Current Status and Future Perspectives in Articles about Coding Learning at Pre-University Level Published from 2009 to 2017: A Content Analysis Study

Hacer Özyurt[a], Özcan Özyurt[b] and Sefa Aras[c]

[a]Karadeniz Technical University, Of Technology Faculty, Trabzon/Türkiye (ORCID: 0000-0001-8621-2335)
[b]Karadeniz Technical University, Of Technology Faculty (ORCID: 0000-0002-0047-6813)
[c]Karadeniz Technical University, Of Technology Faculty (ORCID: 0000-0002-4043-3754)

Article History: Received: 1 November 2018; Accepted: 22 March 2019; Published online: 3 May 2019

Abstract: In recent years, remarkable attention has been paid to coding and computer programming learning and teaching at pre-university level. Many countries are taking important steps in this field every passing year. Examining the studies on coding learning enriched by use of different tools based on specific periods may be guiding for researchers in this field, in which there is an ever-growing interest. In this regard, the purpose of this study is to make a thorough examination of the articles on coding learning and teaching at pre-university level published in international journals from 2009 to 2017 and to show the general trend. The study examined 34 articles published in this field from 2009 to 2017 through content analysis. The data collected by use of Article Classification Form were descriptively analyzed. The research findings show that most articles published in this field covered robotic tools at elementary education level. They also indicate that quantitative – quasi-experimental method was the most preferred method, and various data collection tools, achievements tests being in the first place, were employed. The majority of studies reached positive results from different perspectives while few studies reported neutral results. The results of the present study are considered to be revealing the current situation in this field and lighting the way for other studies to be carried out in the future.

Keywords: Coding learning, programming learning, content analysis

DOI: 10.16949/turkbilmat.477216

1. Introduction

The development of technology has made computers an integral part of everyday life. Application software is the main component that makes computers functional machines. Software technologies are developing and diversifying day by day. However, coding structure and coding logic, which underlie software, remain largely the same. Programming (i.e. coding) is defined as the process of writing a computer programme (Guzdial, 2015). The terms of coding and programming are usually used interchangeably (These two terms are used synonymously in the following parts of this paper). Studies in recent years indicate programming skill among the most important skills to be given to individuals (Guzdial, 2015; Topalli & Cagiltay, 2018). Programming skill is not limited to capability to develop a computer programme. It also requires skills such as high-level thinking, capability to produce different solutions to problems, and ability to establish

Corresponding author: Özcan Özyurt **e-posta:** oozyurt@ktu.edu.tr

cause and effect relationships (Yükseltürk & Altıok, 2015). Apart from this, computational thinking skill, which was introduced by Wing (2006) and has become more important in recent years, holds an important place in programming learning (Guzdial, 2005). It is emphasized that computational thinking skill is a combination of different thinking skills such as creative thinking, algorithmic thinking, critical thinking, problem-solving, cooperative learning, and communication skills (Barr, Harrison & Conery, 2011). Enhancing students' computational thinking skills is accepted as an important goal worldwide (Guzdial, 2015). There are studies establishing a relationship between computational thinking skill and programming. As a matter of fact, Wing (2008) states that students are exposed to computational thinking during programming.

The literature contains studies reporting that even undergraduate students consider programming as a challenging class (Başer, 2013; Pillay & Jugoo, 2005). Students' perceiving programming as difficult causes them to develop a negative attitude towards programming (Başer, 2013). Negative attitude towards programming, in turn, may negatively affect students' achievement. As a matter of fact, there are studies concluding that factors such as negative perception, motivation, and especially low attitude may have a negative effect on computer programming learning (Anastasiadou & Karakos, 2011; Hawi, 2010; Korkmaz & Altun, 2013). Besides attitude, self-efficacy perception is also influential on students' achievement in computer programming classes. Indeed, the literature contains studies reporting that students with a low self-efficacy in programming may fail in programming classes (Altun & Mazman, 2012; Aşkar & Davenport, 2009). Also, the fact that programming learning is a difficult process and requires certain high-level thinking skills may cause individuals that begin programming learning at old ages to have difficulty in this field (Kalelioğlu, 2015). Consistently with this, there are studies stressing the importance of beginning programming learning at young ages (Kalelioğlu, 2015; Kazakoff, Sullivan & Bers, 2013).

The literature includes studies concluding that coding learning at young ages improves children's high-level thinking skills such as problem-solving, creative thinking, critical thinking, logical thinking, and algorithmic thinking (Fessakis, Gouli & Mavroudi, 2013; Portelance, Strawhacker & Bers, 2016). Hence, importance is attached to coding learning and teaching at pre-university level (includes elementary, secondary and high school levels) across the world. In this respect, many countries have amended their curricula (Bers, Flannery, Kazakoff & Sullivan, 2014; Lee, Martin & Apone, 2014). For example, countries such as Finland, New Zealand, United Kingdom, and USA have included coding in their curricula for early ages considering PISA data (Akpınar & Altun, 2014; Guzdial, 2015). Likewise, countries such as the Czech Republic, Korea, China, Turkey, and Spain have been taking important steps for coding learning in recent years (Keçeci, Alan & Zengin, 2016).

1.1. Related Works

Various learning environments have been developed and are still being developed for coding learning at pre-university level. While some of these environments use real code structure, some use visual components such as drag & drop and jigsaw puzzle. Some others use both codes and visual components together (Özyurt, Özyurt & Aras, 2016). Among these environments, the most known ones are Scratch, ScratchJr, Code.org, and Lego (Kalelioğlu, 2015; Lin & Liu, 2012). Apart from the commonly used ones, there are also environments developed and used within the scope of a specific study (Denner, Werner & Ortiz, 2012; Sengupta, Farris & Wright, 2012). Additionally, there are studies aiming to teach coding through different robotic tools (Sullivan & Bers, 2016; Elkin, Sullivan & Bers, 2016; García & De la Rosa, 2016).

