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

Atajan Rovshenov\textsuperscript{a}\textsuperscript{*}, Fırat Sarsar \textsuperscript{a}

\textsuperscript{a} Ege University, Turkiye.


### 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.

### 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.

### Article Info:
- **Review Article**

### Keywords:
- Programming education, programming teaching, Computer science education, Systematic review

### 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

\textsuperscript{*} Corresponding author. Department of Computer Education and Instructional Technology, Ege University, Turkey.
e-mail address: rovshenovatajan@gmail.com

Doi: [http://doi.org/10.31681/jetol.1201010](http://doi.org/10.31681/jetol.1201010)

Received 8 Nov 2022; Revised 27 Dec 2022; Accepted 1 Jan 2023

ISSN: 2618-6586. This is an open Access article under the CC BY license.
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.

![Diagram](image)

**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.

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

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.

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Education Level</th>
<th>Time Span</th>
<th>Type of Publication</th>
</tr>
</thead>
<tbody>
<tr>
<td>children’s programming</td>
<td>Pre-School</td>
<td>January 2012-February 2020</td>
<td>Experimental Studies</td>
</tr>
<tr>
<td>computing education</td>
<td>Elementary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>computer science education</td>
<td>Secondary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>programming teaching</td>
<td>High-School</td>
<td></td>
<td></td>
</tr>
<tr>
<td>programming instruction</td>
<td>University</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pair programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>novice programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>introductory programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>teaching programming concepts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>block-based programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>programming training</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.](image)

![Fig. 4. Distribution of the studies on programming education by country.](image)

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.

![Methodological tendencies of the studies in programming education](image)

**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.**

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

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.](image)

### 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.](image)

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
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.

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.
Regarding the limitations of the studies on programming education, it was found that sample size 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.

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 the results are consistent (Alaqsaam, 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 & Chakroborty, 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, Türkiye, 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 & Shuja (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, Håkkinen, Vesisenaho & Viiri, 2021; Grotta & Proda, 2019; Grover & Pea, 2013; Kalelioğlu, Gülbahar & Kuluk, 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; Grettér & 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 & Wang, 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 (Crelin, 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


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](https://doi.org/10.1109/ICALT.2014.186)


**Appendix**

**Appendix A. Selected Studies**


course. *Interactive Learning Environments*, 25(2), 220-234. [https://doi.org/10.1080/10494820.2016.1278391](https://doi.org/10.1080/10494820.2016.1278391)


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](https://doi.org/10.2190/EC.47.3.b)


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...


