



Participatory Educational Research (PER)
Vol. 6(2), pp. 26-36, December, 2019
Available online at <http://www.perjournal.com>
ISSN: 2148-6123
<http://dx.doi.org/10.17275/per.19.10.6.2>

The effects of programming education planned with TPACK framework on learning outcomes

Handan ATUN*

Department of Computer Education and Instructional Technology, Necmettin Erbakan University, Konya, Turkey

Ertuğrul USTA

Department of Computer Education and Instructional Technology, Necmettin Erbakan University, Konya, Turkey

Article history

Received:
21.05.2019

Received in revised form:
17.07.2019

Accepted:
18.07.2019

Key words:

Tpack; primary education;
programming

This study aims at investigating the effects of Programming Education Planned with TPACK (Technological Pedagogical Content Knowledge) Framework on middle school students' learning outputs within the ITS (Informational Technology and Software) course. Although TPACK is known as a teacher training program, this study demonstrates it can be used in K12 education. The sample of the research consists of 41 6th grade level students from a Turkish middle school. This study used a quasi-experimental research design which compares pre-test and post-test results for experimental and control groups. Data were collected through quantitative scales. The effects of programming education planned with TPACK framework on students' academic achievement, perception of problem-solving skills and computational thinking skills are investigated. According to the results, the means of academic achievement, problem solving inventory and computational thinking skill scale scores of the experimental group are significantly higher, which means TPACK framed lesson has a positive impact on learning outcomes. As a result of this study it can be concluded that matching technology that is suitable for the relevant content is crucial for learning, using appropriate technology is a good strategy for learning technology, higher order skills are improved by technology supported learning and academic achievement can be enhanced by using enriched activities in a technological environment.

Introduction

The rapid integration of technology into life in Information and Communication Technologies (ICT) era resulted in the use of it in all aspects of everyday life. Education and indeed instruction is one of the areas which is widely affected by ICT, and its positive influence in this regard is demonstrated in the related literature (Polly, Mims, Shepherd & İnan, 2010). It goes without saying that this very situation creates necessities for teachers to come up with novel methodologies as well as approaches which are compatible with

*Correspondency: handanatun@gmail.com

technological tools (Oster-Levinz & Klieger, 2010). Recently, many undergraduate programs have included the “Educational Technology” course in order to increase the effective use of technology in teaching and learning processes. However, it would not be wrong to state that teachers have not yet internalized the vital skills pertaining to technology in education as desired meaning that undergraduate programs are still not satisfactory for teacher candidates (Polly, Mims, Shepherd & İnan, 2010). This may be due to teachers' inability to successfully relate their pedagogical knowledge to their content knowledge. According to Shulman (1987), a teacher should have knowledge of content, general pedagogical knowledge, program knowledge, pedagogical content knowledge, student knowledge, educational content knowledge and knowledge of educational outputs, objectives and values. If technological knowledge is added to the findings of Shulman, the knowledge that the teacher should have according to the requirements of our age will be completed to the full extent. Mishra and Koehler (2006) constituted the Technological Pedagogical Content Knowledge-TPACK as an alternative in this manner. TPACK was the abbreviation of the known Technological Pedagogical Content Knowledge in the international sense. TPACK was later transformed into TPACK (Thompson & Mishra, 2007) so as to increase the effectiveness of skills of reading and memorizing. Instead of considering technology as a separate field, the TPACK framework draws attention to the relationships, interactions and the convenience and constraints of the teachers' knowledge of content, pedagogy and technology. In this model, content knowledge (CK), pedagogical knowledge (PK) and technological knowledge (TK) are the main information required for teacher development. Besides, when these areas are merged in pairs; Technological Pedagogical Knowledge (TPK), Technological Content Knowledge (TCK) and Pedagogical Content Knowledge (PCK) are formed three of which constitute the model: TPACK (Mishra & Koehler, 2006). Pedagogical content knowledge is related to the learning approach and also choice of the material that best suits the course content (Angeli & Valanides, 2005; Cox & Graham, 2009). Technological Content Knowledge is about what technology should be used to teach successfully and Technological Pedagogical Knowledge is relative to choosing the most appropriate IT tool according to the determined learning / teaching needs (Hu & Fyfe, 2010).