The literature involves various studies on programming learning at pre-university level by use of the existing coding learning environments. In one of such studies, Kalelioğlu (2015) conducted a study with k-12 students by using the Code.org platform. The researcher investigated the effect of Code.org on programming learning and its contribution to reflective thinking for problem-solving. The study found out that Code.org does not have an effect on reflective thinking for problem-solving, but it can be efficiently used in programming learning. Another study was a quasi-experimental one conducted on elementary school students by use of Scratch (Saez-Lopez, Roman-Gonzalez & Vazquez-Cano, 2016). The study explored the effect of Scratch on students' attitude towards programming and on their success in programming. The study findings indicated that Scratch has a positive effect on success in programming. The study also revealed that students display a positive attitude towards the Scratch environment and find it useful and funny. Another study involving the use of ScratchJr (Papadakis, Kalogiannakis & Zaranis, 2016) searched the effect of this environment on preschool students' programming learning and problem-solving skills. The findings obtained in the study demonstrated that programming learning improves students' problem-solving skills. Sullivan & Bers (2016) carried out a quasi-experimental study with preschool students by using the Kiwi robotic kit. They sought the learning feedbacks of programming through robotic kits. The findings showed that the robotic kit is instructive and useful. In another study, Denner, Werner & Ortiz (2012) conducted a quasi-experimental study with middle school girls by using the environment they developed. They investigated the effect of female students' programming their own games on their programming learning. They found out that students' programming their own games improves their coding skill.

The recent importance attached to coding learning at pre-university level makes the determination of current trends in this field important. The literature contains two content analysis studies in this field. The first one is Lye & Koh's (2014) study. They showed the current trends of empirical research dealing with the development of computational thinking by means of programming and provided possible implications for research and instruction. They examined 27 empirical articles covering k-12 students. They suggested that many articles reached positive results, and research in this field should be conducted from different perspectives. The second one is Çatlak, Tekdal & Baz's (2015) study. They examined the studies involving Scratch. Based on the findings of the articles they examined, they revealed the use of Scratch in programming learning and its effects on different thinking skills. Though the importance attached to programming learning at pre-university level is clear, there are quite few content analysis studies in this field, which is interesting enough. These studies are limited in both number and coverage. As a matter of fact, while the first study mentioned above only examined the articles about computational thinking and programming covering k-12 students, the second study only addressed the studies concerning Scratch. From this perspective, it is important to make an in-depth examination of the studies recently conducted in this rapidly growing field. This will allow for a broader perspective on the field. Considering all this, the present study is necessary and significant as it determines the current trend and lights the way for future studies. In this regard, the purpose of this study is to examine the articles about coding learning and teaching at pre-university level published in international journals from 2009 to 2017. The study covered the period starting from 2009 because studies in this field began to become widespread as of that year. As a matter of fact, the review yielded only a couple of articles published in this field before 2009. Accordingly, the articles published as of 2009 were chosen. The study sought answers to the following research questions:

a. What is the distribution of the articles published in the field of coding learning and teaching at pre-university level by journal and year?
b. What tools were used as learning environments in the published articles?
c. Which research methods were employed in the published articles?
d. Which data collection tools were employed in the published articles?
e. What were the sample levels and sizes in the published articles?
f. Which data analysis methods were employed in the published articles?
g. What purposes were pursued in the published articles?
h. What results were obtained in the published articles?
i. What were the relationships between tools used and sample levels in the published articles?
j. What were the relationships between tools used and methods in the published articles?
k. What were the relationships between tools used and data collection tools in the published articles?

2. Method

This study applied content analysis to the articles about coding learning at pre-university level (includes elementary, secondary and high school levels) published in international journals from 2009 to 2017. Content analysis involves organizing, categorizing, and comparing texts and obtaining results from them by this means (Cohen, Manion & Morrison, 2007). The content analysis method was chosen for this study because it combines data that are similar to one another in some respects on the basis of certain concepts and themes and turn them into readable forms for readers (Bauer, 2000). The following sections present details concerning the method and the procedure.

2.1. Data Collection Tools and Procedure

Widely used databases such as EBSCOhost Web, ERIC, Google Scholar, ISI Web of Knowledge, Sage, ScienceDirect, SpringerLink, and Wiley Interscience were used for reviewing the articles about coding learning and teaching at pre-university level contained in the literature. Determining key words was important to identify the articles in this field. Hence, a couple of sample articles were examined, and a large set of key words was created. By this means, it was aimed to access all the publications in the field. These key words are alphabetically listed below:

“children programming”, “computer programming”, “coding training”, “early childhood programming”, “educational robot”, “robotics”, “k-12 programming”, “kindergarten programming”, “programming”, “programming education”, “programming learning”, “programming environment”, “visual programming environment”.

Articles were searched on databases from 15 September to 15 October 2017. The purpose was to examine only the articles published in journals. Thus, among the reached documents, only those published as articles were taken into account. Documents such as conference papers, reports, and theses or dissertations were ignored. The articles obtained through the first review were subjected to preliminary examination by the researchers. With this examination, the researchers decided on the suitability of the articles in terms of sample size (pre-university) and research purpose. At the end of the preliminary examination, 34 articles were recorded. Thus, it was decided to subject 34 articles to content analysis. Appendix 1 provides the full list of the articles examined in the study in an alphabetical order by author. Prior to the content analysis, Article Classification Form (ACF) developed by the researchers was used for classifying the obtained articles. First, a draft form was created in accordance with the research questions and considering the different content analysis article classification forms available (Göktaş et al., 2012). Then the researchers and Turkish language experts worked on that form and finalized it. The final version of the article classification form has seven parts. The first part includes the title of the articles, the journal it was published in, the year it was published in, etc. The other parts cover the environment used/developed in the article, method, data collection tool, sample, data analysis method, and purpose/result, respectively. The final version of ACF is presented in Appendix 2.

2.2. Data Analysis

The data obtained from the articles examined through content analysis were analyzed by use of descriptive statistical methods such as percentage and frequency. Some actions were taken to improve the validity and reliability of the study. In this regard, first the articles were shared between the two researchers. In this stage, each researcher filled in ACFs for the articles he examined. In the second stage, each researcher checked the accuracy of the data obtained by the other researcher for the articles he examined. In case of any disagreement emerging during such checking process, the two researchers discussed the relevant point and made the final decision. In the last stage, the third researcher randomly selected ten articles from among the articles and checked the classification accuracy of these articles over ACFs. Then all the researchers worked together and calculated frequencies and percentages for the data in each ACF. These calculations were repeated in such a way that answers to each research question were covered. In the end, the obtained numerical data were presented in tables, charts, and graphs.

