Journal of Educational Technology & Online Learning
Volume 6 | Issue 1 | 2023
http://dergipark.org.tr/jetol

Research trends in programming education: A systematic review of the articles published between 2012-2020

Atajan Rovshenov[a] *, Fırat Sarsar [a]

[a] Ege University, Turkiye.

Suggested citation: Rovshenov, A. & Sarsar, F. (2023). Research Trends in Programming Education: A Systematic Review of the Articles Published Between 2012-2020. *Journal of Educational Technology & Online Learning, 6*(1), 48-81.

Highlights

- The United States and Turkiye conduct the majority of education programming research studies.
- In programming education, quantitative methods are the most frequently used research methods.
- Programming education positively impacts students' learning and academic success, as well as their computational thinking abilities.

Article Info: Review Article

Keywords: *Programming education, programming teaching, Computer science education, Systematic review*

Abstract

This study examines the methodological dimensions of programming education articles published in educational sciences journals indexed in SSCI by exploring their general trends. To do this, 162 articles published between January 2012 and February 2020 in 30 international journals indexed in SSCI were analyzed with a systematic review method using the "Educational Technology Publication Classification Form" as a data collection tool. The results revealed that most of the studies in this field were conducted in the United States and Turkiye. The number of these studies has increased since 2015, and those studies were carried out using quantitative research methodology. Mostly questionnaires and achievement tests were used as a data collection tool, a convenience sampling method was used, and descriptive analyses were adopted to analyze the data. As a result, the articles examined in this study showed that programming education positively contributes to learners' learning and success levels and the development of their computational thinking skills. We believe that these results will shed light on future studies related to programming education.

1. Introduction

The rapid development of information and communication technologies (ICT) has changed the characteristics expected from individuals, and the lifestyle and social structure have transformed. Similarly, shaping the existing knowledge and skills according to these changing conditions and producing different products. Furthermore, the ability of individuals to develop new and original products has increased the significance addressed to the ICT sector (Lee & Lee, 2015). Having a say in ICT is possible if original and unique technological products are developed. Therefore, raising individuals who do not only consume but also produce technology comes to the fore as a goal that developed countries put more emphasis on (Dağhan, Kibar, Çetin, Telli & Akkoyunlu, 2017). Because computing and programming skills have a

* Corresponding author. Department of Computer Education and Instructional Technology, Ege University, Turkiye.
e-mail address: rovshenovatajan@gmail.com

Doi: http://doi.org/10.31681/jetol.1201010
Received 8 Nov 2022; Revised 27 Dec 2022; Accepted 1 Jan 2023
ISSN: 2618-6586.

crucial role in achieving this goal, many countries which believe that programming should be taught starting from an early age renew their curricula so that students can acquire this skill (Gülbahar & Kalelioğlu, 2018; Webb et al., 2018; Wohl, Beck & Blair, 2017).

Before discussing the current status of programming education, it is important to agree on the terminology. The concept of programming is defined as "having the expected tasks and operations performed as a result of entering the user commands created within the framework of certain syntax rules to the computer through a programming language and make it function properly" (Butterfield, Ngondi & Kerr, 2016, pp. 24).

The programming process consists of the following steps (Veena & Gowrishankar, 2018):

- *Identify the problem*: The user identifies the problem aimed to be solved with the software created as a result of observation. The factors and variables that cause the problem are determined at this stage.
- *Seek appropriate solutions*: Flow charts are created to solve the problems.
- *Develop the codes*: The programming language suitable for the problem's solution is determined in line with the flow chart. Then, codes are developed using a programming language.
- *Interpret and compile*: Translating these codes into a language that the computer can understand. Then compilation is carried out, and the program is run.

Detect and eliminate the errors: After running the program, examinations are carried out to eliminate the syntax and logic errors. After making sure that the errors are fixed, the compilation process is conducted again.

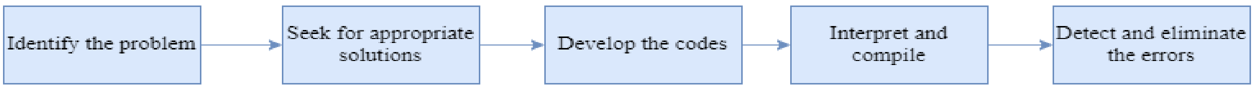

Fig.1. Steps followed in the programming process (Veena & Gowrishankar, 2018).

When we look at Figure 1, it is seen that the programming process is comprehensive, and there are metacognitive abilities that need to be trained and acquired to learn to program.

1.1.Programming Education or Computer Science Integration

When Figure 1 is examined, it is seen that the theoretical information about algorithm development, the rules of the programming language, and the practice should be practiced to develop a program. In addition, individuals need to have various cognitive skills to solve problems with programming. Individuals who would like to learn to program should have higher-order thinking skills to use ICT effectively, design algorithms, and know the programming language. However, there is a consensus among students, teachers, and experts that programming education is not easy (Gurer, Cetin & Top, 2019; Qian et al., 2020; Scherer, Siddiq & Sánchez Viveros, 2019). Studies in the literature show that programming lessons are difficult for learners and teachers (Cheah, 2020; Qian & Lehman, 2017). In recent years, where programming knowledge has increased day by day, examining the difficulties learners and teachers face is important in developing new teaching techniques and conducting scientific research.

Gomes and Mendes (2007) categorized the difficulties that students may have during programming education as follows:

- *Teaching Approaches:* Dynamic terms are prepared with inert materials, which are not designed in line with the learner's styles. In addition, the teachers prefer theoretical knowledge and content rather than improving students' problem-solving skills through practice.

- *Learning Methods:* Students use inappropriate learning methods in their self-study to improve their academic programming success, and they do not have practice related to programming.
- *Thinking Skills:* Many students' problem-solving skills are not enough to create algorithms and understand the logic behind programming.
- *The Nature of Programming:* Programming has content that requires a very high abstraction level, and the programming languages have a very complex syntax.
- *Psychological Factors:* Students' attitudes towards learning programming are low, and they often have to learn programming during the academically busy times of the term.

The difficulties encountered in learning programming have led to the development of new teaching approaches and techniques by researchers. For example, researchers have proposed various computing and programming teaching approaches such as computer science (CS) unplugged, physical computing, visual computing, and game-based learning to reduce the difficulties experienced by learners in programming education and motivate them to learn to program (Battal, Afacan Adanir & Gulbahar, 2021; Benitti, 2012; Caeli & Yadav, 2019; Kalelioglu & Sentence, 2020; Lindberg, Laine, & Haaranen, 2019; Noone & Mooeny, 2018; Kelleher & Pausch, 2005; Yesharim & Ben-Ari, 2018). The aim of the approaches developed to make programming education, which is considered complex, easier, is to embody abstract information, to teach students programming logic by showing complex syntaxes in programming languages step by step (Hundt, Schlarb & Schmidt, 2017; Salleh, Shukur & Judi, 2013; Tuparov, Tuparova & Jordanov, 2014). As a result, the number of studies that propose solutions to the challenges encountered in teaching programming has increased (Robins, Rountree & Rountree, 2003).

Developing different teaching techniques for programming and extending programming education at the K-12 and university levels is not enough for well-structured programming education. Researchers should examine all pedagogical factors to guide instructors, researchers, teachers, and industry members in conducting qualified programming education. For these reasons, a systematic review must be conducted, which will offer the following general methodological trends and outcomes of programming education in educational science literature.

This article presents a review of research trends in programming education. It is based on examining 162 empirical research articles published in eminent educational science journals. The novelty of this work is represented by programming education research in the context of demographic, methodological, and study results. Moreover, knowing the general and methodological research trends in programming education could assist researchers and practitioners in planning future studies and serve as a resource for policymakers when designing computer science education programs. Additionally, this study provides new research direction issues identified from the review. The rest of the paper is organized in the following manner. The next section describes the methodology of conducting this systematic review by demonstrating the basic stages of this research. Then, the findings demonstrated related to research questions. After that, the discussion and analysis are presented. Finally, limitations and new research directions for future research are described.

2. Methodology

This research, which examines the studies on programming education published in 30 international journals indexed in SSCI, was conducted using a systematic review. The systematic review is a study that aims to structure the research area by classifying the studies published on a particular subject and identifying new research gaps (Borrego, Foster & Froyd, 2014; Petticrew & Roberts, 2008). In this study, a systematic review was adopted. It organizes similar data within the framework of specific concepts and themes and transforms them into a form that the readers can understand. Borrego, Foster & Froyd (2014) stated that systematic review studies generally consist of standard stages. In this respect, they suggested some steps for the regular conduct of systematic review studies applied in the current research.

The study consists of the following steps: (1) developing research questions, (2) selecting the journals to be included in the systematic review, (3) selecting the manuscripts related to the subject from those journals, (4) determining the selection criteria for the articles to be examined, and (5) ensuring validity and reliability. Figure 2 provides information about the phases of the study.

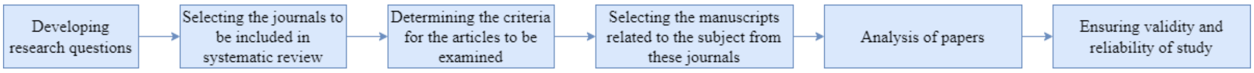

Fig.2. Flowchart of the research

2.1. Developing Research Questions

The first stage of the review process defines the study's aim and develops appropriate research questions.

1. What are the demographic characteristics of research conducted on programming education?

1.1. In which years were these studies conducted most?
1.2. In which countries were those studies conducted the most?

2. What are the methodological trends of the research conducted on programming education?
   2.1. Which methods were used in the studies?
   2.2. At what levels of education were the studies conducted?
   2.3. What is the sample size in the studies?
   2.4. Which sampling methods were used in the studies?
   2.5. Which data collection tools were used in the studies?
   2.6. Which data analysis methods were used in the studies?
   2.7. What are the dependent variables examined in the studies?

3. What are the results of the studies?
4. What are the limitations of the studies?
5. What are the future research implications in the studies?

2.2. Selecting the Journals to be included in the Systematic Review

One of the essential stages of systematic review is the selection of articles in line with the research problem. This study examines the demographical, methodological dimensions and the results of the programming education studies published in 30 educational sciences journals indexed in SSCI between January 2012 and February 2020. Table 1 shows the selected educational sciences journals indexed in SSCI between January 2012 and February 2020.

Table 1.

Information about selected educational sciences journals indexed in SSCI between January 2012 and February 2020.

	Title of Journal	Number of Selected Related Publications
1	Asia-Pacific Education Researcher	7
2	Australasian Journal of Educational Technology	2
3	British Journal of Educational Psychology	1
4	British Journal of Educational Technology	8
5	Contemporary Educational Psychology	1
6	Computers & Education	32
7	Comunicar. Media Education Research Journal	1
8	Cultura and Educacion	1
9	Educational Technology & Society	10
10	Education and Science	1
11	ETR&D-Educational Technology Research and Development	5
12	IEEE Transactions on Learning Technologies	8
13	Interactive Learning Environments	12
14	International Journal of Educational Technology in Higher Education	3
15	International Review of Research in Open and Distributed Learning	4
16	Journal of Computer Assisted Learning	11
17	Journal of Creative Behavior	2
18	Journal of Educational Computing Research	33
19	Journal of Research in Science Teaching	1
20	Journal of Research on Technology in Education	4
21	Journal of Science Education and Technology	4
22	Journal of Teacher Education	1
23	Journal of The Learning Sciences	1
24	Learning, Media and Technology	1
25	Studies in Educational Evaluation	1
26	Thinking Skills and Creativity	2
27	Journal of Special Education	1
28	Telematics and Informatics	2
29	Cognition and Instruction	1
30	Innovations in Education and Teaching International	1
	Sum	**162**

2.3. Determining the inclusion/ exclusion criteria and selecting the manuscripts related to the subject from journals

Table 2 shows the information about the criteria list for reviewing selected journals. The keywords shown in column 1 of the table have been chosen specifically to access publications related to programming education. In the 2nd column, the information about the education levels of the studies conducted on programming education is given. All education levels were included in the research to provide detailed information about the status of the research subject at the education levels. In addition to filtering by keywords, the publication range of the studies was selected as 2012-2020. The reason for choosing 2012-2020 is to provide up-to-date results by examining the studies published in recent years. Due to the inadequate number of studies related to programming education, the articles were not selected according to any referee evaluation criteria. The articles that met the above criteria and were published in the journals were included in this study.

As a result of the search according to these criteria, 162 articles were included in the systematic review (See the Appendix A for selected studies).

Table 2.

Information about criterion list for reviewing selected journals.

Keywords	Education Level	Time Span	Type of Publication
children's programming	Pre-School	January 2012- February 2020	Experimental Studies
computing education	Elementary		
computer science education	Secondary		
programming teaching	High-School		
programming instruction	University		
pair programming			
novice programming			
introductory programming			
teaching programming concepts			
block-based programming			
programming training			

2.4. Analysis of papers

In this study, to collect data, "Educational Technologies Publication Classification Form" was developed by Goktas, Kucuk, Aydemir, Telli, Arpacik, Yildirim & Reisoglu (2012) was used with the permission of the first author via e-mail. "Educational Technologies Publication Classification Form" was prepared as a draft by the research group, and then it was examined by an expert opinion and a foreign language expert. The data collection tool was revised according to expert opinions, and a reliability test was performed. The form consists of 7 sections: general information about the article, the subject of the article, the method of the article, data collection tools, sample, and data analysis methods. This paper cited over 150 times according to Google Scholar data, in which this form was published as of December 2022.