Mishra and Koehler (2006) defined TPACK as the basis for effective teaching by using technology. Besides, the meaning of TPACK includes; concept teaching using technology, pedagogical techniques that use technology through constructivist approaches to teach course content, knowledge of what makes concepts difficult or easy to learn and how these problems can be overcome by technology, knowledge of input and epistemological theories and knowledge of how technologies can be used to build on existing knowledge and lastly to develop new epistemologies or strengthen old ones (Mishra & Koehler, 2006). The ISTE (International Society for Technology in Education) (2007) standards, that 21st century students should have, include the “Digital Citizenship and Technology Practices and Concepts”. Students who have “Digital Citizenship” competency give importance to the legitimacy and accuracy of the information they have reached on the internet. They have a positive attitude towards using technology that supports learning and cooperation. Moreover, students should be able to use technological applications, understand pertinent concepts and use new technology to learn via transferring their prior knowledge. One of the ways these standards can be referred to during instruction is producing a lesson plan developed with the TPACK framework. Bearing in mind the pedagogy; the methods that will be resorted to while presenting the input, the content itself and the technology that will be used is crucial in that sense. In fact the harmony of all these three important components is essential as the more there is harmony the more effective and memorable the process becomes. Therefore, TPACK framework enables teachers to make effective lesson plans and at the same time enables

students to be successful and productive individuals (Lingenfelter, 2015).

According to Yadin, the programming course is thought to be the hardest course by the students (2011). Thusly to be able to eliminate this prejudice of learners it is decided to support a programming class with TPACK framework in middle (secondary) school. Since problem-solving and computational thinking skills are believed to be the efficient factors in learning/teaching of programming (Bers, Flannery, Kazakoff & Sullivan, 2014; Chao, 2016; Chen, Shen, Barth-Cohen, Jiang, Huang, & Eltoukhy, 2017; Kalelioglu, & Gülbahar, 2014,) they are selected as the determinative skills in this study. The aim of this study to that end is to determine the effects of programming education with the TPACK framework on the learning outcomes of secondary school students.

For this purpose, the following questions will be answered:

- (1) Is there any significant difference between the pre-test and post-test scores of the students in relation to their academic achievement?
- (2) Is there any significant difference between the pre-test and post-test scores of the students in relation to their problem-solving skills?
- (3) Is there any significant difference between the pre-test and post-test scores of the students in relation to their computational thinking skills?

Methods

This study, which examines the effect of TPACK-supported instruction in terms of different variables in the teaching of Information Technologies and Software course, is carried out by a quantitative method within the framework of quasi-experimental design.

Participants

The participants consist of 41 6th grade students who were studying in 2017-2018 academic year in a rural area in Turkey. The participants are selected due to the convenience and proximity factors for doing research. One of the classes was selected randomly as the control group and the other as the experimental group. In the experimental and control groups, the same teacher was the instructor.

Table 1. Descriptive features of participants.

Gender	Experimental Group	Control Group	Total
Male	9	10	19
Female	11	11	22
Total	20	21	41

As shown in the Table 1, the participants consist of 41 6th grade students (22 boys and 19 girls) aged between 11-13 years. The experimental group consists of 20 students (9 boys and 11 girls) and the control group consists of 21 students (10 boys and 11 girls).