3. Findings

The data collected through ACF were analyzed within the context of the research questions. The findings obtained through analyses are presented.

3.1. Findings Concerning the Journals and the Years in which the Articles were Published

In the study, 34 articles published in 24 different international journals were examined. Figure 1 presents findings concerning the distribution of the articles published from 2009 to 2017 by journal. The Figure 1 shows the names of the journals in which two or more articles were published and the numbers of the articles published in such journals. All the journals in which only one article was published are presented under the category of “others”.

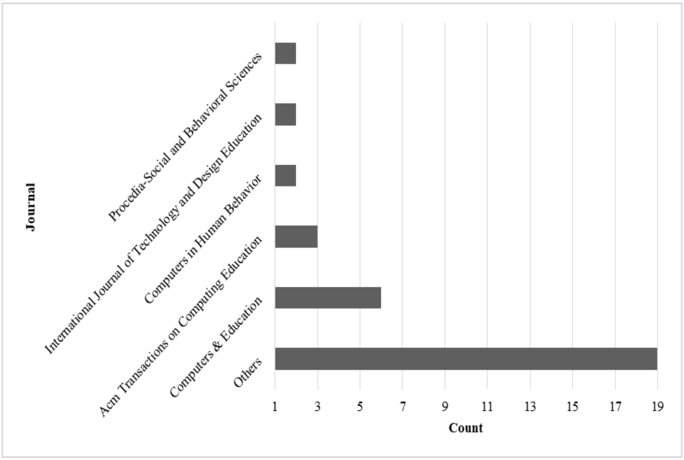


Figure 1. The list of the journals in which the articles were published

As shown in Figure 1, most articles (n=6) were published in Computers & Education journal. It is followed by ACM Transactions on Computing Education (n=3). The journals in which only one article was published are presented under the category of “other”. Thus, there are 19 journals under the category of “other”. Some examples to these journals are Educational Technology & Society, British Journal of Educational Technology, IEEE Transactions on Education, and Early Childhood Education Journal. Figure 2 presents the distribution of the articles by year of publication.

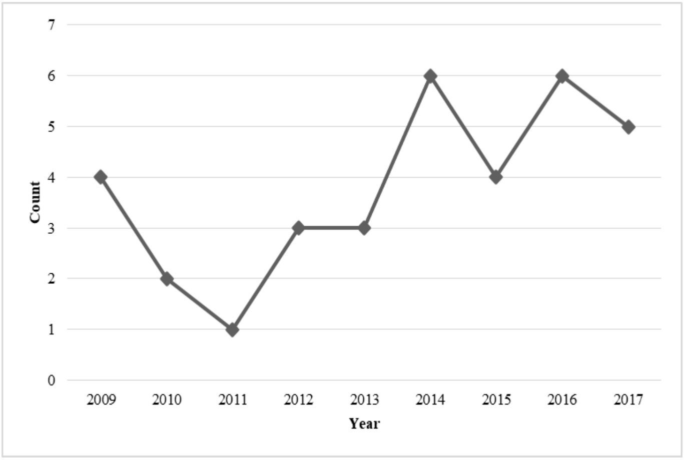


Figure 2. The distribution of the articles by year of publication

As shown in Figure 2, the publication frequency of the articles increased over the last years. As a matter of fact, the numbers of the articles published in 2014, 2015, 2016, and 2017 are six, four, six, and five, respectively. The number of the published articles considerably increased especially after 2013.

3.2. Findings Concerning the Learning Environments used in the Articles

Figure 3 presents findings concerning the learning environments used in the articles. According to Figure 3, eight articles about coding learning at pre-university level used robotic environments. Some examples to robotic environments are tools such as KIWI robotics kit, mBot, and TangibleK Robotics, which serve for teaching the logic of programming. Robotic environments are followed by Scratch, which was used in seven articles. The other environments used in the articles are Scratch Jr (n=2), Lego (n=2), and Code.org (n=1), respectively. Seven other articles presented and listed one or several existing coding environments. Still other seven articles presented the developmental processes of the environments developed by researchers themselves. Apart from widely known Scratch, ScratchJr, and Code.org, researchers developed the coding learning environments which they named as Grenfoot, PAT, and PiktoMir.

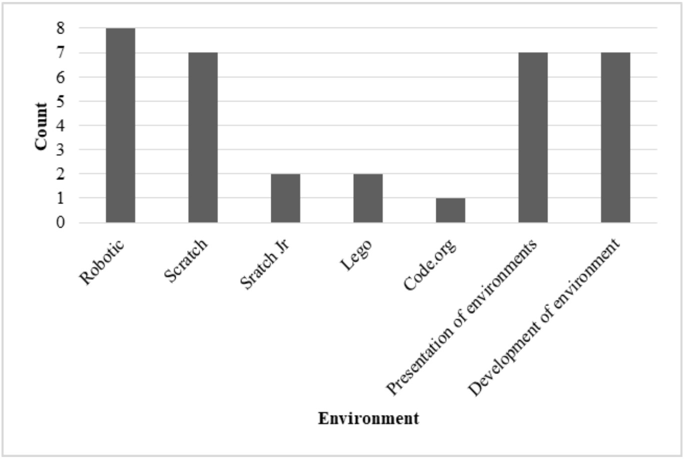


Figure 3. The list of the environments used in the articles

3.3. Findings Concerning the Research Methods Employed in the Articles

Figure 4 presents findings concerning the methods employed in the articles. As shown in Figure 4, the most preferred research method is (n=14) quantitative – quasi-experimental method. It is followed by qualitative (n=5) and mixed (n=5) methods.