The data obtained from the articles examined by the systematic review were analyzed using descriptive statistical methods (percentage and frequency). In line with the data collection tool, the frequencies of the data and the percentages depending on these frequencies were calculated to correspond to each research question. The numerical data were presented using tables and graphs.

2.5. Ensuring validity and reliability of the study

To ensure the validity and reliability of the study, a systematic process was followed by the researchers, especially during data collection and analysis. Validity is related to how accurately, appropriately, and meaningfully the developed measurement tool measures the variable (Wallen & Fraenkel, 2013). To ensure the validity of this study, a table was used by the researcher to record the data collected in addition to the data collection tool. For providing internal validity of the research results, the findings obtained in the studies examined were used without adding any comments and were described as they are. An expert examined this table in instructional technology, and it was revised according to the expert's suggestions.

External validity is the generalizability of the results obtained within the scope of the research to the population (McMillan & Schumacher, 2001). To ensure the study's external validity, the articles examined were based on the manuscript selection criteria determined by the researcher, and all studies that met these criteria were included in the analysis. Reliability is the consistency of the results obtained within the scope of the research (Krippendorff, 2004). To ensure reliability in the study, two experts in educational sciences facilitated the researchers in determining which articles should be included. Then, the experts did an independent search, the data obtained were compared, and the articles were chosen. Afterward, the researchers came together and compared the analyses, discussions were held until a consensus was reached, consistency was ensured in the analyses' statements, and the analyses took their final form.

Then, the data obtained from these articles were processed into the form prepared by the researcher, and an instructional technology field expert checked the data related to each article. Their accuracy was confirmed, and necessary corrections were made.

3. Findings

The data collected using the "Educational Technologies Publication Classification Form" were analyzed based on the research questions. The findings are presented below in parallel with the research questions.

3.1. Findings of the Demographic Characteristics of Studies on Programming Education

Examining the distribution of studies on programming education by years and the countries in which they were conducted will be helpful for researchers in this field and teachers who teach computer science at different education levels. The data on the distribution of the reviewed studies by countries and years in which they were published are depicted in Figure 2 and Figure 3.

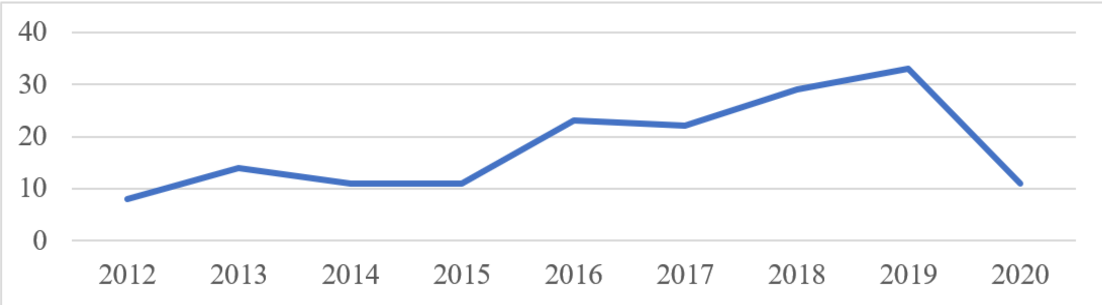

Fig. 3. Distribution of the studies on programming education by years.

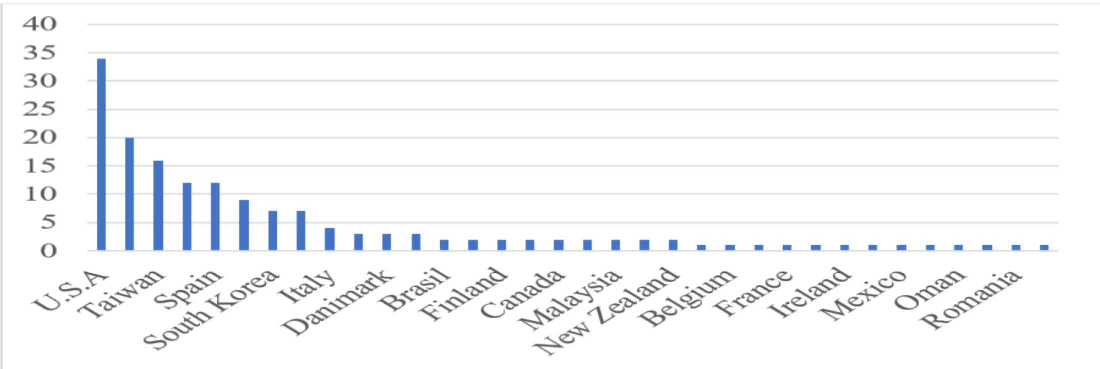

Fig. 4. Distribution of the studies on programming education by country.

When Figures 2 and 3 are analyzed, it is clear that there is an increase in the number of studies on programming education. The countries where these studies are conducted most are the USA, Turkiye, and Taiwan, respectively.

3.2.1. Findings of the Methodological Tendency of the Studies in Programming Education

Analysis of the methodological tendencies of the studies in programming education is important in terms of estimating which research method is common in the literature, the effect of the methods used on the results of the research, and the probable limitations that will arise in the studies to be conducted on the similar subjects. Figure 5 depicts the numerical data related to research methods on programming education.

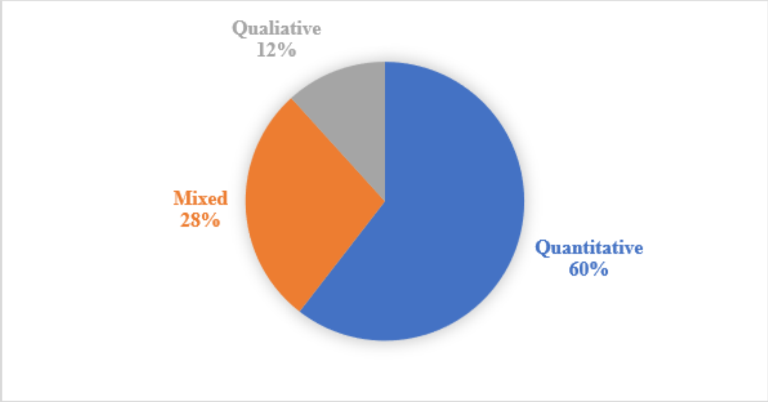

Fig. 5. Methodological tendencies of the studies in programming education.

It was found that the researchers mostly prefer the quantitative research method (60%) in their studies on programming education. Secondly, they also use a mixed research method (28%), and the qualitative research method is the least preferred one.

3.2.2. Findings of the Sample Size of the Studies in Programming Education

Sample selection methods, sample size, and the levels of education in which the studies are conducted significantly impact the study results. Researchers can determine the sample size and the level of education by looking at the variables in similar studies (Wallen & Fraenkel, 2013). Therefore, it is important to examine and interpret these characteristics conducted with a systematic review for future research. In this regard, sample selection methods, sample size, and education levels in which the studies are conducted were analyzed (See Table 3).

Table 3.

Findings of the Sample Size and Numbers of Studies in Programming Education.

	1-10	11-30	31-100	101-300	301-1000	1000 and over	**Total**
Pre-school	2	4	5	2	-	-	**13**
Primary school (1-4)	3	2	3	8	-	-	**16**
Middle school (6-8)	5	3	13	11	3	-	**35**
High school (9-12)	-	1	8	3	1	-	**13**
Undergraduate	1	7	37	21	7	2	**75**
Graduate (Teachers)	-	2	4	1	1	1	**9**
Faculty Members	-	-	-	1	-	-	**1**
Total	**11**	**19**	**70**	**47**	**12**	**3**	**162**

Table 3 shows that the educational level in which the studies were conducted mostly on programming education is at the K-12 level. However, most of the studies were conducted with students at the undergraduate level. In addition, the participants at graduate level studies are entirely teachers. On the contrary, it was found that the researchers do not usually prefer the sample group, which consists of faculty members. It was also found that the highest number of participants in the studies on programming education is 31-100. This is followed by the studies conducted with 101-300 participants. However, the number of studies with 1-10, 11-30, 301-1000, and more than 1000 participants is relatively low.

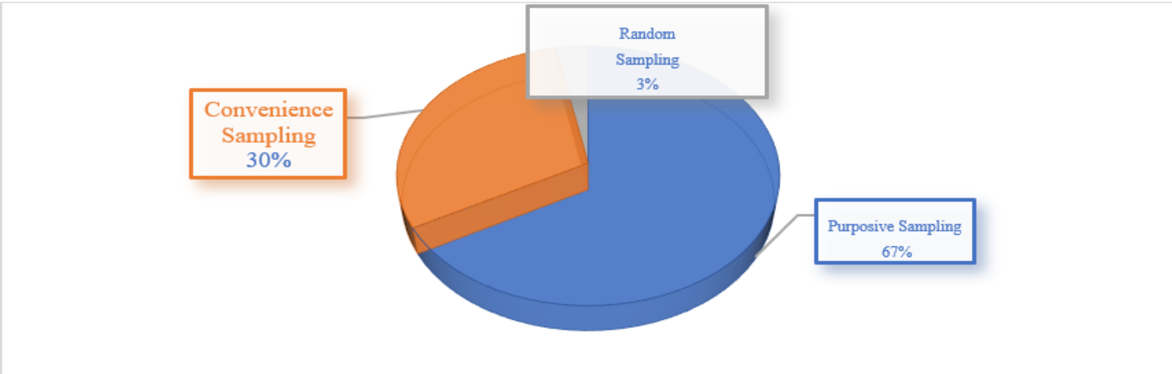

Fig. 6. Findings of the sample selection methods in programming education.

3.2.4 Findings of the Dependent Variables Examined in the Studies related to Programming Education

Figure 7 shows the dependent variables in the studies conducted in programming education.

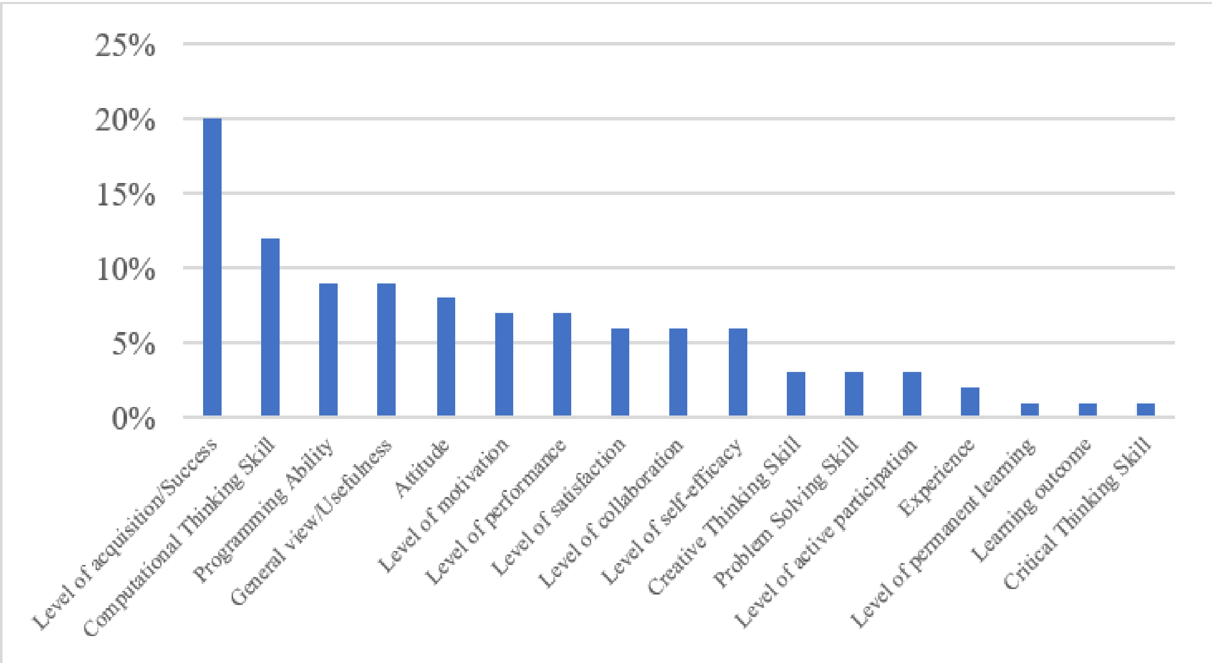

Fig. 7. Findings of the dependent variables examined in studies programming education.

Figure 7 shows that the most common dependent variable used in the studies in programming education is the level of acquisition/success level (20%). On the contrary, the level of permanent learning (1%), learning outcome (1%), and critical thinking skills (1%) are the dependent variables that were examined least. In addition, more than one variable was examined in 82 of the 162 studies reviewed.