Instructional program

In order to teach programming to 6th grade students in line with Ministry of National Education's curriculum, two different instruction programs are prepared; one of them for the experimental group, the other one is for the control group. The instruction programs consist of



14 targeted objectives (TTKB, 2018) characterizing student activity/desired output. The learner:

- Recognizes the interface and properties of the block-based programming tool. [Scratch (scratch.eba.gov.tr), Kodadia2023, Code.org, Codecademy etc. programming platforms are available].
- Describes the functions of a program presented in the block-based programming tool.
- Extracts errors from a program that is presented in a block-based programming tool.
- Develops a program presented in the block-based programming tool according to the given criteria.
- Creates programs that include linear logic structure.
- Tests programs that include linear logic structure and extracts errors.
- Creates programs that contain decision structure.
- Tests programs containing decision structure and extracts the errors.
- Creates programs with multiple decision structures.
- Tests programs that contain multiple decision structures and extracts their errors.
- Creates programs that contain the loop structure.
- Tests programs that include the loop structure and extracts errors.
- Selects the most appropriate decision structures to adapt an algorithm.
- Uses different programming structures to find solutions to complex problems.

The activities for experimental group of the instruction program were prepared based on TPACK Learning Activities for Computer Science (Carton, 2017), which were initially created through the inspiration of the TPACK Learning activities of Harris and Hofer (2009). Some of the activities of both experimental and control group lesson plans are given in the Table 2.

Table 2. TPACK learning activities used in lesson plan

Activity Type	Description	Coding Platform	Explanation/Knowledge Type
Peer feedback	Students receive and give feedback on programming and projects (Carton, 2017).	Scratch	Students login Scratch. They change the activity as they like and save it as a project and write to each other through Scratch. (TPK)
Identifying problems	Students identify real-world problems that can be solved in computer environment (Carton, 2017).	Blockly Games	Students gain an idea that animal classification can be carried out on computer. (TCK)
Debugging/Troubleshooting	Students fix problems in computer systems and systematically extract errors in code indexes (Carton, 2017).	Code.org	Students find and correct the errors made by their teachers in the activity. (TCK)
Testing artifacts	Students systematically test computational works to determine whether criteria and restrictions are met (Carton, 2017).	Blockly Games Scratch	Students estimate, test, and correct the code they need to write. (TPACK)

Simplifying complex code	Students change a part of the code to contain fewer codes without causing any changes (Carton, 2017).	Code.org	Students convert the long code that they write to loop code. (TCK)
Decomposing problems	Students divide problems into more manageable sub-problems (Carton, 2017).	Blockly Games	In order to write the required code, students first determine which codes are repeated in a cycle and then determine how many loops they have entered. (TPACK)
Evaluating problems	Students evaluate problems to determine that they can be solved in a computable way (Carton, 2017).	Paper	The students evaluate whether the problem can be transformed into codes. (PCK)
Discussing problems	Students discuss and ask clarifying questions about a problem's ability to be solved with a computational approach (Carton, 2017).	Code.org	The students discuss the algorithm as a group on the given codes. (PCK)
Creating artifacts	Students create computational artifacts to solve problems, express themselves, or complete tasks. (Carton, 2017).	Touch Develop Code.org	Students form a shape with codes. (TPACK)
Designing/Creating modules	Students design and create systems of interacting modules and abstractions (Carton, 2017).	Microsoft Small Basic	Students create a code that prints the text and that changes according to the situation on the computer screen. They create a project that covers all the learning process. (TPACK)

As shown in Table, 2 blockly, code.org, scratch, touch develop and small basic code platforms were used for experimental and control groups. Giving and receiving peer feedback, identifying problems, debugging/troubleshooting, testing artifacts, simplifying complex codes, decomposing problems, evaluating problems, discussing problems, creating artifacts, designing/creating modules are learning activities introduced by Carton (2017) as “interpret, improve and abstract” activity types.

Since the experimental and control group students were thought to own sufficient knowledge and skills of computer use, there was no need for a pre-study. The application covers a period of 6 weeks / 12 hours. The application was conducted in the Computer Laboratory with the computers, speakers, an interactive board and internet connection.