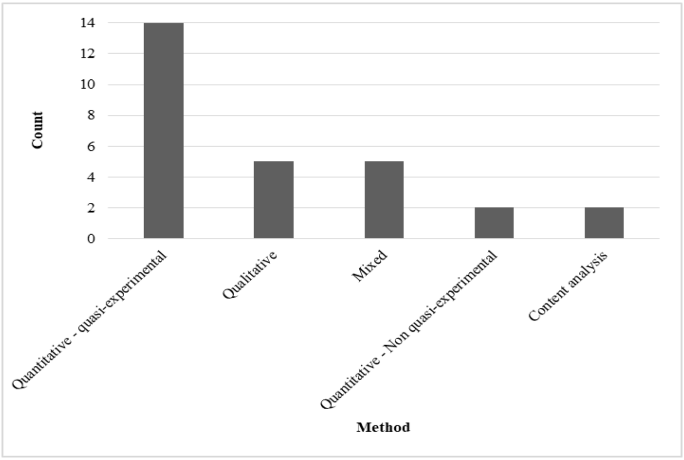


Figure 4. Research methods employed in the articles

The articles employing the data collection tools such as achievement tests, scales, and surveys are generally put under the category of quantitative methods. However, for a more elaborate perspective, the articles employing achievement tests and quasi-experimental design were classified as quantitative – quasi-experimental. On the other hand, the articles employing a survey/scale and not employing a quasi-experimental design were classified as quantitative – non-quasi-experimental. The methods employed in 28 articles are presented in Figure 4 above. This is because six articles did not contain a research method and did not collect any data about the existing or developed environments. These articles were aimed at presenting (n=5) and developing (n=1) environments, as shown under the previous sub-heading.

3.4. Findings Concerning the Data Collection Tools Employed in the Articles

Figure 5 presents findings concerning the data collection tools employed in the articles. A total of 45 data collection tools classified under six different categories were employed in the studies. The fact that the number of the data collection tools (n=45) is bigger than that of the articles (n=34) results from the collective use of more than one single tool in some studies. The most used data collection tool (n=17) is achievement test. It is followed by survey/scale (n=10) and observation (n=8), respectively.

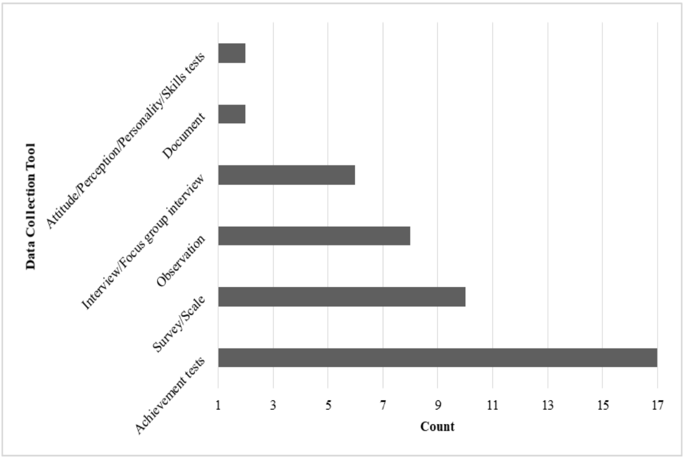
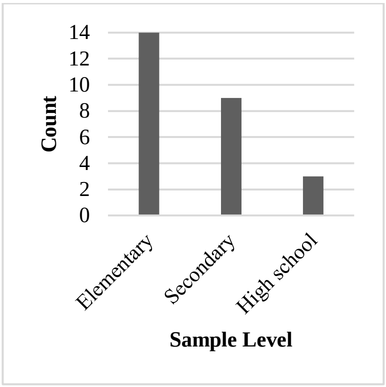
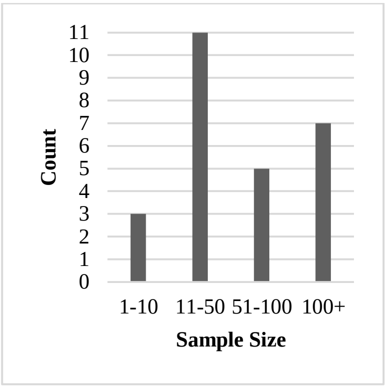


Figure 5. Data collection tools employed in the articles

3.5. Findings Concerning the Sample Levels and Sizes of the Articles

Figure 6 (a) and (b) present findings concerning the sample levels and sizes of the examined articles. Figure 6 (a) and (b) contain information about sample level and sample size for 26 articles in total. The most adopted sample level is elementary education level (n=14), which is followed by secondary education level (n=9) and high school level (n=3), respectively.





(a) (b)

Figure 6. Sample levels (a) and sample sizes (b) in the articles

As to sample size, the studies were largely conducted with groups of 11-50 people. As a matter of fact, 11 studies had a sample size of 11-50 people. They are followed by seven studies with a sample size of 100 or more people and five studies with a sample size of 51-100 people. That is to say, most studies (n=14) were carried out with elementary school students, and the most preferred sample size (n=11) is 11-50 people.

3.6. Findings Concerning the Data Analysis Methods Employed in the Articles

Figure 7 presents findings concerning the data analysis methods employed in the examined articles. As shown in Figure 7, a total of 55 analyses were made under seven different categories. The fact that the number of the data analysis methods (n=55) is bigger than that of the articles (n=34) results from the use of more than one analysis method in some studies. Mean/standard deviation analyses (n=13) were employed most, and they were followed by t-test (n=12) and other analyses.

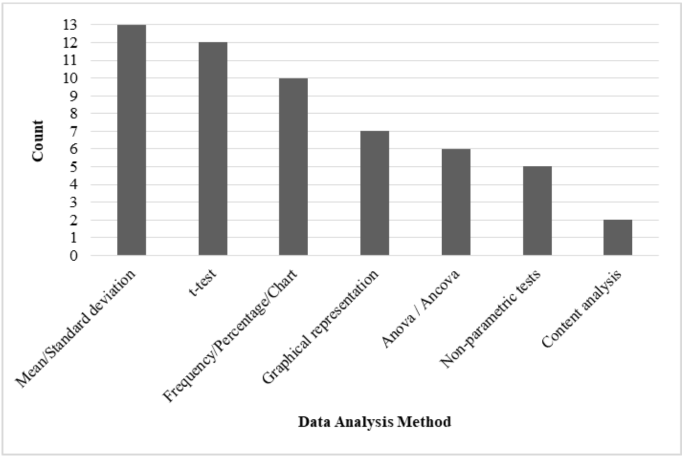


Figure 7. Data analysis methods in the articles

3.7. Findings Concerning the Purposes of the Articles

Table 1 present findings concerning the purposes of the articles examined in the study. As shown in Table 1, 52 purposes falling under 17 different categories were pursued in the studies. Table 1 shows these purposes in descending order (i.e. from the most repeated one to the least). The studies mostly (n=16) aimed to investigate the effect of coding learning environment(s) on coding learning. This is followed by the purpose of determining student views (n=8) and identifying the effect on problem-solving skill (n=6).