3.2.5. Findings of Data Collection Tools Used in the studies in Programming Education

Examining the data collection tools used in the studies in programming education may contribute to the development of new data collection tools for the related studies. Figure 8 shows the findings related to the data collections tools examined in the study. Surveys are the most common data collection tools preferred by researchers as one of the quantitative research methods. Secondly, the achievement tests were used as a data collection tool. The least preferred data collection tool is attitude, perception, personality, or ability tests.

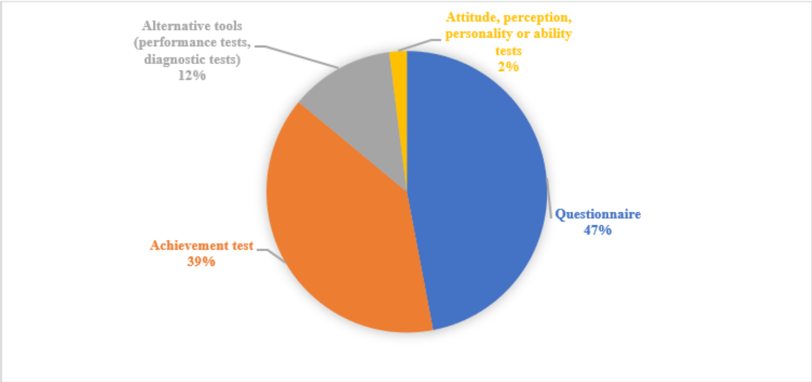

Fig. 8. Findings of quantitative data collection tools used in the studies in programming education.

Figure 9 shows the information about qualitative data collection tools used in the studies in programming education. It is clear from the figure that the most common qualitative data collection method used by the researchers was an interview (41%) and recordings (35%). On the contrary, observation (23%) is the least preferred qualitative data collection tool.

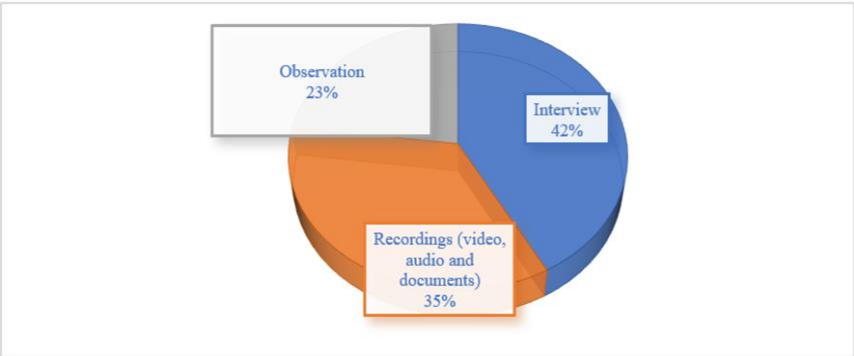

Fig. 9. Findings of qualitative data collection tools used in the studies.

The percentages of qualitative data collection tools used are higher than the number of studies conducted using the quantitative research method. Apart from the studies carried out with the qualitative research method, the number of studies conducted with the mixed research method.

3.2.6. Findings of the Data Analysis Methods Used in the Studies in Programming Education

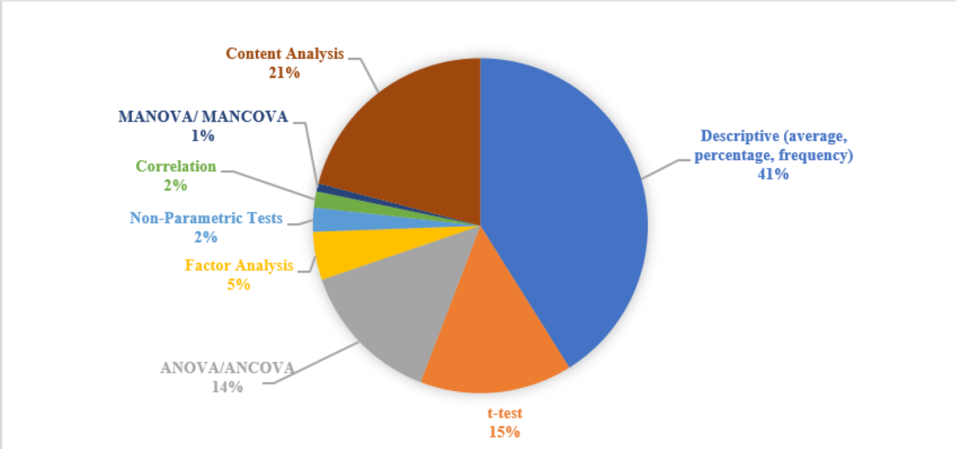

Fig. 10. Findings of data analysis methods used in programming education.

3.3. Findings of the Results of the Studies in Programming Education

The results of the studies in programming education were analyzed in terms of cognitive, affective processes, and the learning environment, and the findings are shown in Figure 10, Figure 11, and Figure 12. In these tables, positive and negative results are also included.

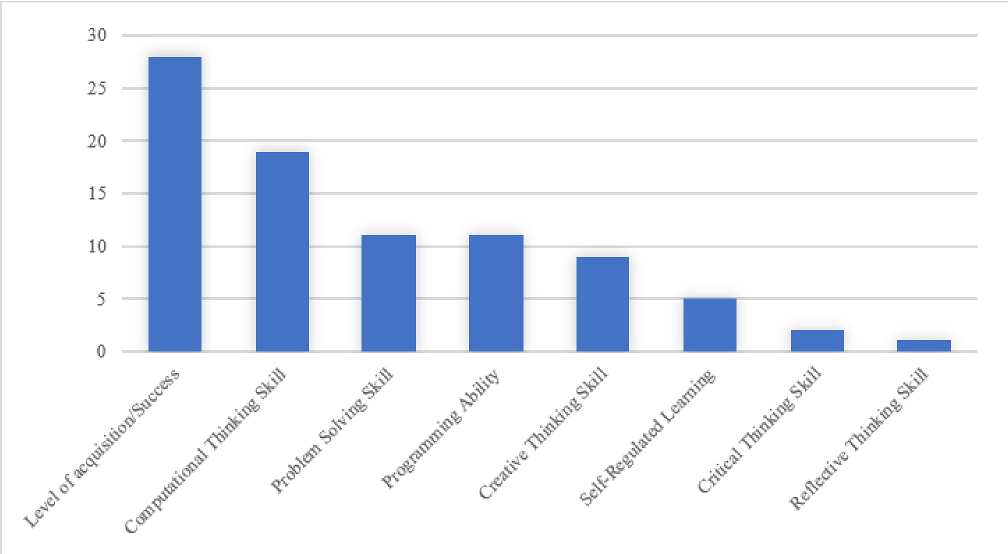

Fig. 11. Findings of the cognitive processes in the studies.

Figure 11 shows that programming education positively affects the learners' learning level and academic achievement the most (28%). Furthermore, it is seen that programming education also positively affects learners' thinking skills such as computational thinking (19%), problem-solving (11%), creative thinking (9%), critical thinking (2%), and reflective thinking (1%). However, there is no significant difference between programming education and academic success (7%) and computational thinking skills (2%) in some studies. In addition, some studies show that programming education does not significantly differ between computational thinking skills (2%) and individual learning skills (1%).

Figure 12 shows the results of the studies examined in terms of affective processes.

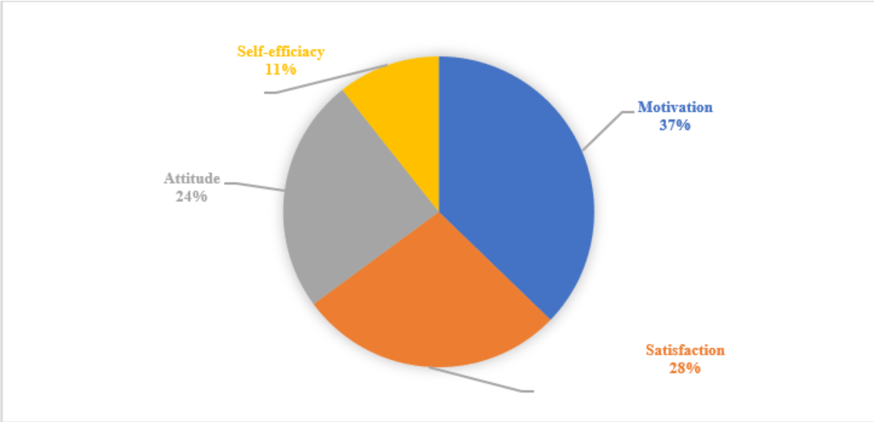

Fig. 12. Findings of the impact of affective processes in the programming education.

The studies show that affective processes include the dimensions such as motivation, satisfaction, attitude, and self-efficacy. When the positive effects are examined, it is clear that programming education has a positive contribution to learners' motivation (35%), satisfaction level (26%), attitude (23%), and self-efficacy levels (7%). However, there is no significant difference in attitudes (2%) and motivation (2%).

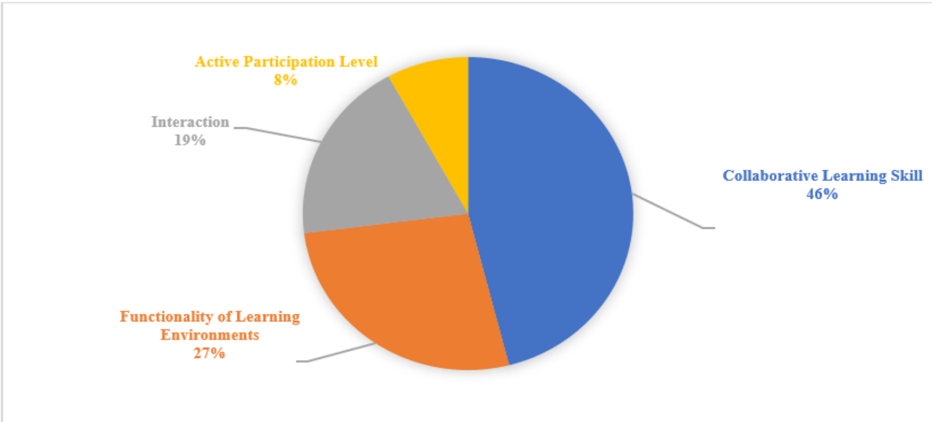

Fig. 13. Findings of the impact of the learning environment in programming education.

Based on the data, the findings that show the effects of the learning environment and the programming education on the learning environment or system are shown in Figure 13.

Figure 13 shows that programming education positively impacts learners' collaborative learning skills (46%). However, programming learning environments are also found to impact usability and practicality (13%) positively. It was also found that programming education positively impacts learners' interaction (19%) and active participation levels (8%) in the learning environment.

3.4. Findings of the Limitations of the Studies on Programming Education.

The limitations of studies on programming education were examined, which will contribute to a strong interpretation of the findings within the scope of the research. In addition, reviewing the limitations of the studies is important for the reproducibility of similar research (Ahadi, Hellas, Ihantola, Korhonen & Petersen, 2016). Figure 14 shows the information on the limitations of the reviewed studies.

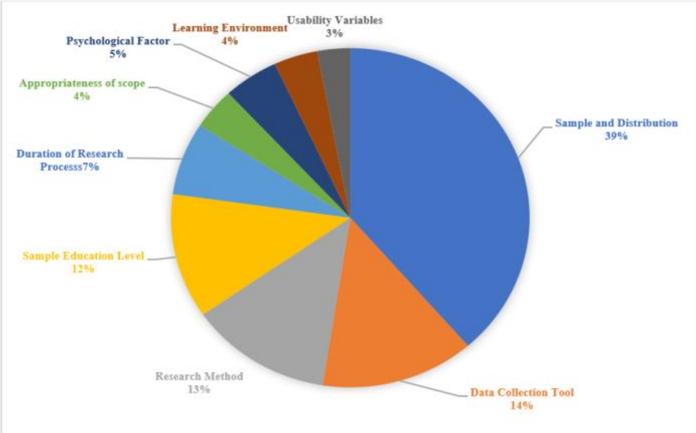

Fig. 14. Findings of the limitations of the studies in programming education.

Regarding the limitations of the studies on programming education, it was found that sample size sample and distribution (39%) is the most common limitation stated by the researchers. The least common limitation is the appropriateness of scope (4%). In addition, the psychological factor, environmental, and usability variables are close in number in terms of the working environment and process of the studies examined. In addition, in 98 of the studies, no information was given by the researcher about the limitations.

3.5. Future Research Implications in the Studies in Programming Education

In the studies conducted on programming education, examining the future implications is important to interpret the findings from different perspectives and to reach new ideas for future research. In addition, it is expected that the data obtained within the scope of the implications will shed light on the studies that will focus on a similar topic. The results are presented in Figure 15.