Data collection tools

The achievement test, problem solving inventory for children at elementary level and computational thinking scales are made use of to collect data. The achievement test for the pre-test and post-test use in the study was developed by the researcher with 40 questions related to the Problem Solving and Programming unit. Content validity and face validity of the test are examined by some experts in this field. As a result, the number of test items were



reduced to 25. The final version of the achievement test consists of 25 multiple choice questions, including eight knowledge, four comprehension, four applications, five analyses and four syntheses, pursuant to Bloom's Taxonomy. The purpose of using this achievement test is to control the students' knowledge about "Programming, Linear Logic Structure, Decision Structure, Multiple Decision Structure, Loop Structure, Complex Problem Structure" in terms of pre-test and post-test, and to reveal the cognitive differences between the groups.

With a view to measuring students' perceptions of problem-solving skills, Problem Solving Inventory for Children at Elementary Level was developed by Serin, Bulut Serin and Saygılı (2010). The scale consists of 24 items which are developed in 5-point Likert type scale. The scale consists of three factors: Confidence in Problem Solving, Self-Control and Avoidance. The internal consistency coefficient of the scale is 0.80. The scale is original and the first inventory of this sort developed for the field of education in Turkey.

In order to measure the students' computer thinking skills, [The Scale of Computer Thinking Skill Levels for Secondary Level Level] developed by Korkmaz, Çakır and Özden (2016) was used. The scale consists of 22 items developed in 5-point Likert type scale. The scale consists of 5 different factors: Creativity, Algorithmic Thinking, Collaboration, Critical Thinking and Problem Solving. The internal consistency coefficient of the scale is 0.81

Data analysis

The independent samples t-test is used to examine the differentiation between two groups which are independent from each other and equal in average (Ross, 2014). The quantitative measurement tools were applied to all groups of the 6th graders in the schools. Due to the fact that almost all of the achievement scores were the same, class A was selected as the experimental group and class B was selected as the control group. The assumption of normality required for parametric tests was tested before analyzing students' t-test results of the pre-test and post-test scores in the experimental and control groups.

In order to test the effect of TPACK-supported instruction program on pre-test post-test control group design, independent samples t-test was used in SPSS. For independent samples, it was investigated if differences between t-tests of two unrelated sampling means was significant or not. Before T-test analysis, whether the variances are homogenous or not is also ensured. For this, p value should be greater than 0.05. According to all pre-test and post-test results of the experimental and control groups, it was concluded that the variances were homogeneous ($p > .05$).

In the analysis, the difference between the difference points of the experimental and control groups were examined by p value. It is then concluded that; there is a significant difference when $p < .05$ and there is no significant difference when $p > .05$.

Results

Research findings related to students' academic achievement

1. Is there any significant difference between the pre-test scores of the experimental and control groups in relation to their academic achievement?

Table 3 - Pre-test analysis of experimental and control groups in terms of academic achievement.

Group	N	X	S	sd	t	p
Experimental	20	3,55	1,36	39	0,058	0,954
Control	21	3,52	1,54			

Based on these findings, it can be said that experimental and control group students are not different from each other in terms of academic achievement and their mean scores are almost the same.

2. Is there any significant difference between the post-test scores in relation the academic achievement of the experimental and control groups?

Table 4 - Post-test analysis of experimental and control groups in terms of academic achievement.

Group	N	X	S	sd	t	p
Experimental	20	18,55	4,80	39	2,45	0,019
Control	21	14,29	6,21			

As shown in Table 4, it is seen that there is a significant difference between the post-test average scores of students ($t = 2.45$, $p < .05$). The average number of questions that the experimental group answered correctly in the post-academic achievement test was $x = 18.55$, and the average number of questions that the control group answered correctly in the post-academic achievement test was $x = 14.29$. The mean score of the control group was $x = 74,20$ and the mean score of the control group was found as $x = 57,16$.

Based on these findings, it would be fair to articulate that the average of the experimental group was significantly higher than the average of the control group, and TPACK-supported education improves the academic achievement of the students.