Table 1. The list of the purposes pursued in the articles

Rank	Purposes	n
1	Determining the effect on programming learning	16
2	Determining student views	8
3	Investigating the effect on problem-solving skill	6
4	Presenting an environment	5
5	Determining attitudes towards programming	3
6	Developing a measurement tool	2
7	Making a content analysis	2
8	Determining the effect on mathematics achievement	1
9	Determining the effect on computational thinking skill	1
10	Determining teacher views	1
11	Investigating the effect on reflective thinking for problem-solving	1
12	Investigating the effect on sorting skill	1
13	Investigating the effect on passing to high-level programming language	1
14	Investigating the effect of adding programming to the curriculum	1
15	Examining parent-child cooperation in programming learning	1
16	Developing an environment	1
17	Examining one-to-one programming learning	1

3.8. Findings Concerning the Results of the Articles

Table 2 lists the results of the articles subjected to content analysis in number and frequency. As shown in Table 2, the articles reached 74 results classified under 24 different categories. These results fell under two categories: positive and neutral. Of 24 different results, 19 are positive, and five are neutral. Overall, 65 positive results (f=87.84%) and nine neutral results (f=12.16%) were reached.

Table 2. Findings concerning the results of the articles

	Rank	Results	n	f (%)
Positive	1	Positive effect on programming learning	13	17.57
	2	Usability/Availability	10	13.51
	3	Amusing environment	9	12.16
	4	Satisfaction	5	6.76
	5	Enhancing interest	5	6.76
	6	Positive effect on problem-solving skill	5	6.76
	7	Increasing motivation	3	4.05
	8	Positive attitude towards programming	2	2.70
	9	Positive attitude towards project-based learning	2	2.70
	10	Developing a successful scale	2	2.70
	11	The necessity of teaching programming to middle school students	1	1.35
	12	Positive effect on sorting skill	1	1.35
	13	Positive effect on mathematics achievement	1	1.35
	14	Positive effect on computational thinking skill	1	1.35
	15	Understanding concepts related to computer sciences	1	1.35
	16	Contribution to rule-based learning	1	1.35
	17	Improvement of social and affective skills	1	1.35
	18	Active learning	1	1.35
	19	Raising awareness	1	1.35
Subtotal			65	87.84
Neutral	**20**	Not having an effect on programming learning	3	4.05
	21	Not having an effect on problem-solving skill	3	4.05
	22	Not having an effect on reflective thinking for problem-solving	1	1.35
	23	Not having an effect on attitude towards programming	1	1.35
	24	Not having an effect on passing to high-level programming language	1	1.35
Subtotal			9	12.16
Total			74	100.0

3.9. Findings Concerning the Relationships between the Tools used in the Articles and the Sample Levels/Methods/Data Collection Tools

Findings concerning the environments, sample levels, research methods, and data collection tools employed in the articles have been presented under separate sub-headings above. Cross analyses were made to reveal the relationships between the data under such sub-headings. In this way, an attempt was made to determine the relationships of the learning environments used in the articles with the sample sizes, research methods, and data collection tools adopted. Table 3 presents findings concerning answers to the last three research questions in detail.

Table 3. Findings concerning the relationships between the learning environments and the sample levels, research methods, and data collection tools

Environment / Sample level	Scratch	Scratch Jr	Code.org	Lego	Robotic	Presentation of environments	Development of environment	Total
Elementary	2*	1	1	1	6	-	3	14
Secondary	3	1	-	-	1	2	2	9
High school	1	-	-	1	-	-	1	3
Undefined	1	-	-	-	1	5	1	8
Total	7	2	1	2	8	7	7	34
Method								
Content analysis	1	-	-	-	-	1	-	2
Quantitative – Quasi-experimental	3	1	-	-	5	2	3	14
Quantitative – Non-quasi-experimental	-	-	-	-	1	-	1	2
Qualitative	1	-	-	2	1	-	1	5
Mixed	2	1	1	-	-	-	1	5
Total	7	2	1	2	7	3	6	28
Data Collection tools								
Observation	3	1	-	2	-	-	2	8
Interview/Focus group interview	3	-	1	1	1	-	-	6
Achievement tests	4	2	1	-	5	1	4	17
Attitude/Perception/Personality/Skills tests	1	-	-	-	1	-	-	2
Survey/Scale	3	1	1	-	2	1	2	10
Document	1	-	-	-	-	1	-	2
Total	15	4	3	3	9	3	8	45

* indicates the frequency.

The matrix in Table 3 was used for revealing the relationships between the environments used in the articles and these three variables (sample level, research method, and data collection tool). In this matrix, columns indicate environments, and rows indicate each one of these three variables. The number of the studies in which the entries (instrumental variable) in each intersecting row and column were used together according to the analysis results was written in each cell of the matrix. Just the mark "-" was put in the cell when there was no study with a row-column intersection. The sum of the cell values of rows and columns refers to total values for each row and column. A sample reading on Table 3 for the relationship between environment and sample level is as follows:

Row (Elementary education level): A total of 14 studies were conducted at elementary education level. Two of them used Scratch. Only one study was carried out with each one of Scratch Jr, Code.org, and Lego. Robotic tools were used in six studies. Three studies were aimed at developing an environment for elementary education level.

Column (Scratch): A total of seven studies used Scratch. Two of them were conducted with elementary school students, three with secondary education students, and one with high school students. A study using Scratch did not provide information about sample level (This study was made as a content analysis study).