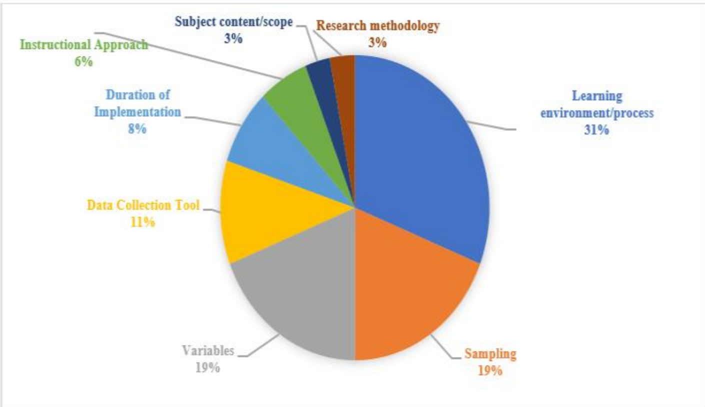

Fig. 15. Findings of the recommendations presented in the research on programming education.

When the recommendations proposed in the studies on programming education were analyzed, it was found that the most important recommendations were made about the learning environment and learning process (31%). The content (3%) and research methodology (3%) are the least common. Figure 13 shows that the numerical data of the recommendations stated in the studies are very few. This can be explained by the recommendations not displayed in the studies whose data were examined. For example, 99 of the reviewed studies did not include any suggestions for future research.

4. Discussion and Conclusion

This paper focused on examining the methodological dimensions of programming education articles published in educational sciences journals indexed in SSCI by exploring their general trends. To do this, 162 articles published between January 2012 and February 2020 in 30 international journals indexed in SSCI were analyzed with a systematic review method using the "Educational Technology Publication Classification Form" as a data collection tool.

4.1. Discussion of findings related to demographic characteristics of research conducted on programming education.

The studies on programming education regarding demographic characteristics revealed that the number of studies has increased to over 20 since 2015. When we look at similar studies in the literature, it is seen that there has been an increase in the number of scientific studies on programming education, especially after 2010. Furthermore, when the research results are compared with the other research results, it is seen that the results are consistent (Alaqsam, Ghabban, Ameerbakhsh, Alfadli &Fayez, 2021; Apiola, Saqr, López-Pernas & Tedre, 2022; Becker & Quille, 2019; Decker & McGill, 2017; Lukkarinen, Malmi, & Haaranen, 2021; Luxton-Reilly, A., Albluwi, I., Becker, Giannakos, Kanika, Chakraverty & Chakraborty, 2020; Kumar, Ott, & Szabo, 2018; Omer, Farooq & Abid, 2021; Papamitsiou, Giannakos, Simon, & Luxton-Reilly, 2020; Scaico, Scaico & Queiroz, 2018; Sobral, 2021; Sun, Guo & Zhou, 2022; Szabo et al., 2019). The increase can be due to the importance of this issue worldwide, mainly among businesspeople who have a career in technology (Garo, Kume & Basho, 2015). In addition, this increase since 2010 may be related to the integration of computer science and programming education as a course in the curricula of countries such as Bulgaria, Czech Republic, Denmark, Estonia, Greece, Ireland, Italy, Poland, Portugal, and England (Balanskat & Engelhardt, 2015; Manches & Plowman, 2017).

It is seen that the studies are primarily conducted in the USA, Turkiye, Taiwan, Spain, and South Korea. Furthermore, it is seen that the results of the research are similar to the results of other studies (Apiola et al., 2022; Decker et al., 2017; Scaico et al., 2018; Szabo et al., 2019). In addition, in a different literature review, results show that most studies on programming education are carried out in Malaysia, Australia, England, Portugal, and Brazil (Maia, Serey & Figueiredo, 2017; Sobral, 2021).

4.2. Discussion of findings related to methodological trends of the research conducted on programming education.

Quantitative methodology was most frequently used in programming education research in the reviewed articles, followed by mixed and qualitative methods. This finding is consistent with that of Hao et al. (2019), Lukkarinen (2021), Luxton- Reilly et al. (2018), Scaico et al. (2018), Shahid, Wajid, Haq, Saleem & Shujja (2019), Tunga & Tokel (2018), who reviewed methodological trends of programming education research.

When the sampling methods used in studies were examined, it was found that the sampling method which was used most was purposive sampling (67%), convenience sampling (30%). However, it was found that very few of the samples were selected randomly (3%) in the studies. Other findings in the literature also support these findings (Sanders, Sheard, Becker, Eckerdal, & Hamouda, 2019).

Regarding the type of education level, undergraduate education and K-8 level is the most common education level used in computer science and programming studies. Various studies in the literature show similar results (Berssanette & de Francisco, 2021; Hao et al., 2019; Maia, Serey& Figueiredo, 2017; Santos et al., 2020; Sun et al., 2022). Another finding is that 9-12 grades are not preferred as a sample group within the K-12 level. The number of studies conducted with high school students is low may be related to the fact that computer science courses take place among the elective courses across many countries. Another less preferred sample group is when teachers and faculty members are included. This may be because reaching the teachers and faculty members to carry out research is not easy (Guzdial, 2016; Saini & Chomal, 2017).

The distribution of sample sizes preferred in reviewed studies mainly consists of 31-100 and followed by 101-300 participants. The results are in line with several research studies (Grover, Basu, Bienkowski, Eagle, Diana & Stamper, 2017; Sol, Santos, Reis & Pereira, 2021). However, the number of studies conducted with 1-10, 11-30, and 301-1000 participants is relatively low. Maybe these numbers can be explained because the researcher could not reach the target audience to collect data.

Among the data collection tools used in the studies, it was found that the tools used were compatible with the research method. The most common quantitative data collection tool in the studies reviewed is the questionnaire and achievement tests. Qualitative data collection tools such as interviews, video and audio recordings, documents, and alternative data collection tools (performance tests, diagnostic tests, concept maps, portfolio, rubrics) and attitude, perception, personality, or ability tests were also rarely used the studies. The results of data collection tools are consistent with several research in the literature (Scaico et al., 2018; Shahid et al., 2019; Sanders et al., 2019).

Descriptive statistics, which is quantitative data analysis method, was mostly used in the publications examined within the scope of the study: frequency analysis, percentage, and average are the most preferred types of descriptive statistics. Furthermore, T-test and ANOVA/ANCOVA were the most used techniques in predictive statistics. On the other hand, content analysis was mostly used in qualitative research studies. Previous studies show similar results (Hawlitschek, Berndt & Schulz, 2022; Sanders et.al., 2019).

Learning success, computational thinking skill, and programming ability, motivation, performance, student view, collaboration were most examined dependent variables in reviewed papers. These results match with various studies in literature (Agbo, Oyelere, Suhonen & Adewumi, 2019; Anindyaputri, Yuana & Hatta, 2020; Bati, 2021; Çam & Kıyıcı, 2022; Fagerlund, Häkkinen, Vesisenaho & Viiri, 2021; Grotta & Proda, 2019; Grover & Pea, 2013; Kalelioğlu, Gülbahar & Kukul, 2016; Lockwood & Mooney, 2018; Saqr, Ng, Oyelere & Tedre, 2021; Scaico et.al., 2018; Sol, Santos, Reis & Pereira, 2021; Shahid et.al., 2019; Tikva & Tambouris, 2021; Vihavainen, Airaksinen & Watson, 2014). However, creative thinking, problem-solving, and critical thinking skills are rarely examined as dependent variables in reviewed. This finding is not parallel with many studies associated with the above cognitive abilities (Korkmaz, 2018; Popat & Starkey, 2019).

4.3. Discussions related to the results of the studies.

The findings of reviewed studies revealed that programming education has various positive contributions to learners in terms of cognitive processes. In the literature, programming education is found to have positive contributions to the development of learning and achievement levels of the learners (Hughes-Roberts, Brown, Standen, Desideri, Negrini, Rouame & Hasson, 2019), computational thinking skills (Chalmers, 2018; Ioannou & Makridou, 2016; Gretter & Yadav, 2016;), programming skills (Claypool, 2013; Liu, Zhi, Hicks & Barnes, 2017), problem-solving skills (Çiftci & Bildiren, 2020), creative thinking skills (Peng & Wang, 2019), individual learning skills, critical thinking skills (Yang, Yang & Hwang, 2014) and reflective thinking skills (Durak, Yılmaz & Yılmaz, 2019).

Furthermore, it has been seen that programming education has various positive contributions to learners in affective processes. In the literature, it is stated that programming education depends on learners' motivation levels (Law, Lee & Yu, 2010; Nikula, Gotel & Kasurinen, 2011; Papastergiou, 2009), satisfaction levels (Bishop-Clark, Courte & Howard, 2006), self-efficacy levels and it has a positive contribution to their perceptions (Cheng, 2019; Mason & Cooper, 2013) and attitudes (Chen, Haduong, Brennan, Sonnert & Sadler, 2019).

In the studies examined within the scope of this study, programming education has a positive impact on the development of collaborative learning skills of the learners (Crellin, Williams, Chandler & Collinson, 2009; Da Silva Estácio & Prikladnicki, 2015; Othman & Zain, 2015; Yu & Roque, 2019; Lui, Kafai, Litts, Walker & Widman, 2020), the levels of interaction in the learning environment (Kavitha & Ahmed, 2013) and active participation levels (Cukierman, 2015). In addition, studies suggest that learners find programming learning environments practical (Bati, Gelderblom & Biljon, 2014; Becker & Quille, 2019).

5. Limitations

A systematic review conducted in this study is limited only to the studies published in Educational Sciences journals indexed in SSCI between January 2012 and February 2020 in English. Due to their high impact rates, SSCI journals publish quality studies. Since publishing articles in these journals take around 1-2 years, it should not be assumed that all the studies examined are up to date. In addition, it should be noted that the research results only reflect the studies in the field of educational sciences. However, because there has been an increase in the number of interdisciplinary studies on programming education, such studies may not have been published in only educational sciences journals. Since programming education is among the current research topics, new research in this field is also found in conferences and journals with other indexes. Therefore, the research results only reflect the results of the studies in the SSCI indexed journals.

6. Implications for Future Research

In line with the findings of this paper, the following suggestions can be made for future research:

- To comprehensively examine the research results, articles published on different dates and in non-indexed journals may also be analyzed in future studies.
- Since the studies are mainly carried out with quantitative methods, the number of theoretical studies on how to use qualitative and mixed methods and how these methods will be carried out in research can be increased.
- It can be suggested select the sample randomly by paying attention to the sample selection methods. In addition, future research can be conducted using different sample levels.
- To bring different perspectives to research, it can be suggested that more research should be conducted focusing on instructional technologies used in programming education.

Acknowledgements:

This research is the summary of the masters dissertation titled "An analysis of research on programming education" written by Atajan Rovshenov and supervised by Fırat Sarsar.

References

Alaqsam, A., Ghabban, F., Ameerbakhsh, O., Alfadli, I., & Fayez, A. (2021). Current Trends in OnlineProgramming Languages Learning Tools: A Systematic Literature Review. *Journal of Software Engineering and Applications*, *14*(7), 277-297. 10.4236/jsea.2021.147017

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Adewumi, S. (2019, November). A systematic review of computational thinking approach for programming education in higher education institutions. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1-10). https://doi.org/10.1145/3364510.3364521

Ahadi, A., Hellas, A., Ihantola, P., Korhonen, A., & Petersen, A. (2016, November). Replication in computing education research: researcher attitudes and experiences. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 2-11). https://doi.org/10.1145/2999541.2999554

Anindyaputri, N. A., Yuana, R. A., & Hatta, P. (2020). Enhancing Students' Ability in Learning Process of Programming Language using Adaptive Learning Systems: A Literature Review. *Open Engineering*, *10*(1), 820-829.

