Journal of Computer and Education Research
April 2021 Volume 9 Issue 17
http://dergipark.org.tr/jcer
(ISSN:2148-2896)

Review Article

Research Implications for Computer Science Education based on Darmstadt Model

Yasemin GÜLBAHAR[1], Filiz KALELİOĞLU*[2]

[1] *Ankara University, Ankara, Turkey, gulbahar@ankara.edu.tr*

[2] *Başkent University, Ankara, Turkey, filizk@baskent.edu.tr*

* *Corresponding Author: filizk@baskent.edu.tr*

Article Info

Received: *19 October 2020*
Accepted: *27 December 2020*

Keywords: Computer science education, research, systematic literature review

10.18009/jcer.806875

Publication Language: English

Abstract

The purpose of the current study is to examine published studies in computer science education in a systematic way, and to present a history of the research and new research trends in this area. This research study reports the findings of the systematic literature review according to the educational relevant areas dimension of the Darmstadt Model. The procedures of systematic text analysis were performed as a qualitative content analysis. Prior to the systematic text analysis, the primary term 'computer science education and K-12' was searched for along with data in the abstract, title and keyword section for publications between 2013 and 2018 in the databases and digital libraries of Academic Search Complete, Business Source Complete, Eric, Science Direct, and the IEEE Digital Library. A total of 87 articles formed the sample of the study. Although the current study was limited to the stated journal articles, it provides insight to the field by shedding light on important issues relevant to future research studies.

To cite this article: Gülbahar, Y. & Kalelioğlu, F. (2021). Research implications for computer science education based on darmstadt model. *Journal of Computer and Education Research, 9* (17), 39-61. DOI: 10.18009/jcer.806875

Introduction

In recent reports, there has been a call for changes in education based on a vision that focuses on future jobs and expected work-related skills. In the next five years, skills gaps continue to be high as new job opportunities require new skills (World Economic Forum, 2020), where computer science comprises a significant proportion of this insight. Among the skills required, cognitive abilities, content, and process skills play an important role. Cognitive abilities include cognitive flexibility, creativity, logical reasoning, complex problem solving, and mathematical reasoning (World Economic Forum, 2016; Özbey & Köyceğiz-Gözeler, 2020) that are either accomplished or can be supported by computer science education. Moreover, among the skills are computer literacy (sometimes referred to as information and communication technologies), and among the process skills are critical thinking, logical thinking, and reasoning performance, which again can be taught through computer science concepts.

Owing to these facts, computer science education has gained significant importance and many countries are aligning their curriculums to fulfil these needs in different ways. The starting age for students in computer science teaching has already been lowered to either kindergarten or primary schools in many countries (Bocconi, Chioccariello, Dettori, Ferrari, & Engelhardt, 2016). Provided as compulsory or elective, computer science courses aim to deliver computer science concepts that are generally aligned with computer literacy. According to the properties of the target group, appropriate pedagogies like code reading, code tracing, sub-goal modelling, code annotation, and live coding; and instructional approaches like game-based, unplugged activities, physical programming, block-based or text-based programming can be considered alternatives to teaching through different tools and techniques like toys, computers, and robotics. In such a rich and varied environment, selecting the best approach for the target group can be difficult and any selection may not provide the best intended learning outcome. This is the point where reliance should be placed upon empirical evidence, guided through the results of published empirical research.

Although many research studies have been conducted over the decades in the computer sciences, there have been few studies that have focused on K-12 computer science education. Future research should be planned in an effective way so as to guide this process in the most efficient manner in order to avoid loss of time and effort. Therefore, the purpose of the current study is to provide insight for future researchers of computer science education based on a systematic literature review.

Computer Science Education in K-12

Having a research history of no more than a decade, discussions and innovations about computer science education at the K-12 level have gained attention after reports such as ‘The New Educational Imperative’ (Stephenson, Gal-Ezer, Haberman, & Verno, 2005) and ‘Running on Empty’ (Wilson, Sudol, Stephenson, & Stehlik, 2010) by the Computer Science Teachers Association (CSTA), and have rapidly accelerated following the 2012 publication of The Royal Society’s ‘Shutdown or Restart’ report (The Royal Society, 2012). The emergence of computational thinking as a concept by Wing (2006) also provided strong support to the phenomenon. Curriculum standards, curricula, differentiated implementations at various levels, and target groups have since seen a continual change that has brought about significant challenges during the process.