The findings presented in Table 3 can be read as in the examples above. By this means, different inferences can be made with regards to the relationships of the environments with each variable (sample level, method, and data collection tool). With these inferences, it is possible have in-depth information about the studies subjected to content analysis. Sample inferences that can be made over these studies through the data provided in Table 3 can be listed as follows:

- *Mostly robotic tools were used in studies at elementary education level.*
- *Code.org was used only at elementary education level and only in one study.*
- *Specially developed environments were mostly developed at elementary education level.*
- *Scratch was mostly used in studies at secondary education level.*
- *At high school level, one study was conducted using Scratch; one study was conducted using Lego; and one study was conducted for developing an environment.*
- *Quantitative – quasi-experimental studies were mostly carried out with robotic tools. They were followed by studies using Scratch (n=3) and studies using the environments developed by researchers (n=3).*
- *No qualitative study was conducted with Scratch JR.*
- *All the data collection tools were employed in the studies involving Scratch.*
- *Observation and document were not employed as data collection tools in the studies in which robotic tools were used.*

4. Discussion, Conclusion, and Recommendations

In this study, 34 articles about coding learning at pre-university level published in international journals from 2009 to 2017 were subjected to content analysis. The articles were examined in terms of the number of articles, their distribution by year and journal, the environments used, research methods, data collection tools, sample levels, sample sizes, data analysis methods, purposes, and results. Also, the relationships of the environments used in the articles with three different variables (sample level, research method, and data collection tool) were revealed.

The study found out that the articles published in this field increased over the last years. This may be considered as an indicator of the importance attached to coding learning and teaching at pre-university level. As a matter of fact, the literature contains studies showing that important steps have been taken in this field across the world in recent years (Kazakoff, Sullivan & Bers, 2013; Kalelioğlu, 2015). Two of the journals with most publications in this field are Computers & Education and Computers in Human Behavior, which are journals with high impact factors. There are also a lot of other journals in which the articles in this field were published. Quality and diversity of the journals in which the articles were published may be regarded as an indicator of that research in this field is getting richer and richer.

Robotic tools and Scratch had a wide coverage in the articles examined. It is also remarkable that the literature contains studies in which coding environments developed by researchers for specific purposes were used apart from the known environments. There are also studies concluding that robotic activities head students to constructivist learning (Goh & Ali, 2014). This may be accounting for the frequent use of robotic tools in the studies. One reason for much use of the Scratch environment besides robotic tools may be that it is a strong, flexible, and rich environment in which different media components are used and is among the first applications developed in this field. Maloney et al. (2010) and Çatlak et al. (2015) also emphasized these aspects of the Scratch environment. Apart from that, the abundance of the coding environments developed by researchers for specific purposes may be deemed as an indicator of the importance attached to this field. Likewise, there are a considerable number of environments developed considering different pedagogical elements, besides the systems existing in the literature (Sengupta et al., 2012).

Among the methods employed in the articles examined, quantitative – quasi-experimental, qualitative, and mixed methods are much more common. The wide use of quantitative methods in the studies may have two main reasons. The first reason may be the quickness and easiness these methods provide relative to qualitative methods. As another reason, these methods may be used by very nature of such studies. The distribution of the data collection tools and statistical analyses in the studies also supports this. While achievement tests and survey/scale are the most preferred data collection tools, mean/standard deviation, t-test, and frequency/percentage/chart are the most used data analysis methods. The reason for this gets clearer when methods, data collection tools, and analyses are evaluated together. According to these results, it is possible to say that data collection tools and data analysis methods appropriate to research methods were employed. The reason for wide use of quantitative methods and data collection tools and data analysis methods relevant to them may be such advantages they provide as generalization of results, possibility of reaching a large sample size in a short time and obtaining data from such sample, and saving time and cost. Several content analysis studies in the literature report similar results as well (Alper & Gülbahar, 2009; Göktaş et al., 2012). The examined studies also adopted qualitative and mixed methods to some degree. According to Driscoll (1995), in parallel with the development of educational systems, different research methods should be blended in educational research rather than use of a single research method. In this regard, the use of mixed methods in the studies is noteworthy.

As to sample level and sample size, most studies were conducted at elementary education level, and the most frequent sample size was 11-50 people. That the most studies were conducted at elementary education level and they were followed by the studies at secondary education level is not surprising by nature of the studies. This is because there are a lot of studies stressing the importance of coding learning at early ages (Bers et al., 2014; Fessakis et al., 2013; Keçeci et al., 2016; Portelance et al., 2016). Though the most frequent sample size was 11-50 people, there are also a considerable number of studies with a sample size of 51-100 people and with a sample size of 100 or more people. The largeness of sample size in general may be associated with the frequency of quantitative studies. As a matter of fact, it is expected for sample size to be as large as possible in quantitative studies (Göktaş et al. 2012).

Lastly, the study reached remarkable results about the purposes and results of the examined studies. The studies mostly aimed to determine the effect on programming learning. This is an expected result. Because, in general, the effects of these studies on students' learning are investigated. Also, a variety of other purposes were pursued such as determining student views and developing a measurement tool. Another important purpose pursued in the studies was investigating the effects on attitudes, problem-solving skills, and certain thinking skills such as reflective thinking. The existence of research suggesting that coding learning at pre-university level will positively affect individuals' high-level thinking skills may have been influential on this. As a matter of fact, the literature contains studies suggesting that coding learning at young ages improves children's high-level thinking skills such as problem-solving, critical thinking, creative thinking, logical thinking, and algorithmic thinking (Çatlak et al., 2015; Fessakis et al., 2013; Kalelioğlu, 2015; Portelance et al., 2016). Such wide range of purposes pursued in the studies may be seen as an indicator of the importance and necessity of evaluating coding learning at pre-university level from different perspectives. The results of the studies may be evaluated within the context of their purposes. A lot of different results were obtained in the studies pursuing a wide range of different purposes. They mostly reached positive results. There were also neutral results implying a lack of positive effect, but they were limited in number. The primary positive results are positive effect on programming learning, usability/availability, and perception of these environments as funny and satisfactory by students. The fact that the results of the studies are largely positive from different perspectives reveals the value and importance of the studies in this field. This may also be expected to motivate future researchers to conduct different studies in this field.