Apiola, M., Saqr, M., López-Pernas, S., & Tedre, M. (2022). Computing Education Research Compiled: Keyword Trends, Building Blocks, Creators, and Dissemination. *IEEE Access*. 10.1109/ACCESS.2022.3157609

Balanskat, A., & Engelhardt, K. (2015). Computer programming and coding priorities, school curricula, and initiatives across Europe. *European schoolnet*. http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03

Bati, T. B., Gelderblom, H., & Van Biljon, J. (2014). A blended learning approach for teaching computer programming: design for large classes in Sub-Saharan Africa. *Computer Science Education*, *24*(1), 71-99. https://doi.org/10.1080/08993408.2014.897850

Bati, K. (2021). A systematic literature review regarding computational thinking and programming in early childhood education. *Education and Information Technologies*, 1-24. https://doi.org/10.1007/s10639-021-10700-2

Battal, A., Afacan Adanır, G., & Gülbahar, Y. (2021). Computer Science Unplugged: A Systematic Literature Review. *Journal of Educational Technology Systems*, 50(1), 1-24. https://doi.org/10.1177/00472395211018801

Becker, B. A., & Quille, K. (2019, February). 50 years of cs1 at sigcse: A review of the evolution of introductory programming education research. In *Proceedings of the 50th acm technical symposium on computer science education* (pp. 338-344). https://doi.org/10.1145/3287324.3287432

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, *58*(3), 978-988. https://doi.org/10.1016/j.compedu.2011.10.006

Berssanette, J. H., & de Francisco, A. C. (2021). Active learning in the context of the teaching/learning of computer programming: A systematic review. *Journal of Information Technology Education. Research*, *20*, 201. https://doi.org/10.28945/4767

Bishop-Clark, C., Courte, J., & Howard, E. V. (2006). Programming in pairs with Alice to improve confidence, enjoyment, and achievement. *Journal of educational computing research*, *34*(2), 213-228. https://doi.org/10.2190/CFKF-UGGC-JG1Q-7T40

Borrego, M., Foster, M. J., & Froyd, J. E. (2014). Systematic literature reviews in engineering education and other developing interdisciplinary fields. *Journal of Engineering Education*, *103*(1), 45-76. https://doi.org/10.1002/jee.20038

Butterfield, A., Ngondi, G. E., & Kerr, A. (Eds.). (2016). *A dictionary of computer science*. Oxford University Press. https:/doi.org/10.1093/acref/9780199688975.001.0001

Caeli, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. *TechTrends*, *64*(1), 29-36. https://doi.org/10.1007/s11528-019-00410-5

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, *17*, 93-100. https://doi.org/10.1016/j.ijcci.2018.06.005

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, 12(2), 1-14. https://doi.org/10.30935/cedtech/8247

Chen, C., Haduong, P., Brennan, K., Sonnert, G., & Sadler, P. (2019). The effects of first programming language on college students' computing attitude and achievement: a comparison of graphical and textual languages. *Computer Science Education*, *29*(1), 23-48. https://doi.org/10.1080/08993408.2018.1547564

Cheng, G. (2019). Exploring factors influencing the acceptance of visual programming environment among boys and girls in primary schools. *Computers in Human Behavior*, *92*, 361-372. https://doi.org/10.1016/j.chb.2018.11.043

Claypool, M. (2013). Dragonfly: strengthening programming skills by building a game engine from Scratch. *Computer Science Education*, *23*(2), 112-137. https://doi.org/10.1080/08993408.2013.781840

Crellin, J., Duke-Williams, E., Chandler, J., & Collinson, T. (2009). Virtual worlds in computing education. *Computer Science Education*, *19*(4), 315-334. https://doi.org/10.1080/08993400903384950

Cukierman, D. (2015, June 4-8). Predicting success in university first year computing science courses: The role of student participation in reflective learning activities and in i-clicker activities. In *Proceedings of the 2015 acm conference on innovation and technology in computer science education* (pp. 248-253). https://doi.org/10.1145/2729094.2742623

Çam, E. & Kıyıcı, M. (2022). The impact of robotics assisted programming education on academic success, problem solving skills and motivation. *Journal of Educational Technology and Online Learning*, *5*(1), 47-65. https://doi.org/10.31681/jetol.1028825

Çiftci, S., & Bildiren, A. (2020). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. *Computer science education*, *30*(1), 3-21. https://doi.org/10.1080/08993408.2019.1696169

Dağhan, G., Nuhoğlu Kibar, P., Menzi Çetin, N., Telli, E., & Akkoyunlu, B. (2017). 21st century learners' and teachers' charactersistics from ICT preservice teachers' perspectives. *Educational Technology Theory and Practise*, *7*(2), 215-235. https://doi.org/10.17943/etku.305062

Da Silva Estácio, B. J., & Prikladnicki, R. (2015). Distributed pair programming: A systematic literature review. *Information and Software Technology*, *63*, 1-10. https://doi.org/10.1016/j.infsof.2015.02.011

Decker, A., & McGill, M. M. (2017, March). Pre-college computing outreach research: Towards improving the practice. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 153-158). https://doi.org/10.1145/3017680.3017744

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

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, *29*(1), 12-28. https://doi.org/10.1002/cae.22255

Garo, E., Kume, V., & Basho, S. (2015). " Programming" an Entrepreneur. *Academic Journal of Interdisciplinary Studies*, *4*(1 S1), 45-45. https://dx.doi.org/10.5901/mjss.2015.v4n1s1p45

Goktas, Y., Kucuk, S., Aydemir, M., Telli, E., Arpacik, O., Yildirim, G., & Reisoglu, I. (2012). Educational technology research trends in Turkey: A content analysis of the 2000-2009 decade. *Educational Sciences: Theory and Practice*, *12*(1), 191-199.

Gomes, A., & Mendes, A. J. (2007, September 3-7). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE* (Vol. 7). https://www.researchgate.net/publication/228328491_Learning_to_program_-_difficulties_and_solutions

Grotta, A., & Prado, E. P. V. (2019, July). Benefits of The Project-Based Learning to Cope with Computer Programming Education: A Systematic Literature Review. In *PBL2019 Immersive Virtual International Conference, 2019b. Proceedings. 1-10.*

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, *42*(1), 38-43.

Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education (TOCE)*, *17*(3), 1-25. https://doi.org/10.1145/3105910

Gretter, S., & Yadav, A. (2016). Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills. *TechTrends*, *60*(5), 510-516. https://doi.org/10.1007/s11528-016-0098-4

Gurer, D., M., Cetin, I., & Top, E. (2019). Factors affecting students' attitudes toward computer programming. *Informatics in Education*, *18*(2), 281-296. https://doi.org/10.15388/infedu.2019.13

Guzdial, M. (2016). Bridging Computer Science to US Schools. Communications of the *ACM*, *59*(5), 24-25. https://doi.org/10.1145/2898963

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

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, *37*(2), 83-137. https://doi.org/10.1145/1089733.1089734

Kalelioglu, F., & Sentance, S. (2020). Teaching with physical computing in school: the case of the micro: bit. *Education and Information Technologies*, *25*(4), 2577-2603. https://doi.org/10.1007/s10639-019-10080-8

Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, *4*(3), 583-596.

Hao, Q., Smith IV, D. H., Iriumi, N., Tsikerdekis, M., & Ko, A. J. (2019). A systematic investigation of replications in computing education research. *ACM Transactions on Computing Education (TOCE)*, *19*(4), 1-18. https://doi.org/10.1145/3345328

Hawlitschek, A., Berndt, S., & Schulz, S. (2022). Empirical research on pair programming in higher education: a literature review. *Computer Science Education*, 1-29. https://doi.org/10.1080/08993408.2022.2039504

Hughes-Roberts, T., Brown, D., Standen, P., Desideri, L., Negrini, M., Rouame, A., & Hasson, C. (2019). Examining engagement and achievement in learners with individual needs through robotic-based teaching sessions. *British journal of educational technology*, *50*(5), 2736-2750. https://doi.org/10.1111/bjet.12722

Hundt, C., Schlarb, M., & Schmidt, B. (2017). SAUCE: A web application for interactive teaching and learning of parallel programming. *Journal of Parallel and Distributed Computing*, *105*, 163-173. https://doi.org/10.1016/j.jpdc.2016.12.028

Ioannou, A., & Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, *23*(6), 2531-2544. https://doi.org/10.1007/s10639-018-9729-z

Kanika, Chakraverty, S., & Chakraborty, P. (2020). Tools and techniques for teaching computer programming: A review. *Journal of Educational Technology Systems*, *49*(2), 170-198. https://doi.org/10.1177/0047239520926971

Kavitha, R. K., & Ahmed, M. I. (2013). Knowledge sharing through pair programming in learning environments: An empirical study. *Education and Information Technologies*, *20*(2), 319-333. https://doi.org/10.1007/s10639-013-9285-5

Krippendorff, K. (2004). Content analysis: an introduction to its methodology (2nd Ed.). Thousand Oaks, CA: Sage Publications.

Korkmaz, Ö. (2018). The effect of scratch-and Lego mindstorms EV3-Based programming activities on academic achievement, problem-solving skills and logical-mathematical thinking skills of students. *MOJES: Malaysian Online Journal of Educational Sciences*, *4*(3), 73-88.

Law, K. M., Lee, V. C., & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers & Education*, *55*(1), 218-228. https://doi.org/10.1016/j.compedu.2010.01.007

Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, *58*(4), 431-440. https://doi.org/10.1016/j.bushor.2015.03.008

Lindberg, R. S., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology*, *50*(4), 1979-1995. https://doi.org/10.1111/bjet.12685

Liu, Z., Zhi, R., Hicks, A., & Barnes, T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education*, *27*(1), 1-29. https://doi.org/10.1080/08993408.2017.1308651

Lockwood, J., & Mooney, A. (2017). Computational Thinking in Education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools.*(2) 1, 1-20

Lui, D., Kafai, Y., Litts, B., Walker, J., & Widman, S. (2020). Pair physical computing: high school students' practices and perceptions of collaborative coding and crafting with electronic textiles. *Computer Science Education*, *30*(1), 72-101.https://doi.org/10.1080/08993408.2019.1682378

Lukkarinen, A., Malmi, L., & Haaranen, L. (2021). Event-driven Programming in Programming Education: A Mapping Review. *ACM Transactions on Computing Education (TOCE)*, *21*(1), 1-31. https://doi.org/10.1145/3423956

Luxton-Reilly, A., Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L. & Szabo, C. (2018, July). Introductory programming: a systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 55-106). https://doi.org/10.1145/3293881.3295779

Maia, M. C. O., Serey, D., & Figueiredo, J. (2017, October). Learning styles in programming education: A systematic mapping study. In *2017 IEEE Frontiers in Education Conference (FIE)* (pp. 1-7). IEEE. 10.1109/FIE.2017.8190465

Mason, R., & Cooper, G. (2013). Mindstorms robots and the application of cognitive load theory in introductory programming. *Computer Science Education*, *23*(4), 296-314. https://doi.org/10.1080/08993408.2013.847152

McMillan, J.H., Schumacher, S. (2001). Research in education: a conceptual introduction. (5th Ed.). New York: Addison Wesley Longman Inc. 660.

Nikula, U., Gotel, O., & Kasurinen, J. (2011). A motivation guided holistic rehabilitation of the first programming course. *ACM Transactions on Computing Education (TOCE)*, *11*(4), 1-38. https://doi.org/10.1145/2048931.2048935

Noone, M., & Mooney, A. (2018). Visual and textual programming languages: a systematic review of the literature. *Journal of Computers in Education*, *5*(2), 149-174. https://doi.org/10.1007/s40692-018-0101-5

Omer, U., Farooq, M. S., & Abid, A. (2021). Introductory programming course: review and future implications. *PeerJ Computer Science*, *7*, 1-33. https://doi.org/10.7717/peerj-cs.647

Othman, M., &, Zain, N. (2015). Online collaboration for programming: Assessing students' cognitive abilities. *Turkish Online Journal of Distance Education*, *16*(4), 84-97. https://doi.org/10.17718/tojde.88618

Papamitsiou, Z., Giannakos, M., Simon, & Luxton-Reilly, A. (2020, August). Computing education research landscape through an analysis of keywords. In *Proceedings of the 2020 ACM Conference on International Computing Education Research* (pp. 102-112). https://doi.org/10.1145/3372782.3406276

Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & education*, *52*(1), 1-12. https://doi.org/10.1016/j.compedu.2008.06.004

Peng, Y. C., & Wang, T. I. (2019, December 2-5). The Investigation on Creative Thinking into Projected-Base Programming Course for College Students. In *International Conference on Innovative Technologies and Learning* (pp. 713-725). Springer, Cham. https://10.1007/978-3-030-35343-8

Petticrew, M., & Roberts, H. (2008). *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons.

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, *128*, 365-376. https://doi.org/10.1016/j.compedu.2018.10.005

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, *13*(2), 137-172. https://doi.org/10.1076/csed.13.2.137.14200

Salleh, S. M., Shukur, Z., & Judi, H. M. (2013). Analysis of research in programming teaching tools: An initial review. *Procedia-Social and Behavioral Sciences*, *103*, 127-135. https://doi.org/10.1016/j.sbspro.2013.10.317

Sanders, K., Sheard, J., Becker, B. A., Eckerdal, A., & Hamouda, S. (2019, July). Inferential statistics in computing education research: A methodological review. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 177-185). https://doi.org/10.1145/3291279.3339408

Santos, S. C., Tedesco, P. A., Borba, M., & Brito, M. (2020). Innovative approaches in teaching programming: A systematic literature review. In *Proceedings of the 12th International Conference on Computer Supported Education* (Vol. 1, pp. 205-214).

Saini, J. R., & Chomal, V. S. (2017). Use of Analytical Hierarchy Process for Selection of Elective Subjects by Pre-Final Year Students of Computer Science. *International Journal on Recent and Innovation Trends in Computing and Communication*, *5*(5), 1196-1202.

Saqr, M., Ng, K., Oyelere, S. S., & Tedre, M. (2021). People, ideas, milestones: a scientometric study of computational thinking. *ACM Transactions on Computing Education (TOCE)*, *21*(3), 1-17. https://doi.org/10.1145/3445984

Scaico, P. D., Scaico, A., & de Queiroz, R. J. B. (2018, October). An Initial Analysis of the Research on Interest and Introductory Programming: A Systematic Review of this Literature. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE. 10.1109/FIE.2018.8659254

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*, *111*(5), 764. https://doi.org/10.1037/edu0000314

Shahid, M., Wajid, A., Haq, K. U., Saleem, I., & Shujja, A. H. (2019, November). A review of gamification for learning programming fundamental. In *2019 International Conference on Innovative Computing (ICIC)* (pp. 1-8). IEEE. 10.1109/ICIC48496.2019.8966685

Sobral, S. R. (2021). Teaching and Learning to Program: Umbrella Review of Introductory Programming in Higher Education. *Mathematics*, *9*(15), 1737. https://doi.org/10.3390/math9151737

Sol, R., Santos, E. A., Reis, M. C., & Pereira, L. (2021). Computer Supported Collaborative Learning for Programming: A Systematic Review. *CSEDU (2)*, 184-191.

