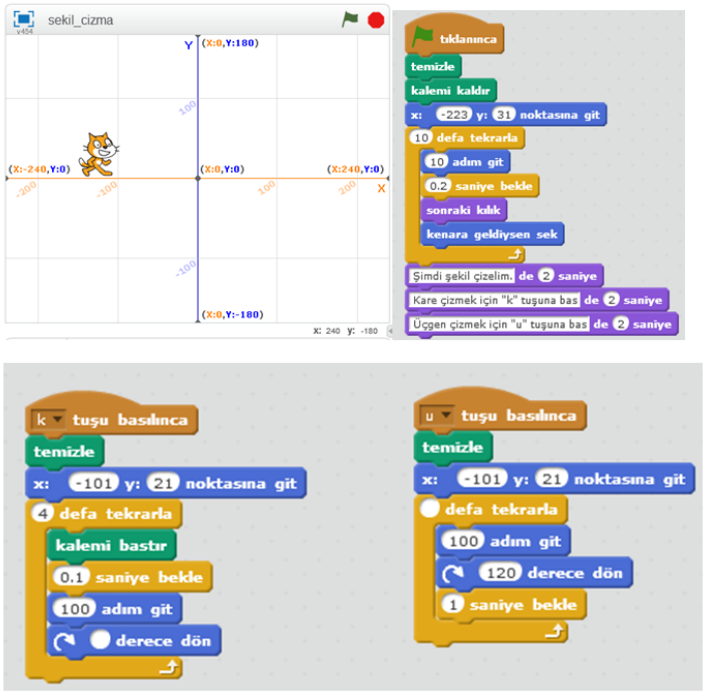

Figure 2. Writing of Commands in Square and Triangle Drawings

Data Analysis

SPSS 18.0 program was used for data analysis. Descriptive statistics, frequency analysis, mean, standard deviation, matched sample t test and independent sample t test were applied. The significance level was accepted as 0.05. Assuming that the degree range is equal in responses to scale items, the highest value was subtracted from the lowest value and divided into the number of degrees. The value of this range is 4/5 = 0.8 and is given in Table 3.

Table 3. Limits of Score Distribution Related to Research Scale

Options	Limits
Always (5)	4.20-5.00
Most of the time (4)	3.40-4.19
Sometimes (3)	2.60-3.39
Rarely (2)	1.80-2.59
Never (1)	1.00-1.79

FINDINGS

Does teaching students in the verbal departments of Faculty of Arts and Sciences Scratch program have any effect on their reflective thinking skills towards problem solving?

The matched group t-test was applied to determine the difference between perception levels of students' reflective thinking skills towards problem solving before and after teaching of algorithm and programming with Scratch software and the results are given in Table 4. The results of the matched group t-test showed that when the students' reflective thinking skills towards problem solving before and after instruction of programming with Scratch was examined, a significant difference in favor of the post-test was found in the overall score ($t(81)=2,015$; $p<.05$) and in sub-dimension of questioning ($t(81)=2,094$; $p<.05$). While average of the pre-test scores of the students was $\bar{x}=52,46$, the average of the post-test scores was $\bar{x}=54,09$, and average of pre-test scores in the sub-dimension of questioning was $\bar{x}=19,01$ whereas the average of post-test scores was $\bar{x}=19,65$. There was no significant difference in sub-dimensions of evaluation ($t (81) = 1,183$; $p> .05$) and causation ($t (81) = 1,891$; $p> .05$). This finding provided that 12-week instruction of programming with Scratch positively affected the students' levels of reflective thinking skills towards problem solving.

Table 4. T-test results of pre-test and post-test average scores of students' reflective thinking skills towards problem solving

Reflective thinking skill towards problem solving		X	N	S.s	t	df	p
Total	Pre-test	52,46	82	9,09	2,015	81	0,047
	Post-test	54,09	82	10,00			
Questioning	Pre-test	19,01	82	3,34	2,094	81	0,039
	Post-test	19,65	82	3,62			
Causation	Pre-test	14,88	82	2,99	1,891	81	0,062
	Post-test	15,45	82	3,01			
Evaluation	Pre-test	18,57	82	3,63	1,183	81	0,240
	Post-test	18,99	82	3,80			

Based on the final test scores, what is the perception level of students' reflective thinking skills towards problem solving?

The descriptive statistics of the items in the scale applied to the students in order to determine their perception levels of reflective thinking skills towards problem solving were calculated, and the findings are given in Table 5. According to this table, the students think that they most of the time perform the skills required by reflective thinking towards problem solving in both total and all dimensions.