Research findings related to students' problem-solving skills

1. Is there any significant difference between the problem-solving inventory pre-test scores of the experimental and control groups?

Table 5 – Pre-test analysis of experimental and control groups in terms of problem-solving skills.

Group	N	X	S	sd	t	p
Experimental	20	2,34	0,67	39	-1,46	0,152
Control	21	2,71	0,92			

Based on these findings, it can be said that experimental and control group students are not different from each other in terms of problem-solving skills.

2. Is there any significant difference between the problem-solving inventory post-tests of the experimental and control groups?

Table 6 – Post-test analysis of experimental and control groups in terms of problem-solving skills.

Group	N	X	S	sd	t	p
Experimental	20	3,63	0,60	39	2,97	0,005
Control	21	2,93	0,87			



As shown in Table 6, it is seen that there is a significant difference between the post-test mean scores of the students ($t = 2.97$, $p = <.05$). While the mean of the experimental group responses to the problem-solving inventory was $x = 3,63$, the mean of the responses belonging to the control group was $x = 2,93$.

Based on these findings, it can be said that the mean of the experimental group is significantly higher than the mean of the control group, that is TPACK-supported education improves the students' problem-solving skills perception.

Research findings related to students' computational thinking skills

1. Is there any significant difference between the pre-test scores of the students in relation to the computational thinking skills?

Table 7 – Pre-test analysis of experimental and control groups in terms of computational thinking skills.

Group	N	X	S	sd	t	p
Experimental	20	2,56	0,73	39	-2,00	0,058
Control	21	2,97	0,83			

Based on these findings, it can be said that experimental and control group students do not differ from each other in terms of computer thinking skills.

2. Is there any significant difference between the post-test scores of the students in relation to their computational thinking skills?

Table 8 – Post-test analysis of experimental and control groups in terms of computational thinking skills.

Group	N	X	S	sd	t	p
Experimental	20	3,87	0,69	39	2,55	0,015
Control	21	3,29	0,77			

As shown in table 7, it is seen that there is a significant difference between the students' post-test mean scores of computational thinking skills ($t = 2.55$ $p <.05$). The average of the responses of the experimental group to the computational thinking skill scale was $x = 3.87$, and the average of the answers of the control was $x = 3.29$.

Based on these findings, the mean of the experimental group seems to be significantly higher than the average of the control group.

Conclusion

The arrival of educational technologies in our lives has paved the way for the integration of technology into education. Here, the integration of technology into education means not only adding technology to education but also making it an integral part of all the educational processes in accordance with the learning objectives. What is more, the key phases i.e. planning, evaluation and execution should go hand in hand with technology. Thence technology integration depends on teachers' knowledge of technology, pedagogy and content knowledge (Pierson, 2001) to a large extent. Mishra and Koehler (2006) formed the framework of Technological Pedagogical Content Knowledge (TPACK) to address this. Since then technology-based teaching programs has been positively influenced by the TPACK framework (Angeli & Valanides, 2009) and the learning / teaching activities have begun to be

based on the TPACK framework (Harris and Hofer, 2009). It is revealed that, TPACK is an effective framework not only for teacher education but also for primary and secondary education and it can be used for the benefit of both students (primary/secondary/high school) and teachers. There are studies found in the literature investigating the effects of course plans on teachers and students which are formed by TPACK framework (Brill, Listman & Kapila, 2015; Kontkanen et al., 2017; Hofer & Harris, 2010; Wetzel & Marshall, 2011; Doering et al., 2014; Aisyah, 2013). However, no research witnessed regarding the use of TPACK framework in the computing/informatics/programming courses curricula examining learning of students in the secondary school. In order to fill this gap diagnosed, the purpose of this study is to investigate the effects of Information Technologies and Software course created by TPACK framework on students. Since the programming course is generally seen as the most difficult course to be dealt with (Yadin, 2011), it has been decided to do this research on Programming subject of Problem Solving and Programming unit. As a result of the research conducted with the pre-test, post-test, and semi-experimental design with control group, the following findings and remarks are shared.