4.1. Limitations

This study applied content analysis to the articles about coding learning at pre-university published in international journals from 2009 to 2017. We expect the results of this study to be useful in that they show the strengths and weaknesses of the studies conducted in this field. We also think that they may guide future studies by serving as an important source. However, the present study has certain limitations. The primary limitation is the coverage of the articles published only in international journals from 2009 to 2017. It should be remembered that the study is limited to 34 articles. In this regard, examination of different resources such as articles, conference papers, and theses or dissertations published at different dates in future research may provide a broader insight into the development and change of research in this field. Examination of only the studies about coding learning may be seen as another limitation. Publications on computational thinking skills, which are associated with this topic, may be covered in future studies. By this means, it may be possible to review the field from a larger variety of perspectives.

References

Akpınar, Y., & Altun, A. (2014). The need for programming education in information society schools. *Elementary Education Online, 13*(1), 1-4.

Alper, A., & Gülbahar, Y. (2009). Trends and issues in educational technologies: A review of recent research in TOJET. *The Turkish Online Journal of Educational Technology, 8*(2), 124-135.

Altun, A., & Mazman, S.G. (2012). Validity and reliability study of Turkish form of perception of self-efficacy perception related to programming. *Journal of Measurement and Evaluation in Education and Psychology, 3*(2), 297- 308

Anastasiadou, S.D., & Karakos, A.S. (2011). The beliefs of electrical and computer engineering students' regarding computer programming. *The International Journal of Technology, Knowledge and Society*, *7*(1), 37-51.

Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java programming among engineering students. *The Turkish Online Journal of Educational Technology*, *8*(1), 26-32.

Başer, M. (2013). Developing attitude scale toward computer programming. *International Journal of Social Science*, *6*(6), 199-215.

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology, 38*(6), 20-23.

Bauer, M. W. (2000). *Classical content analysis: A review*. In M.W. Bauer & G. Gaskell (Eds.), Qualitative researching with text, image and sound (pp. 131-151). London: Sage.

Bers, M. I., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*(2014), 145–157.

Cohen, L., Manion, L., & Morrison, (2000). *Research Methods in Education* (5th ed.). London: Routledge Falmer.

Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). The status of teaching programming with scratch: a document review work. *Journal of Instructional Technologies & Teacher Education, 4*(3), 13-25.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. *Computers & Education*, *58*(1), 240-249.

Driscoll, M. (1995). Paradigms for research in instructional systems. In Gary J. Anglin (Ed.). *Instructional technology: Past, present, and future* (2nd ed., pp. 322-329). Englewood, CO: Libraries. Unlimited.

Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO Robotics Kit in Preschool Classrooms. *Computers in the Schools*, *33*(3), 169-186.

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*(2013), 87-97.

García, P. G., & De la Rosa, F. (2016). RoBlock-Web app for programming learning. *International Journal of Emerging Technologies in Learning, 11*(12), 45-53.

Goh, H., & Ali, M.B. (2014). Robotics as a tool to stem learning. *International Journal for Innovation Education and Research, 2*(10), 66-78.

Göktaş, Y., Küçük, S, Aydemir M., Telli E., Arpacık, Ö., Yıldırım G., & Reisoğlu, İ. (2012). Educational technology research trends in Turkey: A content analysis of the 2000-2009 decade. *Educational Sciences: Theory & Practice, 12*(1), 191-196.

Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics, 8*(6), 1-165.

Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education, 54*(4), 1127-1136.

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

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal, 41*(4), 245-255.

Keçeci, G., Alan, B., & Zengin, F. K. (2016). Educational computer games assisted learning coding attitude scale: Validity and reliability study. *Education Sciences, 11*(4), 184-194.

Korkmaz, Ö. (2012). The Impact of critical thinking and logical-mathematical intelligence on algorithmic design skills. *Journal of Educational Computing Research, 46*(2), 173-193.

Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K–8 curriculum. *ACM Inroad, 5*(4), 64-71.

Lin, J. M. C., & Liu, S. F. (2012). An investigation into parent-child collaboration in learning computer programming. *Journal of Educational Technology & Society, 15*(1), 162-173.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior, 41*(2014), 51-61.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE), 10*(4), 1-15.

Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: Difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry, 5*(3), 1-27.

Özyurt, H., Özyurt Ö, & Aras S. (2016). *An examination of the environments where children can learn coding*, 10th International Computer & Instructional Technologies Symposium, 16-18 May 2016, Rize, Turkey.

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study. *International Journal of Mobile Learning and Organisation, 10*(3), 187-202.

Pillay, N. & Jugoo, V. R. (2005). An investigation into student characteristics affecting novice programming performance. *SIGCSE Bulletin 37*(4), 107-110.

Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2016). Constructing the ScratchJr programming language in the early childhood classroom. *International Journal of Technology and Design Education, 26*(4), 489-504.

Saez-Lopez, J. M., Roman-Gonzalez, M., & Vazquez-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*(2016), 129-141.

Sengupta, P., Farris, A. V., & Wright, M. (2012). From agents to continuous change via aesthetics: learning mechanics with visual agent-based computational modeling. *Technology, Knowledge and Learning, 17*(1-2), 23-42.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education, 26*(1), 3-20.

Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education, 120*(2018), 64-74.

Wing J.M (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Yükseltürk, E., & Altıok, S. (2015). Pre-Service information technologies teachers' views on computer programming teaching. *Amasya Education Journal, 4(*1), 50-65.

Appendix 1. List of articles (n=34) examined in the study (Ordered by authors)

1. Akcaoglu, M., & Koehler, M. J. (2014). Cognitive outcomes from the Game-Design and Learning (GDL) after-school program. Computers & Education, 75, 72-81.
2. Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to “real” programming. ACM Transactions on Computing Education (TOCE), 14(4), Article 25.
3. 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.
4. Bierman, K. L., Heinrichs, B. S., Welsh, J. A., Nix, R. L., & Gest, S. D. (2017). Enriching preschool classrooms and home visits with evidence-based programming: sustained benefits for low income children. Journal of Child Psychology and Psychiatry, 58(2), 129-137.
5. Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. Journal of Media Literacy Education, 4(2), 121-135.
6. Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). The status of teaching programming with scratch: A document review work. Journal of Instructional Technologies & Teacher Education, 4(3), 13-25.
7. Calao L.A., Moreno-León J., Correa H.E., Robles G. (2015) Developing mathematical thinking with Scratch. Lecture Notes in Computer Science, 9307 (2015), 17-27.
8. Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. Computers & Education, 58(1), 240-249.
9. Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO robotics kit in preschool classrooms. Computers in the Schools, 33(3), 169-186.