Table 5. Perception levels of reflective thinking skills of students towards problem solving

Reflective thinking skill towards problem solving	N	Item number	X	S.s
Total	82	14	3,86	0,71
Questioning	82	5	3,93	0,72
Causation	82	4	3,86	0,75
Evaluation	82	5	3,80	0,76

Do students in the verbal department of Faculty of Arts and Sciences show significant difference in developing reflective thinking skills towards problem solving according to the demographic characteristics (such as gender, department, duration of computer use)?

Table 6 shows the pre-test and post-test average scores of the students' reflective thinking skills towards problem solving based on frequency of daily computer use. According to this table, it was determined that in the whole pre-test results, those who used the computer for more than 1 hour had higher reflective thinking skills towards problem solving than those using the computer less than 1 hour. According to the results of the post-test, a significant difference was found in favor of those who used computer more than 1 hour only in the sub-dimension of evaluation. This case, except for sub-dimension of evaluation, was interpreted as the students' skills started to be close to each other after 12-week Scratch instruction, regardless of daily computer use duration. It can be said that teaching programming with Scratch contributes positively to reflective thinking skills.

Table 6. T-test results of pre-test and post-test average scores of reflective thinking skills towards problem solving according to frequency of daily computer use

Reflective thinking towards problem solving		Duration of daily computer use	N	X	s. s	t	p
Pre-test	Total	Less than 1 hour	40	49,98	8,53	3,098	0,003
		More than 1 hour	36	56,06	8,57		
	Questioning	Less than 1 hour	40	18,35	3,13	2,331	0,022
		More than 1 hour	36	20,03	3,14		
	Causation	Less than 1 hour	40	14,05	3,06	2,836	0,006
		More than 1 hour	36	15,94	2,72		
	Evaluation	Less than 1 hour	40	17,58	3,34	3,267	0,002
		More than 1 hour	36	20,08	3,34		
Post-test	Total	Less than 1 hour	40	52,00	10,07	1,813	0,074
		More than 1 hour	36	56,17	9,92		
	Questioning	Less than 1 hour	40	19,03	3,78	1,497	0,139
		More than 1 hour	36	20,28	3,49		
	Causation	Less than 1 hour	40	14,85	2,90	1,657	0,102
		More than 1 hour	36	16,00	3,15		
	Evaluation	Less than 1 hour	40	18,13	3,88	2,036	0,45
		More than 1 hour	36	19,89	3,64		

The pre-test and post-test scores of the students were analysed according to gender, department, ownership of computer, having attended programming course and having written program before, but no significant difference was found according to these variables.

CONCLUSION AND DISCUSSION

In this research, teaching of algorithm which constitutes working logic of the computer was carried out using method of working in groups and teaching of Scratch programming was performed by digital story design, and it was aimed to find out how this instruction affected the students' reflective thinking skills towards problem solving. For this purpose, a 12-week study was planned and "Reflective Thinking Skill Towards Problem Solving Scale" was applied to the students before and after the study.

According to the results of the study, it was determined that at the end of the study, the students who study in Archaeology, Philosophy and History had a positive effect on their reflective thinking skills towards problem solving after instruction of programming with Scratch and the instruction was effective especially in the sub-dimension of questioning. At the end of the study, it was found that the students generally performed the skills required for reflective thinking towards problem solving. In a similar study, Yıldız Durak (2018) stated that programming with scratch positively affected the reflective thinking skills of 5th grade students. Also, Erümit (2019) stated that the mathematical and game preparation activities with Scratch had positive effects on algorithmic thinking and reflective thinking skills for problem solving. Norris and Jackson (1992) suggested that teaching of programming improves problem solving skills, and it will be useful to teach programming for all students at university level, especially when critical thinking and developing skills of problem solving are an objective. In their study, Fessakis, Gouli and Mavroudi (2013) demonstrated that five and six-year-old preschool children enjoyed attractive learning activities and had the opportunity to improve their mathematical concepts, problem solving and social skills as a consequence of computer programming study. Clements and Gullo (1984) determined that the group who had education with programming got higher scores than computer assisted education group had according to the criteria of reflectivity and different thinking, metacognitive abilities and ability of describing directions. Sarıtepeci (2017) has determined that use of digital storytelling activities has an impact on development of students' reflective thinking skills. On the other hand, Kalelioğlu (2015) did not find any significant difference in the end of the study carried out during an experimental research examining the effect of code.org program, similar to Scratch, on the reflective thinking skills of secondary school students, but the students were understood to have developed a positive attitude towards programming. As a result of the study, it was determined that students' skills started to be close to each other regardless of daily computer use duration after 12-week course of Scratch.

Finally, it can be said that after teaching of programming language with Scratch, students' reflective thinking skills towards problem solving get more positive. Based on the research findings, it may be suggested that reflective assessment and computational thinking skills related to problem solving should be included in the training programs as part of the programming courses applied at different levels. The results obtained here indicate the students' own perceptions. It would be more accurate to apply project-based, observational and reliable-valid success tests in order to measure students' reflective thinking skills towards problem solving in a

proper and accurate way. Also, the relationship between the scores obtained from reflective thinking skills scale towards problem solving and the academic achievement of the students can be researched.