There is no significant difference between the pre-tests of the experimental and control groups according to their academic achievement, indicating that both groups are identical in terms of academic achievement in programming subject. In the t-test analysis of post-tests, a significant difference was found in favor of the experimental group. Similarly, in a research conducted, it was found out that the lesson plans prepared in the framework of TPACK were effective in teaching abstract topics in mathematics and science (Brill, et al., 2015) and in teaching of geography (Doering, et al., 2014). There is an improvement in problem-solving skills of both groups' compared pre and post-test results. However, experimental group students' problem-solving skills developed significantly. Dealing with the programming course already contributes to improved problem-solving skills and computational thinking skills (Bers, Flannery, Kazakoff & Sullivan, 2014), yet TPACK framework adds onto this effect (Tee & Lee, 2011). Similar to the findings in problem-solving t-tests, a significant difference was found in t-test scores of computational thinking skills in favor of the experimental group. The higher-order skills such as problem solving and computational thinking were high in the experimental group, bearing resemblance to studies of nature. In one of these studies, it was found that the use of TPACK model in education improves students' critical thinking and computer literacy (Aisyah, 2013), while in another, TPACK model helps students to develop skills such as critical thinking, creativity and collaborative communication (Brown, et al., 2011).

According to these results, using TPACK as a framework to deliver computing/programming/informational technology courses can be suggested considering the potential benefits for students in terms of academic achievement, problem-solving and computational thinking skills. Taking into account appropriate educational settings for the 21st century, using and improving TPACK framework for primary and secondary school students is fundamental.

There are some limitations in this research that need to be mentioned. Owing to the fact that the research is limited to only 41 people in total, it is recommended to keep the sample wider in future research. At the same time, it is useful to keep the age range wider in possible future research for the research is limited to 6th graders only. Moreover, the effects of creating a lesson plan with TPACK framework on teachers are not touched upon in this paper. In future research, challenges faced alongside struggles and accomplishments while using TPACK as a framework can be examined.



References

- Aisyah, A. R. (2013). The Development of Working Design through Characterized Technology Pedagogy and Content Knowledge in the Elementary Schools' Instructional. *Procedia-Social and Behavioral Sciences*, 103, 1016-1024.
- Angeli, C., and Valanides, N. (2005). Preservice elementary teachers as information and communication technology designers: An instructional system design model based on an expanded view of pedagogical content knowledge. *Journal of Computer Assisted Learning*, 21(4), 292-302.
- Angeli, C., & Valanides, N. (2009). Epistemological and methodological issues for the conceptualization, development, and assessment of ICT-TPCK: Advances in technological pedagogical content knowledge (TPCK). *Computers & Education*, 52(1), 154-168.
- Bers, M. U., Flannery, L., Kazakoff, E. R., and Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.
- Brill, A. S., Listman, J. B. and Kapila, V. (2015). Using robotics as the technological foundation for the TPACK framework in K-12 classrooms. In 2015 ASEE Annual Conference & Exposition (pp. 26-1679).
- Brown, C. A., Neal, R. E. & Fine, B. (2011). Using 21st Century Thinking Skills Applied to the TPACK Instructional Model. Paper presented at the annual meeting of the AECT International Convention, Hyatt Regency Jacksonville Riverfront, Jacksonville, FL, Nov 08, 2011
- Carton, R. (2017). TPACK learning activity types for secondary computer science courses. *Graduate Research Papers*. 139.
- Chao, P. Y. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education*, 95, 202-215.
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162-175.
- Cox, S., and Graham, C. R. (2009). Using an elaborated model of the TPACK framework to analyze and depict teacher knowledge. *TechTrends*, 53(5), 60-69.
- Doering, A., Koseoglu, S., Scharber, C., Henrickson, J., & Lanegran, D. (2014). Technology integration in K-12 geography education using TPACK as a conceptual model. *Journal of Geography*, 113(6), 223-237.
- Harris, J., and Hofer, M. (2009). Instructional planning activity types as vehicles for curriculum-based TPACK development. In Society for Information Technology & Teacher Education International Conference (pp. 4087-4095). Association for the Advancement of Computing in Education (AACE).
- Hofer, M., and Harris, J. (2010). Differentiating TPACK development: Using learning activity types with inservice and preservice teachers. In Society for Information Technology & Teacher Education International Conference (pp. 3857-3864). Association for the Advancement of Computing in Education (AACE).
- Hu, C., and Fyfe, V. (2010). Impact of a new curriculum on pre-service teachers' Technical, Pedagogical and Content Knowledge (TPACK). Curriculum, technology and transformation for an unknown future: Proceedings ascilite Sydney, 185-189.
- ISTE (International Society for Technology in Education). (2007). National educational technology standards and performance indicators for students. Eugene, OR: International Society for Technology in Education