The recent reports of the State of Computer Science Education Policy and Implementation (Code Advocacy Coalition, 2018), and by Blikstein (2018) about the state of the phenomenon, reported the findings of various implementations. The reports revealed a growing interest and success of these implementations by prioritising research on the teaching, learning, and assessment of computer science education. Blikstein (2018) specifically mentioned the need for ongoing and thorough research in order to facilitate more successful implementations in the areas of computer science concepts, programming tools design and experience, tools for formal learning environments, and other forms/paradigms of programming, as well as in the arena of the arts and creative computing.

Blikstein (2018) also recommended improvement of the equitable participation of all students in computer science education. However, although there have been numerous efforts and initiatives like 'CS for All' in the US and 'Informatics for All' in Europe, the expected spread of access and learning outcomes are far from being observed since there is a highly variable level of effort and achievement across different countries and education autonomous regions (Gretter, Yadav, Sands, & Hambrusch, 2019). Diverse and replicated research studies lead to consistent results, and may thereby reveal the best approaches which could help fill the gap between implementations at the national level and the pedagogical target groups.

Hubwieser, Armoni, and Giannakos (2015) postulated certain research questions that they considered both important and relevant to many different contexts, cases, and countries. These questions specifically addressed the topics of alignment of CS competencies, content and learning outcomes, didactical approaches, teaching methods, instructional media and materials, programming environments and languages with the target age group and school context. Furthermore, they structured and deepened their questioning by applying the Darmstadt Model (Hubwieser, 2013; Hubwieser et al., 2011) (see Figure 1) and posing even more questions.

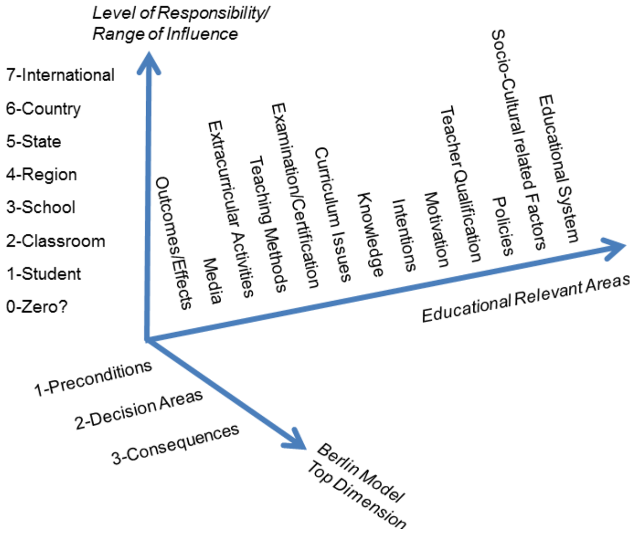

Figure 1. Darmstadt model (Hubwieser, 2013; Hubwieser et al., 2011)

The Darmstadt Model is a three-dimensional model having educational relevant areas as one dimension, level of responsibility/range of influence as a second dimension, and the Berlin Model Top Dimension as a third dimension. Educational relevant areas are focused on the following 13 components:

1. Educational System: Organisational Aspects of Subject, Enrolment, and School type;
2. Sociocultural-Related Factors: History of ICT and Informatics in Schools, Age, Gender, Social and Immigration Background, Family Socialisation, Public Opinion, and Techno-economic Development;
3. Policies: Research and Funding Policies, Education Policies, and Quality Management;
4. Teacher Qualification: Teacher Education, and Professional Experience;
5. Motivation: Students and Teachers;
6. Intentions: Learning Objectives, Competencies, and Standards;
7. Knowledge: CS and ICT;
8. Curriculum Issues;
9. Examination/Certification;
10. Teaching Methods: CSE and General Education;
11. Extracurricular Activities: Contest;

12. Media: Technical Infrastructure, Textbooks, Tools, Didactical Software, Visualisation Software, Unplugged Media, and Haptic Media; and,
13. Research: Outcomes and Effects.

Hence, the current research study not only reports findings of a systematic literature review, but also interprets the findings according to the educational relevant areas dimension of the Darmstadt Model. Hence, the research questions leading the current study are:

- What are the demographic characteristics (number of authors, keywords, country) of the selected articles?
- How did the studies vary in terms of their purpose and research design?
- What were the data collection methods and data analysis approaches in the studies?
- How did the studies vary in terms of their major findings?

Method

The purpose of the current study was to examine published studies in the area of computer science education in a systematic way, and to present a history of the research and also to identify new research trends in this area. The procedures of systematic text analysis were performed as a qualitative content analysis. Prior to the systematic text analysis, the primary search term 'computer science education and K-12' was entered along with the data in the abstract, title and keyword section, for publications between 2013 and 2018 in databases and digital libraries of Academic Search Complete, Business Source Complete, Education Resources Information Center (ERIC), Science Direct, and the IEEE Digital Library.