REFERENCES

Al- Tarawneh, A. (2015). Reflective thinking and its relationship with future problem solving for Mutah University students. *British Journal of Humanities and Social Sciences, 13*, 2, 25-31. http://www.ajournal.co.uk/HSpdfs/HSvolume13(2)/HSVol.13%20(2)%20Article%204.pdf

Bas, G. (2013). The relationship between primary school students' reflective thinking skills and academic achievement in science and technology course with structural equation model. *Hasan Ali Yucel Journal of Education Faculty, 10*(2), 1-12

Brown, D. & Kusiak, J. (2002). *Problem Analysis Techniques. IRM Training- White Paper*. https://www.miun.se/siteassets/fakulteter/nmt/summer-university/problemanalysispdf

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

Deng W, Pi Z, Lei W, Zhou Q, Zhang W. (2020). Pencil Code improves learners' computational thinking and computer learning attitude. *Computer Application in Engineering Education, 28*, 90–104. https://doi.org/10.1002/cae.22177

Deveci Topal, A., Çoban Budak, E. & Kolburan Geçer, A. (2017). The effect of algorithm teaching on the problem-solving skills of deaf-hard hearing students. *Program, 51*(4), 354-372. https://doi.org/10.1108/PROG-05-2017-0038

Durak, H. Y., Yilmaz, F. G. K., & Yilmaz, R. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Educational Technology, 10*(2), 173-197. https://doi.org/10.30935/cet.554493

Epstein, A. S. (2003). How planing and reflection develop young children's thinking skills. *Young Children, 58*(5), 28-36. https://eric.ed.gov/?id=EJ679112

Erümit, A.K. Effects of different teaching approaches on programming skills. *Education and Information Technologies, 25*, 1013–1037. https://doi.org/10.1007/s10639-019-10010-8

Federici, S. (2011). A minimal, extensible, drag-and-drop implementation of the C programming language. *13th Annual Conference on Information Technology Education*, New York, America.

Fessakis, G., Gouli, E. & Mavroudi, E. (2013). Problem Solving by 5-6 Years Old Kindergarten Children in a Computer Programming Environment: A Case Study. *Computers & Education, 63*, 87-97. https://www.learntechlib.org/p/132276/

Gezgin, D., Özcan, S., Ergün, K., Köse, Ö. ve Emir, N. (2017). Opinions of high school students on the use of scratch program in computer programming education, *Proceedings Book of 2nd International Scientific Research Studies Congress on Humanities and Social Sciences (IBAD)*, İstanbul, Turkey, April 20-23, 2017, ISBN: 978-605-66529-0-5, 182-188, Received from: http://dergipark.gov.tr/download/article-file/384530

Gracely, E. (2004). *The Role of Quasi-Experimental Designs in Pain Research. Pain Medicine, 5(*2), 146–147, https://doi.org/10.1111/j.1526-4637.2004.04023.x

Havenga, M., De Villiers, R. & Mentz, E. (2011). Thinking processes used by high-performing students in a computer programming task. TD: *The Journal for Transdisciplinary Research in Southern Africa, 7*(1), 25-40, http://dspace.nwu.ac.za/handle/10394/3605

Hsieh, P. & Chen, N.S. (2012). Effects of reflective thinking in the process of designing software on students' learning performances. *Turkish Online Journal of Educational Technology - TOJET, 11*(2), 88-99.

İTÜ Geliştirme Vakfı Okulları (2012). Bilgisayar ve televizyon. İlköğretim, rehberlik bültenleri, 2 Mayıs, 1-8, Accessed date: 30.07.2019, http://www.itugvo.k12.tr/ilkogretim/Rehberlik_bultenler/bulten_bilgisayar_ve_televizyon.pdf

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior, 52*, 200-210. https://doi.org/10.1016/j.chb.2015.05.047

Kapranos, P. (2007). 21st century teaching & learning kolb cycle & reflective thinking as part of teaching, creativity, innovation, enterprise and ethics to engineers. *International Symposium for Engineering Education*, 2007, Dublin City University, Ireland, 3-11, http://doras.dcu.ie/447/1/Kapranos_ISEE07.pdf

Karabulut, B. (2015). Digital natives, immigrants and hybrids in the age of the information society. *Pamukkale University, Journal of Social Sciences Institute, 21*, 11-23, http://dergipark.gov.tr/download/article-file/411659

Karar Newspaper (2018). *Can't solve problems, doesn't understand what you're reading*. Accessed date: 01.08.2019. https://www.haberler.com/karar-gazetesi/

Kert, S., B. ve Uğraş, T. (2009). Simplicity and Entertainment in Programming Education: The Example of Scratch. *The First International Congress of Educational Research*, Çanakkale, Turkey, 1-15.