Sun, L., Guo, Z., & Zhou, D. (2022). Developing K-12 students' programming ability: A systematic literature review. *Education and Information Technologies*, 1-39. https://doi.org/10.1007/s10639-022-10891-2

Szabo, C., Sheard, J., Luxton-Reilly, A., Becker, B. A., & Ott, L. (2019, November). Fifteen years of introductory programming in schools: a global overview of K-12 initiatives. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1-9). https://doi.org/10.1145/3364510.3364513

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers & Education*, *162*, 1-23. https://doi.org/10.1016/j.compedu.2020.104083

Tunga, Y., & Tokel, S. T. (2018). The use of pair programming in education: A systematic review. In 2018 Educcon Education 4.0 Conference. (pp. 19-29).

Tuparov, G., Tuparova, D., & Jordanov, V. (2014). Teaching sorting and searching algorithms through simulation-based learning objects in an introductory programming course. Procedia-Social and Behavioral Sciences, 116, 2962-2966. https://doi.org/10.1016/j.sbspro.2014.01.688

Veena, A., & Gowrishankar, S. (2018). *Introduction to Python Programming*. CRC Press.

Vihavainen, A., Airaksinen, J., & Watson, C. (2014, July). A systematic review of approaches for teaching introductory programming and their influence on success. In *Proceedings of the tenth annual conference on International computing education research* (pp. 19-26). https://doi.org/10.1145/2632320.2632349

Wallen, N. E., & Fraenkel, J. R. (2013). *Educational research: A guide to the process*. Routledge.

Webb, M. E., Bell, T., Davis, N., Katz, Y. J., Fluck, A., Sysło, M. M., & Brodnik, A. (2018). Tensions in specifying computing curricula for K-12: Towards a principled approach for objectives. *IT-Information Technology*, *60*(2), 59-68. https://doi.org/10.1515/itit-2017-0017

Wohl, B. S., Beck, S., & Blair, L. (2017). The Future of the Computing Curriculum: How the Computing Curriculum Instills Values and Subjectivity in Young People. *International Journal of Computer Science Education in Schools*, *1*(1), 1-9.

Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, *18*(1), 1-24. https://doi.org/10.1145/3077618

Qian, Y., Hambrusch, S., Yadav, A., Gretter, S., & Li, Y. (2020). Teachers' perceptions of student misconceptions in introductory programming. *Journal of Educational Computing Research*, *58*(2), 364-397. https://doi.org/10.1177/0735633119845413

Yang, T. C., Yang, S. J., & Hwang, G. J. (2014, July 7-10). Development of an interactive test system for students' improving learning outcomes in a computer programming course. In *2014 IEEE 14th International Conference on Advanced Learning Technologies* (pp. 637-639). IEEE. https://doi.org/10.1109/ICALT.2014.186

Yesharim, M. F., & Ben-Ari, M. (2018). Teaching computer science concepts through robotics to elementary school children. *International Journal of Computer Science Education in Schools*, *2*(3), 1-22. https://doi.org/10.21585/ijcses.v2i3.30

Yu, J., & Roque, R. (2019). A review of computational toys and kits for young children. *International Journal of Child-Computer Interaction*, *21*, 17-36. https://doi.org/10.1016/j.ijcci.2019.04.001

Appendix

Appendix A. Selected Studies

Akpinar, Y., & Aslan, Ü. (2015). Supporting children's learning of probability through video game programming. *Journal of Educational Computing Research*, *53*(2), 228-259. https://doi.org/10.1177/0735633115598492

Allsop, Y. (2016). A reflective study into children's cognition when making computer games. *British Journal of Educational Technology*, *47*(4), 665-679. https://doi.org/10.1111/bjet.12251

Altintas, T., Gunes, A., & Sayan, H. (2016). A peer-assisted learning experience in computer programming language learning and developing computer programming skills. *Innovations in Education and Teaching International*, *53*(3), 329-337. https://doi.org/10.1080/14703297.2014.993418

Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education*, *148*, 1-16. https://doi.org/10.1016/j.compedu.2020.103807

Benotti, L., Martnez, M. C., & Schapachnik, F. (2017). A tool for introducing computer science with automatic formative assessment. *IEEE Transactions on Learning Technologies*, *11*(2), 179-192. https://doi.org/10.1109/TLT.2017.2682084

Benton, L., Kalas, I., Saunders, P., Hoyles, C., & Noss, R. (2018). Beyond jam sandwiches and cups of tea: An exploration of primary pupils' algorithm-evaluation strategies. *Journal of Computer Assisted Learning*, *34*(5), 590-601. https://doi.org/10.1111/jcal.12266

Berland, M., Martin, T., Benton, T., Petrick Smith, C., & Davis, D. (2013). Using learning analytics to understand the learning pathways of novice programmers. *Journal of the Learning Sciences*, *22*(4), 564-599. https://doi.org/10.1080/10508406.2013.836655

Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology*, *24*(5), 628-647. https://doi.org/10.1007/s10956-015-9552-x

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, *138*, 130-145. https://doi.org/10.1016/j.compedu.2019.04.013

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145-157. https://doi.org/10.1016/j.compedu.2013.10.020

Boldbaatar, N., & Şendurur, E. (2019). Developing Educational 3D Games With StarLogo: The Role of Backwards Fading in the Transfer of Programming Experience. *Journal of Educational Computing Research*, *57*(6), 1468-1494. https://doi.org/10.1177/0735633118806747

Burleson, W. S., Harlow, D. B., Nilsen, K. J., Perlin, K., Freed, N., Jensen, C. N., & Muldner, K. (2017). Active learning environments with robotic tangibles: Children's physical and virtual spatial programming experiences. *IEEE Transactions on Learning Technologies*, *11(1)*, 96-106. https://doi.org/10.1109/TLT.2017.2724031

Bustillo, J., & Garaizar, P. (2016). Using Scratch to foster creativity behind bars: Two positive experiences in jail. *Thinking Skills and Creativity*, *19*, 60-72. https://doi.org/10.1016/j.tsc.2015.08.003

Cabada, R. Z., Estrada, M. L. B., Hernández, F. G., Bustillos, R. O., & Reyes-García, C. A. (2018). An affective and Web 3.0-based learning environment for a programming language. *Telematics and Informatics*, *35*(3), 611-628. https://doi.org/10.1016/j.tele.2017.03.005

Caeli, E. N., & Bundsgaard, J. (2020). Computational thinking in compulsory education: A survey study on initiatives and conceptions. *Educational Technology Research and Development*, *68*(1), 551-573. https://doi.org/10.1007/s11423-019-09694-z

Carbonaro, A. (2019). Good practices to influence engagement and learning outcomes on a traditional introductory programming course. *Interactive Learning Environments*, *27*(7), 919-926. https://doi.org/10.1080/10494820.2018.1504307

Castro, E., Cecchi, F., Valente, M., Buselli, E., Salvini, P., & Dario, P. (2018). Can educational robotics introduce young children to robotics and how can we measure it?. *Journal of Computer Assisted Learning*, *34*(6), 970-977. https://doi.org/10.1111/jcal.12304

Città, G., Gentile, M., Allegra, M., Arrigo, M., Conti, D., Ottaviano, S., & Sciortino, M. (2019). The effects of mental rotation on computational thinking. *Computers & Education*, *141*, 1-11. https://doi.org/10.1016/j.compedu.2019.103613

Cetin, I. (2016). Preservice teachers' introduction to computing: exploring utilization of scratch. *Journal of Educational Computing Research*, *54*(7), 997-1021. https://doi.org/10.1177/0735633116642774

Cetin, I., Sendurur, E., & Sendurur, P. (2014). Assessing the impact of meta-cognitive training on students' understanding of introductory programming concepts. *Journal of Educational Computing Research*, *50*(4), 507-524. https://doi.org/10.2190/EC.50.4.d

Chang, C. K. (2014). Effects of using Alice and Scratch in an introductory programming course for corrective instruction. *Journal of Educational Computing Research*, *51*(2), 185-204. https://doi.org/10.2190/EC.51.2.c

Chao, P. Y. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education*, *95*, 202-215. https://doi.org/10.1016/j.compedu.2016.01.010

Charlton, P., & Avramides, K. (2016). Knowledge construction in computer science and engineering when learning through making. *IEEE Transactions on Learning Technologies*, *9*(4), 379-390. https://doi.org/10.1109/TLT.2016.2627567

Čisar, S. M., Čisar, P., & Pinter, R. (2016). Evaluation of knowledge in Object Oriented Programming course with computer adaptive tests. *Computers & education*, *92*, 142-160. https://doi.org/10.1016/j.compedu.2015.10.016

Corral, J. M. R., Balcells, A. C., Estévez, A. M., Moreno, G. J., & Ramos, M. J. F. (2014). A game-based approach to the teaching of object-oriented programming languages. *Computers & Education*, *73*, 83-92. https://doi.org/10.1016/j.compedu.2013.12.013

Crescenzi, P., Malizia, A., Verri, M. C., Díaz, P., & Aedo, I. (2012). Integrating Algorithm Visualization Video into a First-Year Algorithm and Data Structure Course. *J. Educ. Technol. Soc.*, *15*(2), 115-124.

Cummins, S., Beresford, A. R., & Rice, A. (2015). Investigating engagement with in-video quiz questions in a programming course. *IEEE Transactions on Learning Technologies*, *9(1)*, 57-66. https://doi.org/10.1109/TLT.2015.2444374

Çakır, N. A., Gass, A., Foster, A., & Lee, F. J. (2017). Development of a game-design workshop to promote young girls' interest towards computing through identity exploration. *Computers & Education*, *108*, 115-130. https://doi.org/10.1016/j.compedu.2017.02.002

Çakıroğlu, Ü. (2014). Analyzing the effect of learning styles and study habits of distance learners on learning performances: A case of an introductory programming course. *International Review of Research in Open and Distributed Learning*, *15*(4), 161-185. https://doi.org/10.19173/irrodl.v15i4.1840

Çakıroğlu, Ü., Kokoç, M., Kol, E., & Turan, E. (2016). Exploring teaching programming online through web conferencing system: The lens of activity theory. *Journal of Educational Technology & Society*, *19*(4), 126-139.

Daradoumis, T., Puig, J. M. M., Arguedas, M., & Liñan, L. C. (2019). Analyzing students' perceptions to improve the design of an automated assessment tool in online distributed programming. *Computers & Education*, *128*, 159-170. https://doi.org/10.1016/j.compedu.2018.09.021

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students?. *Journal of Research on Technology in Education*, *46*(3), 277-296. https://doi.org/10.1080/15391523.2014.888272

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, *150*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Dohn, N. B. (2020). Students’ interest in Scratch coding in lower secondary mathematics. *British Journal of Educational Technology*, *51*(1), 71-83. https://doi.org/10.1111/bjet.12759

dos Santos Lopes, M. S., Gomes, I. P., Trindade, R. M., da Silva, A. F., & Lima, A. C. D. C. (2016). Web environment for programming and control of a mobile robot in a remote laboratory. *IEEE Transactions on Learning Technologies, 10(4),* 526-531. https://doi.org/10.1109/TLT.2016.2627565

Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, *116*, 191-202. https://doi.org/10.1016/j.compedu.2017.09.004

Eranki, K. L., & Moudgalya, K. M. (2016). Comparing the Effectiveness of Self-Learning Java Workshops with Traditional Classrooms. *J. Educ. Technol. Soc.*, *19*(4), 59-74.

Erümit, K. A., Karal, H., Şahin, G., Aksoy, D. A., Aksoy, A., & Benzer, A. İ. (2018). Programlama öğretimi için bir model önerisi: Yedi adımda programlama. *Eğitim ve Bilim*, *44*(197). http://dx.doi.org/10.15390/EB.2018.7678

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. *Journal of Computer Assisted Learning*, *32*(6), 576-593. https://doi.org/10.1111/jcal.12155

Ferreira, D. J., da Silva, H. C., Melo, T. F., & Ambrósio, A. P. (2017). Investigation of continuous assessment of correctness in introductory programming. *Journal of Educational Technology & Society*, *20*(3), 182-194.