10. 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.
11. García, P. G., & De la Rosa, F. (2016). RoBlock-Web App for programming learning. International Journal of Emerging Technologies in Learning, 11(12). 45-53.
12. Grout, V., & Houlden, N. (2014). Taking computer science and programming into schools: The Glyndŵr/BCS Turing project. Procedia-Social and Behavioral Sciences, 141, 680-685.
13. Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. Computers in Human Behavior, 52, 200-210.
14. Kalelioglu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: a discussion from learners' perspective. Informatics in Education, 13(1), 33-50.
15. Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. Early Childhood Education Journal, 41(4), 245-255.
16. Keçeci, G., Alan, B., & Zengin, F. K. (2016). Educational computer games assisted learning coding attitude scale: Validity and reliability study. Education Sciences, 11(4), 184-194.
17. 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.
18. Kölling, M. (2010). The greenfoot programming environment. ACM Transactions on Computing Education (TOCE), 10(4), Article 14.
19. Kukul, V., Gökçearslan, Ş., & Günbatar, M. S. (2017). Computer programming self-efficacy scale (cpses) for secondary school students: Development, validation and reliability. Educational Technology Theory and Practice, 7(1), 158-179.
20. Küçük, S., & Şişman, B. (2017). Instructor experiences in one-to-one robotics instruction. Elementary Education Online, 16(1), 312-325.
21. Lau, W. W., & Yuen, A. H. (2009). Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. British Journal of Educational Technology, 40(4), 696-712.
22. Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K-8 curriculum. ACM Inroads, 5(4), 64-71.
23. Lin, J. M. C., & Liu, S. F. (2012). An investigation into parent-child collaboration in learning computer programming. Journal of Educational Technology & Society, 15(1), 162.
24. Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. Computers in Human Behavior, 41, 51-61.
25. Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. ACM Transactions on Computing Education (TOCE), 10(4), Article 16.
26. Miller, P. (2009). Learning with a missing sense: what can we learn from the interaction of a deaf child with a turtle?. American annals of the deaf, 154(1), 71-82.
27. Numanoğlu, M., & Keser, H. (2017). Robot usage in programming teaching – Mbot example. Bartın University Journal of Faculty of Education, 6(2), 497-515.
28. Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study. International Journal of Mobile Learning and Organisation, 10(3), 187-202.
29. Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2016). Constructing the ScratchJr programming language in the early childhood classroom. International Journal of Technology and Design Education, 26(4), 489-504.
30. Rogozhkina, I., & Kushnirenko, A. (2011). PiktoMir: teaching programming concepts to preschoolers with a new tutorial environment. Procedia-Social and Behavioral Sciences, 28, 601-605.
31. 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.
32. Sengupta, P., Farris, A. V., & Wright, M. (2012). From agents to continuous change via aesthetics: learning mechanics with visual agent-based computational modeling. Technology, Knowledge and Learning, 17(1-2), 23-42.
33. Shim, J., Kwon, D., & Lee, W. (2017). The effects of a robot game environment on computer programming education for elementary school students. IEEE Transactions on Education, 60(2), 164-172.

34. Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. International Journal of Technology and Design Education, 26(1), 3-20.

Appendix 2. Article classification form about coding learning at pre-university level

A – ARTICLE IDENTITY

1. Title: ……………………………………………………………………
2. Journal Name: ………………………………………………………………………
3. Origin of the Journal: Domestic ( ) Foreign ( )
4. Year of Publication: ……
5. Authors: Turkish ( ) Foreign ( ) Mixed ( )
6. Language: Turkish ( ) English ( )
7. Index: SCI,SCI exp,AHCI ( ) SSCI ( ) ESCI ( ) ERIC ( ) EBSCO ( ) Other ( ) Not indexed ( )

B – TOOL

1. Scratch ( )	2. Codemonkey ( )	3. Code.org ( )	
4. Robotic ( )	5. Presenting a tool ( )	6. Developing an environment ( )	7. Other ………………………

C – METHOD

1. Literature review ( )
2. Quantitative ( ) → Quasi-experimental ( ) Non-quasi-experimental ( )
3. Qualitative ( ) 4. Mixed ( ) 5. Developing an environment ( ) 6. Presenting an environment ( )

D – DATA COLLECTION TOOL No Data Collected ( )

1. Observation	( )	2. Interview or focus group interview	( )
3. Achievement tests	( )	4. Attitude, perception, personality, or skills tests	( )
5. Survey/scale	( )	6. Document ( ) 7. Other	( )

E – SAMPLE No Data Collected ( )

Sample Level (Grade)		**Sample Size**		
1. Elementary education (1-5)	( )	1. 1-10	( )	
2. Secondary education (6-8)	( )	2. 11-50	( )	
3. High school (9-12)	( )	3. 51-100	( )	
		4. Over 100	( )	

F - DATA ANALYSIS METHOD Literature review ( ) Not provided ( )

1. QUANTITATIVE DATA ANALYSES				**2. QUALITATIVE DATA ANALYSES**	
A.) Descriptive ()		B. Predictive ()		A. Qualitative ()	
1. Frequency/percentage/chart	( )	1. Correlation	( )	1. Content analysis	( )
2. Mean/standard deviation	( )	2. t-test	( )	2. Descriptive analysis	( )
3. Graphical representation	( )	3. ANOVA/ANCOVA	( )	3. Other ………………………	( )
4. Others…………………………	( )	4. MANOVA/MANCOVA	( )		
		5. Factor analysis	( )		
		6. Regression	( )		
		7. Non-parametric tests	( )		
		8. Other …………………………	( )		

G- PURPOSE / RESULT

Purpose(s):

Result(s):