Kızılkaya, G. & Aşkar, P. (2009). Developing reflective thinking skill scale for problem solving. *Education and Science, 34*(154), 82-92. http://egitimvebilim.ted.org.tr/index.php/EB/article/view/550

Lai, T. L., & Land, S. M. (2009). Supporting reflection in online learning environments. In *Educational media and technology yearbook, Springer Science and Business Media, 141-154*, doi: 10.1007/978-0-387-09675-9 9 https://www.researchgate.net/publication/226021889_Supporting_Reflection_in_Online_Learning_Environments

Liao, Y.-K. C., & Bright, G. W. (1991). Effects of Computer Programming on Cognitive Outcomes: A Meta-Analysis. *Journal of Educational Computing Research, 7*(3), 251–268. http://doi.org/10.2190/E53G-HH8K-AJRR-K69M

Millner, A., Huang, W. & Corbett, C. (2013). Creating Opportunities for Young Learners with Hearing Impairments to Program, Design, and Learn Language in New Ways. *IDC'13 Workshop*, June 24 – 27, New York City, USA.

Mohd Rum, S. N., & Ismail, M. A. (2017). Metocognitive Support Accelerates Computer Assisted Learning for Novice Programmers. *Educational Technology & Society, 20*(3), 170–181

Norris, C. & Jackson, L. (1992). The effect of computer science instruction on critical thinking skills and mental alertness. *Journal of Research On Computing in Education, 24*(3), 329. http://eds.b.ebscohost.com/eds/detail/detail?vid=1&sid=c9a26212-f9a0-43af-bc2d-3e216f4979c2%40pdc-v-sessmgr01&bdata=Jmxhbmc9dHI%3d#AN=9610010003&db=f6h

OECD (Organisation for Economic Co-operation and Development). (2001). *Competencies for the knowledge economy, Chapter 4*, 100-118, received from: http://www.oecd.org/innovation/research/1842070.pdf

Ohler, J. (2005). The world of digital storytelling. *Educational leadership, 63*(4), 44-47.

Phan, H.P. (2009). Reflective thinking, effort, persistence, disorganization, and academic performance: A mediational approach. *Electronic Journal of Research in Educational Psychology, 7*(3), 927-952. http://www.investigacion-psicopedagogica.org/revista/articulos/19/english/Art_19_338.pdf

Polya, G. (1985). *How to solve it: A new aspect of mathematical method (Second edition).* Doubleday Anchor Books, Dobleday & Company, Inc. Garden City, New York, https://math.hawaii.edu/home/pdf/putnam/PolyaHowToSolveIt.pdf

Robin, B. (2016). The Power of Digital Storytelling to Support Teaching and Learning. *Digital Education Review, 30*, 17-29

Sarıtepeci, M. (2017). Ortaokul düzeyinde dijital hikaye anlatımının yansıtıcı düşünme becerisi üzerindeki etkisinin incelenmesine yönelik deneysel bir çalışma. *Bartın Üniversitesi, Eğitim Fakültesi Dergisi, 6*, 3, 1367-1384, Doi: 10.14686/buefad.337772

Strampel, K. & Oliver, R. (2007). Using technology to foster reflection in higher education. *Proceedings ascilite*, 973-982: Singapore. https://learnzone.loucoll.ac.uk/sportres/HE/Online%20lectures/Part%20B/Industrial%20Placement/Lesson%207%20-%20Reflective%20Practice/strampel.pdf

Su, A., Huang, C., Yang, S., Ding, T., & Hsieh, Y. (2015). Effects of annotations and homework on learning achievement: An empirical study of Scratch programming pedagogy. *Educational Technology & Society, 18*(4), 331-343.

Taylor, M., Harlow, A., & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia-Social and Behavioral Sciences, 8*, 561-570. Online:10.1016/j.sbspro.2010.12.078,

Uygun, K. ve Çetin, T. (2014). The effect of reflective thinking practices on academic achievement and attitudes in social studies education. *Researcher: Social Science Studies, 2*(2), 50-72, http://rssstudies.com/DergiTamDetay.aspx?ID=16&Detay=Ozet

Vatan Newspaper (2018). *Vahim Tablo*. Access date: 01.08.2019. https://www.haberler.com/vatan-gazetesi/

Yıldız Durak, H. The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. Techology, Knowledge and Learning, 25, 179–195 (2020). https://doi.org/10.1007/s10758-018-9391-y

Yılmaz, N. (2017). Determination of reflective thinking tendencies of 8th grade Turkish teachers and student. Master Thesis, Near East University Institute of Educational Sciences Curriculum and Instruction, KKTC. http://docs.neu.edu.tr/library/6588468234.pdf