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://doi.org/10.1016/j.compedu.2012.11.016

Flannery, L. P., & Bers, M. U. (2013). Let's dance the "robot hokey-pokey!" children's programming approaches and achievement throughout early cognitive development. *Journal of research on technology in education*, *46*(1), 81-101. https://doi.org/10.1080/15391523.2013.10782614

Fields, D., Vasudevan, V., & Kafai, Y. B. (2015). The programmers' collective: fostering participatory culture by making music videos in a high school Scratch coding workshop. *Interactive Learning Environments*, *23*(5), 613-633. https://doi.org/10.1080/10494820.2015.1065892

Fwa, H. L. (2018). An architectural design and evaluation of an affective tutoring system for novice programmers. *International Journal of Educational Technology in Higher Education*, *15*(1), 1-19. https://doi.org/10.1186/s41239-018-0121-2

García-Valcárcel-Muñoz-Repiso, A., & Caballero-González, Y. A. (2019). Robotics to develop computational thinking in early Childhood Education. *Comunicar. Media Education Research Journal*, *27*(1), 1-14. https://doi.org/10.3916/C59-2019-06

Garneli, V., & Chorianopoulos, K. (2018). Programming video games and simulations in science education: exploring computational thinking through code analysis. *Interactive Learning Environments*, *26*(3), 386-401. https://doi.org/10.1080/10494820.2017.1337036

Hao, Q., Barnes, B., Wright, E., & Branch, R. M. (2017). The influence of achievement goals on online help seeking of computer science students. *British Journal of Educational Technology*, *48*(6), 1273-1283. https://doi.org/10.1111/bjet.12499

Hayes, J., & Stewart, I. (2016). Comparing the effects of derived relational training and computer coding on intellectual potential in school-age children. *British Journal of Educational Psychology*, *86*(3), 397-411. https://doi.org/10.1111/bjep.12114

Hershkovitz, A., Sitman, R., Israel-Fishelson, R., Eguíluz, A., Garaizar, P., & Guenaga, M. (2019). Creativity in the acquisition of computational thinking. *Interactive Learning Environments*, *27*(5-6), 628-644. https://doi.org/10.1080/10494820.2019.1610451

Hooshyar, D., Ahmad, R. B., Yousefi, M., Fathi, M., Horng, S. J., & Lim, H. (2016). Applying an online game-based formative assessment in a flowchart-based intelligent tutoring system for improving problem-solving skills. *Computers & Education*, *94*, 18-36. https://doi.org/10.1016/j.compedu.2015.10.013

Hooshyar, D., Binti Ahmad, R., Wang, M., Yousefi, M., Fathi, M., & Lim, H. (2018). Development and evaluation of a game-based bayesian intelligent tutoring system for teaching programming. *Journal of Educational Computing Research*, *56*(6), 775-801. https://doi.org/10.1177/0735633117731872

Hooshyar, D., Ahmad, R. B., Yousefi, M., Yusop, F. D., & Horng, S. J. (2015). A flowchart-based intelligent tutoring system for improving problem-solving skills of novice programmers. *Journal of Computer Assisted Learning*, *31*(4), 345-361. https://doi.org/10.1111/jcal.12099

Howard, A. M., Park, C. H., & Remy, S. (2011). Using haptic and auditory interaction tools to engage students with visual impairments in robot programming activities. *IEEE transactions on learning technologies*, *5*(1), 87-95. https://doi.org/10.1109/TLT.2011.28

Howland, K., & Good, J. (2015). Learning to communicate computationally with Flip: A bi-modal programming language for game creation. *Computers & Education*, *80*, 224-240. https://doi.org/10.1016/j.compedu.2014.08.014

Hsieh, T. C., Lee, M. C., & Su, C. Y. (2013). Designing and implementing a personalized remedial learning system for enhancing the programming learning. *Journal of Educational Technology & Society*, *16*(4), 32-46.

Hsu, C. C., & Wang, T. I. (2018). Applying game mechanics and student-generated questions to an online puzzle-based game learning system to promote algorithmic thinking skills. *Computers & Education*, *121*, 73-88. https://doi.org/10.1016/j.compedu.2018.02.002

Hsu, Y. C., & Ching, Y. H. (2013). Mobile app design for teaching and learning: Educators’ experiences in an online graduate course. *International Review of Research in Open and Distributed Learning*, *14*(4), 117-139. https://doi.org/10.19173/irrodl.v14i4.1542

Hwang, W. Y., Shadiev, R., Wang, C. Y., & Huang, Z. H. (2012). A pilot study of cooperative programming learning behavior and its relationship with students' learning performance. *Computers & education*, *58*(4), 1267-1281. https://doi.org/10.1016/j.compedu.2011.12.009

Iqbal Malik, S., & Coldwell-Neilson, J. (2017). Impact of a new teaching and learning approach in an introductory programming course. *Journal of Educational Computing Research*, *55*(6), 789-819. https://doi.org/10.1177/0735633116685852

Jakoš, F., & Verber, D. (2017). Learning basic programing skills with educational games: A case of primary schools in Slovenia. *Journal of Educational Computing Research*, *55*(5), 673-698. https://doi.org/10.1177/0735633116680219

Jamil, M. G., & Isiaq, S. O. (2019). Teaching technology with technology: approaches to bridging learning and teaching gaps in simulation-based programming education. *International Journal of Educational Technology in Higher Education*, *16*(1), 1-21. https://doi.org/10.1186/s41239-019-0159-9

Judson, E., & Glassmeyer, K. (2019). Are Teachers and Schools Ready to Accept Computer Science as a Graduation Requirement?. *Journal of Research on Technology in Education*, *51*(4), 311-325. https://doi.org/10.1080/15391523.2019.1624661

Kátai, Z. (2015). The challenge of promoting algorithmic thinking of both sciences-and humanities-oriented learners. *Journal of Computer Assisted Learning*, *31*(4), 287-299. https://doi.org/10.1111/jcal.12070

Kazakoff, E. R., & Bers, M. U. (2014). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Journal of Educational Computing Research*, *50*(4), 553-573. https://doi.org/10.2190/EC.50.4.f

Kim, S., Chung, K., & Yu, H. (2013). Enhancing digital fluency through a training program for creative problem solving using computer programming. *The Journal of Creative Behavior*, *47*(3), 171-199. https://doi.org/10.1002/jocb.30

Kim, B., Kim, T., & Kim, J. (2013). and-pencil programming strategy toward computational thinking for non-majors: Design your solution. *Journal of Educational Computing Research*, *49*(4), 437-459. https://doi.org/10.2190/EC.49.4.b

Koorsse, M., Cilliers, C., & Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. *Computers & Education*, *82*, 162-178. https://doi.org/10.1016/j.compedu.2014.11.020

Kucuk, S., & Sisman, B. (2017). Behavioral patterns of elementary students and teachers in one-to-one robotics instruction. *Computers & Education*, *111*, 31-43. https://doi.org/10.1016/j.compedu.2017.04.002

Kuo, W. C., & Hsu, T. C. (2020). Learning computational thinking without a computer: How computational participation happens in a computational thinking board game. *The Asia-Pacific Education Researcher*, *29*(1), 67-83. https://doi.org/10.1007/s40299-019-00479-9

Kuo, F. Y., Wu, W. H., & Lin, C. S. (2013). An investigation of self-regulatory mechanisms in learning to program Visual Basic. *Journal of Educational Computing Research*, *49*(2), 225-247. https://doi.org/10.2190/EC.49.2.f

Lazar, T., Sadikov, A., & Bratko, I. (2017). Rewrite rules for debugging student programs in programming tutors. *IEEE Transactions on Learning Technologies, 11(4),* 429-440. https://doi.org/10.1109/TLT.2017.2743701

Lee, V. C., Yu, Y. T., Tang, C. M., Wong, T. L., & Poon, C. K. (2018). ViDA: A virtual debugging advisor for supporting learning in computer programming courses. *Journal of Computer Assisted Learning*, *34*(3), 243-258. https://doi.org/10.1111/jcal.12238

Leonard, J., Mitchell, M., Barnes-Johnson, J., Unertl, A., Outka-Hill, J., Robinson, R., & Hester-Croff, C. (2018). Preparing teachers to engage rural students in computational thinking through robotics, game design, and culturally responsive teaching. *Journal of Teacher Education*, *69*(4), 386-407. https://doi.org/10.1177/0022487117732317

Lepp, M., Palts, T., Luik, P., Papli, K., Suviste, R., Säde, M., ... & Tõnisson, E. (2018). Troubleshooters for tasks of introductory programming MOOCs. *International Review of Research in Open and Distributed Learning*, *19*(4). https://doi.org/10.19173/irrodl.v19i4.3639

Lin, G. Y. (2016). Self-efficacy beliefs and their sources in undergraduate computing disciplines: An examination of gender and persistence. *Journal of Educational Computing Research*, *53*(4), 540-561. https://doi.org/10.1177/0735633115608440

Lin, C. L., Liang, J. C., Su, Y. C., & Tsai, C. C. (2013). Exploring the relationships between self-efficacy and preference for teacher authority among computer science majors. *Journal of Educational Computing Research*, *49*(2), 189-207. https://doi.org/10.2190/EC.49.2.d

Liu, T. Y. (2016). Using educational games and simulation software in a computer science course: learning achievements and student flow experiences. *Interactive Learning Environments*, *24*(4), 724-744. https://doi.org/10.1080/10494820.2014.917109

Lu, O. H., Huang, J. C., Huang, A. Y., & Yang, S. J. (2017). Applying learning analytics for improving students engagement and learning outcomes in an MOOCs enabled collaborative programming course. *Interactive Learning Environments*, *25*(2), 220-234. https://doi.org/10.1080/10494820.2016.1278391

Luik, P., Suviste, R., Lepp, M., Palts, T., Tõnisson, E., Säde, M., & Papli, K. (2019). What motivates enrolment in programming MOOCs?. *British Journal of Educational Technology*, *50*(1), 153-165. https://doi.org/10.1111/bjet.12600

Luo, F., Antonenko, P. D., & Davis, E. C. (2020). Exploring the evolution of two girls' conceptions and practices in computational thinking in science. *Computers & Education*, *146*, 1-12. https://doi.org/10.1016/j.compedu.2019.103759

Mathrani, A., Christian, S., & Ponder-Sutton, A. (2016). PlayIT: Game based learning approach for teaching programming concepts. *Journal of Educational Technology & Society*, *19*(2), 5-17.

Melander Bowden, H. (2019). Problem-solving in collaborative game design practices: epistemic stance, affect, and engagement. *Learning, Media and Technology*, *44*(2), 124-143. https://doi.org/10.1080/17439884.2018.1563106

Moons, J., & De Backer, C. (2013). The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism. *Computers & Education*, *60*(1), 368-384. https://doi.org/10.1016/j.compedu.2012.08.009

Moreno, J. (2012). Digital competition game to improve programming skills. *Journal of Educational Technology & Society*, *15*(3), 288-297.

Mouza, C., Yang, H., Pan, Y. C., Ozden, S. Y., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*, *33*(3), 61-76. https://doi.org/10.14742/ajet.3521

Mouza, C., Marzocchi, A., Pan, Y. C., & Pollock, L. (2016). Development, implementation, and outcomes of an equitable computer science after-school program: Findings from middle-school students. *Journal of Research on Technology in Education*, *48*(2), 84-104. https://doi.org/10.1080/15391523.2016.1146561

Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting plans to action: The effects of a card-coded robotics curriculum and activities on Korean kindergartners. *The Asia-Pacific Education Researcher*, *28*(5), 387-397. https://doi.org/10.1007/s40299-019-00438-4

Navarrete, C. C. (2013). Creative thinking in digital game design and development: A case study. *Computers & Education*, *69*, 320-331. https://doi.org/10.1016/j.compedu.2013.07.025

Nemiro, J., Larriva, C., & Jawaharlal, M. (2017). Developing creative behavior in elementary school students with robotics. *The Journal of Creative Behavior*, *51*(1), 70-90. https://doi.org/10.1002/jocb.87

Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, *68*(1), 463-484. https://doi.org/10.1007/s11423-019-09708-w

Olelewe, C. J., & Agomuo, E. E. (2016). Effects of B-learning and F2F learning environments on students' achievement in QBASIC programming. *Computers & Education*, *103*, 76-86. https://doi.org/10.1016/j.compedu.2016.09.012

Ortin, F., Redondo, J. M., & Quiroga, J. (2017). Design and evaluation of an alternative programming paradigms course. *Telematics and Informatics*, *34*(6), 813-823. https://doi.org/10.1016/j.tele.2016.09.014

Pala, F. K., & Mıhçı Türker, P. (2019). The effects of different programming trainings on the computational thinking skills. *Interactive Learning Environments*, 1-11. https://doi.org/10.1080/10494820.2019.1635495

Panskyi, T., Rowinska, Z., & Biedron, S. (2019). Out-of-school assistance in the teaching of visual creative programming in the game-based environment–Case study: Poland. *Thinking Skills and Creativity*, *34*, 1-14. https://doi.org/10.1016/j.tsc.2019.100593

Peel, A., Sadler, T. D., & Friedrichsen, P. (2019). Learning natural selection through computational thinking: Unplugged design of algorithmic explanations. *Journal of Research in Science Teaching*, *56*(7), 983-1007. https://doi.org/10.1002/tea.21545

Pellas, N., & Peroutseas, E. (2016). Gaming in Second Life via Scratch4SL: Engaging high school students in programming courses. *Journal of Educational Computing Research*, *54*(1), 108-143. https://doi.org/10.1177/0735633115612785

Pellas, N. (2017). An exploration of interrelationships among presence indicators of a community of inquiry in a 3D game-like environment for high school programming courses. *Interactive Learning Environments*, *25*(3), 343-360. https://doi.org/10.1080/10494820.2015.1127819