Sample

Initially, 152 articles were accessed. Intentionally, and parallel with the aim of the current study, articles which did not mention Computer Science Education conceptually, or did not include research in this field, were removed because they were considered to be not suited to the purposes of the current study. The remaining 87 articles formed the sample of this study.

Data Analysis

The articles were divided equally between the two researchers. Then the researchers individually qualitatively analysed the articles according to predefined criteria. The predefined criteria were: title of the article, year of publication, number of authors, keywords, country of study, purpose of the study, variables, research design/type, target group, sample size/data size, data sources, data collection method, data analysis methods and techniques, and major findings.

During this process, the researchers met weekly in order to discuss their findings and progress. According to the criteria, the articles were coded by each researcher and frequency tables were formed. The tables and codes of each researcher were rechecked and combined into a single file in order to obtain a general picture of the combined results from the two researchers.

In the following section, the results of the inductive analysis are presented according to the research questions. Following detailed document analysis, some parts of the analysed data are transformed into numerical values and illustrated through the use of graphics and word clouds.

Findings

Numbers of Articles per Journal

When the studies were examined in terms of number of articles in each journal, it was seen that 29 were published in the Journal of Educational Computing Research, 26 in Informatics in Education, 13 in Computer Science Education and 19 in different journals (see Table 1). A total of 87 articles were systematically analysed within the scope of this research.

Table 1. Number of articles by journal name

Journal Name	Number of Articles
Journal of Educational Computing Research	29
Informatics in Education	26
Computer Science Education	13
Computers in the Schools	4
Computers & Education	2
Computers in Human Behavior	2
Education and Information Technologies	2
Computing in Science & Engineering	1
IEEE Transactions on Education	1

IEEE Transactions on Learning Technologies	1
Issues in Informing Science and Information Technology	1
Journal of Science Education and Technology	1
Journal of Research on Technology in Education	1
Plos ONE	1
Technical Services Quarterly	1
The International Journal of Information and Learning Technology	1
Total	87

Year of Publication

When the year of publication of the articles were examined; there were four articles published in 2013, 16 in 2014, nine in 2015, 11 in 2016, 16 in 2017, and 31 articles published in 2018 (see Figure 2).

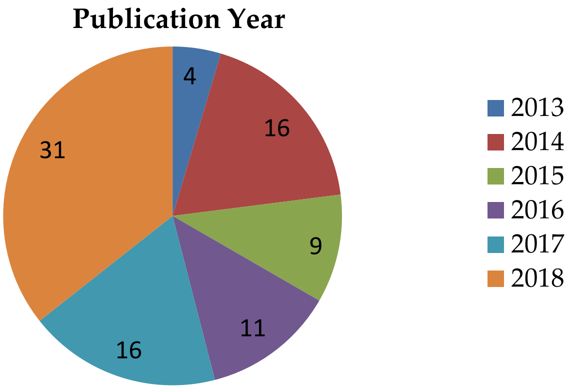

Figure 2. Publication years of the articles

Number of Authors

When the number of authors attributed to the articles was examined; it was found that the number of studies with a single author was 18, with two authors it was 25, with three authors it was 17, with four authors it was 17, with five authors it was four, and the number of studies with six or more authors was six. As can be deduced, half of the articles had only one or two authors, whereas only six articles were written by groups of six or more authors.

Most Used Keywords

In examining the keywords describing the articles; many keywords were found to have been used (see Figure 3). The most used keyword was Programming, which was used as a keyword in 44 of the studies, which was followed by Computational thinking in 21 articles. Other keywords used in the articles were Computing (n = 18), Teaching (n = 17), Computer science education (n = 14), Programming language (n = 10), K-12 (n = 8), STEM (n = 8), Assessment (n = 7), Robotics (n = 6), Scratch (n = 4), Augmented reality (n = 4), Teacher professional development (n = 3), Information technology (n = 3), and Teaching and learning strategies (n = 3).

Information Technology
Programming Language
Teaching and learning strategies
K-12
STEM
Computing
Scratch
Teacher professional development
Programming
Computational Thinking
Assessment
Computer Science Education
Robotics
Teaching
Augmented Reality

Figure 3. Most used keywords

Country of Study

In examining the country in which each of the studies were conducted, it was found that while many studies indicated the country of study, some studies did not include this information. Of the studies that included the country of study (see Figure 4), most were conducted in the USA (n = 21), followed by Turkey (n = 7). Other countries where studies were performed were Taiwan (n = 5), Brazil (n = 5), Greece (n = 4), Germany (n = 4), China (n = 3), Slovenia (n = 2), Lithuania (n = 2), Slovakia (n = 2), and the UK (n = 2).