- Kalelioglu, F., & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33-50.
- Kontkanen, S., Dillon, P., Valtonen, T., Eronen, L., Koskela, H., and Väisänen, P. (2017). Students' experiences of learning with iPads in upper secondary school—a base for proto-TPACK. *Education and Information Technologies*, 22(4), 1299-1326.
- Korkmaz, Ö., Çakır, R., ve Özden, M. Y. (2016). Bilgisayarca Düşünme Beceri Düzeyleri Ölçeğinin (Bdbd) Ortaokul Düzeyine Uyarlanması [Adaptation of Computer Thinking Skill Levels Scale (CTS) to Secondary School Level]. *Gazi Eğitim Bilimleri Dergisi[Gazi Educational Sciences Journal]*, 1(2).
- Lingenfelter, B. (2015). Building Lessons with Web 2.0 & TPACK. American College of Education.
- Mishra, P. and Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *The Teachers College Record*, 108(6), 1017-1054.
- Oster-Levinz, A., and Klieger, A. (2010). Online tasks as a tool to promote teachers' expertise within the Technological Pedagogical Content Knowledge (TPACK). *Procedia-Social and Behavioral Sciences*, 2(2), 354-358.
- Pierson, M. E. (2001). Technology integration practice as a function of pedagogical expertise. *Journal of research on computing in education*, 33(4), 413-430.
- Polly, D., Mims, C., Shepherd, C. E., & Inan, F. (2010). Evidence of impact: Transforming teacher education with preparing tomorrow's teachers to teach with technology (PT3) grants. *Teaching and Teacher Education*, 26(4), 863-870.
- Ross, S. M. (2014). Introduction to probability and statistics for engineers and scientists. Academic Press.
- Serin, O., Bulut Serin, N. B., ve Saygılı, G. (2010). İlköğretim düzeyindeki Çocuklar için Problem Çözme Envanteri'nin (ÇPÇE) geliştirilmesi [Developing Problem Solving Inventory for Children at Primary Level]. *İlköğretim Online*, 9(2).
- Shulman, L. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review*, 57(1), 1-23.
- Tee, M. Y., & Lee, S. S. (2011). From socialisation to internalisation: Cultivating technological pedagogical content knowledge through problem-based learning. *Australasian Journal of Educational Technology*, 27(1).
- Thompson, A. D., and Mishra, P. (2007). Breaking news: TPCK becomes TPACK! *Journal of Computing in Teacher Education*, 24(2), 38, 64.
- TTKB (2018). Bilişim teknolojileri ve yazılım dersi öğretim programı (ortaokul 5 ve 6. Sınıflar). [Information technologies and software course curriculum (5th and 6th grades)].
- Wetzel, K., and Marshall, S. (2011). TPACK goes to sixth grade: Lessons from a middle school teacher in a high-technology-access classroom. *Journal of Digital Learning in Teacher Education*, 28(2), 73-81.
- Yadin, A. (2011). Reducing the dropout rate in an introductory programming course. *ACM inroads*, 2(4), 71-76.