Peng, J., Wang, M., Sampson, D., & van Merriënboer, J. J. (2019). Using a visualisation-based and progressive learning environment as a cognitive tool for learning computer programming. *Australasian Journal of Educational Technology*, *35*(2), 52-68. https://doi.org/10.14742/ajet.4676

Peteranetz, M. S., Flanigan, A. E., Shell, D. F., & Soh, L. K. (2018). Career aspirations, perceived instrumentality, and achievement in undergraduate computer science courses. *Contemporary Educational Psychology*, *53*, 27-44. https://doi.org/10.1016/j.cedpsych.2018.01.006

Pérez-Marín, D., & Pascual-Nieto, I. (2012). A case study on the use of blended learning to encourage computer science students to study. *Journal of Science Education and Technology*, *21*(1), 74-82. https://doi.org/10.1007/s10956-011-9283-6

Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., & Wartella, E. A. (2019). Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. Computers & Education, 128, 52-62. https://doi.org/10.1016/j.compedu.2018.09.006

Pilkington, C. (2018). A playful approach to fostering motivation in a distance education computer programming course: Behaviour change and student perceptions. *The International Review of Research in Open and Distributed Learning*, *19*(3), 282-298. https://doi.org/10.19173/irrodl.v19i3.3664

Qian, Y., Hambrusch, S., Yadav, A., Gretter, S., & Li, Y. (2020). Teachers' perceptions of student misconceptions in introductory programming. *Journal of Educational Computing Research*, *58*(2), 364-397. https://doi.org/10.1177/0735633119845413

Rodrigo, M. M. T., Andallaza, T. C. S., Castro, F. E. V. G., Armenta, M. L. V., Dy, T. T., & Jadud, M. C. (2013). An analysis of java programming behaviors, affect, perceptions, and syntax errors among low-achieving, average, and high-achieving novice programmers. *Journal of Educational Computing Research*, *49*(3), 293-325. https://doi.org/10.2190/EC.49.3.b

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(1), 1-15. https://doi.org/10.1186/s41239-017-0080-z

Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, *97*, 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

Sáez-López, J. M., & Sevillano-García, M. L. (2017). Sensors, programming and devices in Art Education sessions. One case in the context of primary education/Sensores, programación y dispositivos en sesiones de Educación Artística. Un caso en el contexto de Educación Primaria. *Cultura y educación*, *29*(2), 350-384. https://doi.org/10.1080/11356405.2017.1305075

Saritepeci, M. (2020). Developing computational thinking skills of high school students: Design-based learning activities and programming tasks. *The Asia-Pacific Education Researcher*, *29*(1), 35-54. https://doi.org/10.1007/s40299-019-00480-2

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. *The Asia-Pacific Education Researcher*, *29*(1), 55-66. https://doi.org/10.1007/s40299-019-00478-w

Smith, S., & Chan, S. (2017). Collaborative and competitive video games for teaching computing in higher education. *Journal of Science Education and Technology*, *26*(4), 438-457.https://doi.org/10.1007/s10956-017-9690-4

Shaw, R. S. (2019). The learning performance of different knowledge map construction methods and learning styles moderation for programming language learning. *Journal of Educational Computing Research*, *56*(8), 1407-1429. https://doi.org/10.1177/0735633117744345

Shadiev, R., Hwang, W. Y., Yeh, S. C., Yang, S. J., Wang, J. L., Han, L., & Hsu, G. L. (2014). Effects of unidirectional vs. reciprocal teaching strategies on web-based computer programming learning. *Journal of educational computing research*, *50*(1), 67-95. https://doi.org/10.2190/EC.50.1.d

Shi, N., Cui, W., Zhang, P., & Sun, X. (2018). Evaluating the effectiveness roles of variables in the novice programmers learning. *Journal of educational computing research*, *56*(2), 181-201. https://doi.org/10.1177/0735633117707312

Shimic, G., & Jevremovic, A. (2012). Problem-based learning in formal and informal learning environments. *Interactive Learning Environments*, *20*(4), 351-367. https://doi.org/10.1080/10494820.2010.486685

Sisman, B., Gunay, D., & Kucuk, S. (2019). Development and validation of an educational robot attitude scale (ERAS) for secondary school students. *Interactive Learning Environments*, *27*(3), 377-388. https://doi.org/10.1080/10494820.2018.1474234

So, H. J., Kim, D., & Ryoo, D. (2020). Trajectories of Developing Computational Thinking Competencies: Case Portraits of Korean Gifted Girls. *The Asia-Pacific Education Researcher*, *29*(1), 85-100. https://doi.org/10.1007/s40299-019-00459-z

Su, A. Y., Yang, S. J., Hwang, W. Y., Huang, C. S., & Tern, M. Y. (2014). Investigating the role of computer-supported annotation in problem-solving-based teaching: An empirical study of a Scratch programming pedagogy. *British Journal of Educational Technology*, *45*(4), 647-665. https://doi.org/10.1111/bjet.12058

Taylor, M. S., Vasquez, E., & Donehower, C. (2017). Computer programming with early elementary students with Down syndrome. *Journal of Special Education Technology*, *32*(3), 149-159. https://doi.org/10.1177/0162643417704439

Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education, 120,* 64-74. https://doi.org/10.1016/j.compedu.2018.01.011

Xinogalos, S. (2012). An evaluation of knowledge transfer from microworld programming to conventional programming. *Journal of Educational Computing Research*, *47*(3), 251-277. https://doi.org/10.2190/EC.47.3.b

Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development*, *67*(3), 541-575. https://doi.org/10.1007/s11423-018-9622-x

Xinogalos, S., Satratzemi, M., Chatzigeorgiou, A., & Tsompanoudi, D. (2019). Factors affecting students' performance in distributed pair programming. *Journal of Educational Computing Research*, *57*(2), 513-544. https://doi.org/10.1177/0735633117749432

Taylor, M. S. (2018). Computer programming with Pre-K through first-grade students with intellectual disabilities. *The journal of special education*, *52*(2), 78-88. https://doi.org/10.1177/0022466918761120

Teng, C. H., Chen, J. Y., & Chen, Z. H. (2018). Impact of augmented reality on programming language learning: Efficiency and perception. *Journal of Educational Computing Research*, *56*(2), 254-271. https://doi.org/10.1177/0735633117706109

Tekdal, M. (2013). The effect of an example-based dynamic program visualization environment on students' programming skills. *Journal of Educational Technology & Society*, *16*(3), 400-410.

Tomić, B., Jovanović, J., Milikić, N., Devedžić, V., Dimitrijević, S., Đurić, D., & Ševarac, Z. (2019). Grading students' programming and soft skills with open badges: A case study. *British Journal of Educational Technology*, *50*(2), 518-530. https://doi.org/10.1111/bjet.12564

Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research*, *57*(1), 3-31. https://doi.org/10.1177/0735633117743918

Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2019). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research*, *56*(8), 1345-1360. https://doi.org/10.1177/0735633117746747

Van Niekerk, J., & Webb, P. (2016). The effectiveness of brain-compatible blended learning material in the teaching of programming logic. *Computers & Education*, *103*, 16-27. https://doi.org/10.1016/j.compedu.2016.09.008

Vasilopoulos, I. V., & Van Schaik, P. (2019). Koios: Design, development, and evaluation of an educational visual tool for Greek novice programmers. *Journal of Educational Computing Research*, *57*(5), 1227-1259. https://doi.org/10.1177/0735633118781776

Veletsianos, G., Beth, B., Lin, C., & Russell, G. (2016). Design principles for thriving in our digital world: A high school computer science course. Journal of Educational Computing Research, 54(4), 443-461. https://doi.org/10.1177/0735633115625247

Veerasamy, A. K., D'Souza, D., Lindén, R., & Laakso, M. J. (2018). The impact of prior programming knowledge on lecture attendance and final exam. *Journal of Educational Computing Research*, *56*(2), 226-253. https://doi.org/10.1177/0735633117707695

Veerasamy, A. K., D'Souza, D., Lindén, R., & Laakso, M. J. (2019). Relationship between perceived problem-solving skills and academic performance of novice learners in introductory programming courses. *Journal of Computer Assisted Learning*, *35*(2), 246-255. https://doi.org/10.1111/jcal.12326

Verdú, E., Regueras, L. M., Verdú, M. J., Leal, J. P., de Castro, J. P., & Queirós, R. (2012). A distributed system for learning programming on-line. Computers & Education, 58(1), 1-10. https://doi.org/10.1016/j.compedu.2011.08.015

Vieira, C., Magana, A. J., Roy, A., & Falk, M. L. (2019). Student explanations in the context of computational science and engineering education. *Cognition and Instruction*, *37*(2), 201-231. https://doi.org/10.1080/07370008.2018.1539738

Vosinakis, S., Anastassakis, G., & Koutsabasis, P. (2018). Teaching and learning logic programming in virtual worlds using interactive microworld representations. *British Journal of Educational Technology*, *49*(1), 30-44. https://doi.org/doi:10.1111/bjet.12531

Wang, Y., Liang, Y., Liu, L., & Liu, Y. (2016). A multi-peer assessment platform for programming language learning: considering group non-consensus and personal radicalness. *Interactive Learning Environments*, *24*(8), 2011-2031. https://doi.org/10.1080/10494820.2015.1073748

Wang, Y., Li, H., Feng, Y., Jiang, Y., & Liu, Y. (2012). Assessment of programming language learning based on peer code review model: Implementation and experience report. *Computers & Education*, *59*(2), 412-422. https://doi.org/10.1016/j.compedu.2012.01.007

Weintrop, D., & Wilensky, U. (2019). Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms. *Computers & Education*, *142*, 103646. https://doi.org/10.1016/j.compedu.2019.103646

Whitney, M., Lipford, H. R., Chu, B., & Thomas, T. (2018). Embedding secure coding instruction into the ide: complementing early and intermediate CS courses with ESIDE. *Journal of Educational Computing Research*, *56*(3), 415-438. https://doi.org/10.1177/0735633117708816

Witherspoon, E. B., Schunn, C. D., Higashi, R. M., & Shoop, R. (2018). Attending to structural programming features predicts differences in learning and motivation. *Journal of Computer Assisted Learning*, *34*(2), 115-128. https://doi.org/10.1111/jcal.12219

Wu, B., Hu, Y., Ruis, A. R., & Wang, M. (2019). Analysing computational thinking in collaborative programming: A quantitative ethnography approach. *Journal of Computer Assisted Learning*, *35*(3), 421-434. https://doi.org/10.1111/jcal.12348

Wu, L., Looi, C. K., Multisilta, J., How, M. L., Choi, H., Hsu, T. C., & Tuomi, P. (2020). Teacher's perceptions and readiness to teach coding skills: a comparative study between Finland, Mainland China, Singapore, Taiwan, and South Korea. *The Asia-Pacific Education Researcher*, *29*(1), 21-34. https://doi.org/10.1007/s40299-019-00485-x

Yağcı, M. (2018). Web-mediated problem-based learning and computer programming: Effects of study approach on academic achievement and attitude. *Journal of Educational Computing Research*, *56*(2), 272-292. https://doi.org/10.1177/0735633117706908

Yildiz Durak, H. (2018). Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning*, *34*(6), 740-752. https://doi.org/10.1111/jcal.12281

Yildiz Durak, H. (2018). Flipped learning readiness in teaching programming in middle schools: Modelling its relation to various variables. *Journal of Computer Assisted Learning*, *34*(6), 939-959. https://doi.org/10.1111/jcal.12302

Yukselturk, E., & Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, *48*(3), 789-801. https://doi.org/10.1111/bjet.12453

Zendler, A., & Klaudt, D. (2012). Central computer science concepts to research-based teacher training in computer science: An experimental study. *Journal of Educational Computing Research*, *46*(2), 153-172. https://doi.org/10.2190/EC.46.2.c

Zendler, A., Klaudt, D., & Seitz, C. (2014). Empirical determination of competence areas to computer science education. *Journal of Educational Computing Research*, *51*(1), 71-89. https://doi.org/10.2190/EC.51.1.d

Zendler, A., & Reile, S. (2018). The effect of reciprocal teaching and programmed instruction on learning outcome in computer science education. *Studies in Educational Evaluation*, *58*, 132-144. https://doi.org/10.1016/j.stueduc.2018.05.008

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, *53*(4), 562-590. https://doi.org/10.1177/0735633115608444

Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills?. *Computers & Education*, *141*, 103633. https://doi.org/10.1016/j.compedu.2019.103633

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2017). Investigating the period of switching roles in pair programming in a primary school. *Journal of Educational Technology & Society*, *20*(3), 220-233.

Zhong, B., & Li, T. (2020). Can pair learning improve students' troubleshooting performance in robotics education?. *Journal of Educational Computing Research*, *58*(1), 220-248. https://doi.org/10.1177/0735633119829191

Zingaro, D., & Porter, L. (2014). Peer instruction in computing: The value of instructor intervention. *Computers & Education*, *71*, 87-96. https://doi.org/10.1016/j.compedu.2013.09.015