Greece
Finland
Argentina
Slovenia
Malta
Singapore
Czech Republic
Japan
Spain
Turkey
Israeli
Taiwan
Hong Kong
USA
Hungary
Germany
Malta
Korea
Brazil
Australia
China
Canada
Norway
UK
Greece
Serbia
Slovakia
Lithuania
Arabia
New Zealand
Taiwan
Korea
The Netherlands

Figure 4. Countries of study

Purpose of Study

Based on the Darmstadt Model, when the aims or purpose of the articles were classified, it could be seen that the general research objectives were mostly formed under four topics (see Figure 5) as; Sociocultural-Related Factors (n = 20), Outcomes/Effects (n = 19), Teaching methods (n = 19), and Media (n = 16). The following provides a brief explanation of each purpose of study.

Sociocultural-related factors: Factors affecting the students' learning.

Outcomes/effects: The effects of methodologies, tools and media on computer science education and on thinking skills.

Teaching methods: Techniques and teaching methods in computer science education such as game-based and problem-based instruction, plus virtual reality applications.

Media: Mostly focused on the design and development of games, software or tools used in programming instruction.

Curriculum issues: K-12 level curriculum proposals in computer science education, physical programming course design, or course design at the university level.

Intentions: Standards or competencies expected in computer science education.

Motivation: Teacher and student attitudes and their learning motivations.

Knowledge: Learning the concepts of computer science and ICTs.

Teacher qualification: Teacher education and development for computer science education.

Extracurricular activities: Competitions and contests, and their effects.

Educational system: Institutional structure studies mostly at the macro level.

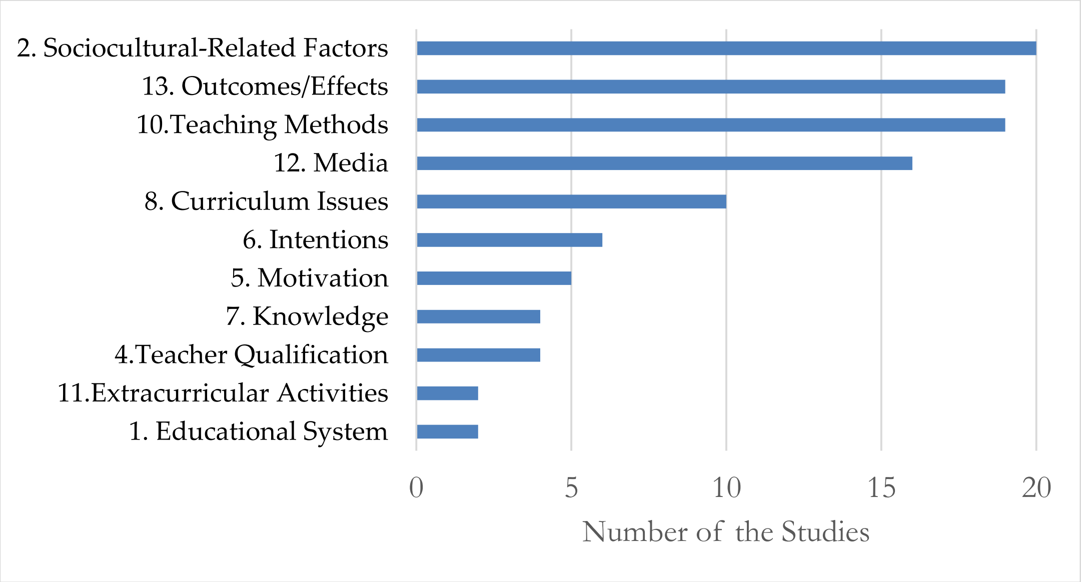

Figure 5. Grouping of research studies by purpose of study according to the dimensions of Darmstadt model

Variables

In considering the variables examined in the studies, information about the variables was not clearly presented in most of the articles. However, from the available information, it was seen that the most researched variables were Prior grade in programming courses (n = 6), Types of tasks that students work on in research (n = 6), Perception (n = 6), Self-efficacy for learning computers or programming (n = 5), Attitudes towards learning programming or computer science (n = 4), Achievement in different courses (n = 4), Computational thinking (n = 4), Interest in CS (n = 4) and, Learning Performance (n = 3).

Research Design/Type

When the studies were investigated in terms of their research design, it was seen that most relied upon quantitative measures, where Experimental design was the most preferred, rather than Qualitative measures. Survey was the second most preferred approach, followed by Case study, and Mixed methods design (see Figure 6). Although there was some overlapping of concepts in the classification such as Experimental design already being a Quantitative approach, the reason for this treatment is that the concepts are presented as mentioned by the authors of the studies.

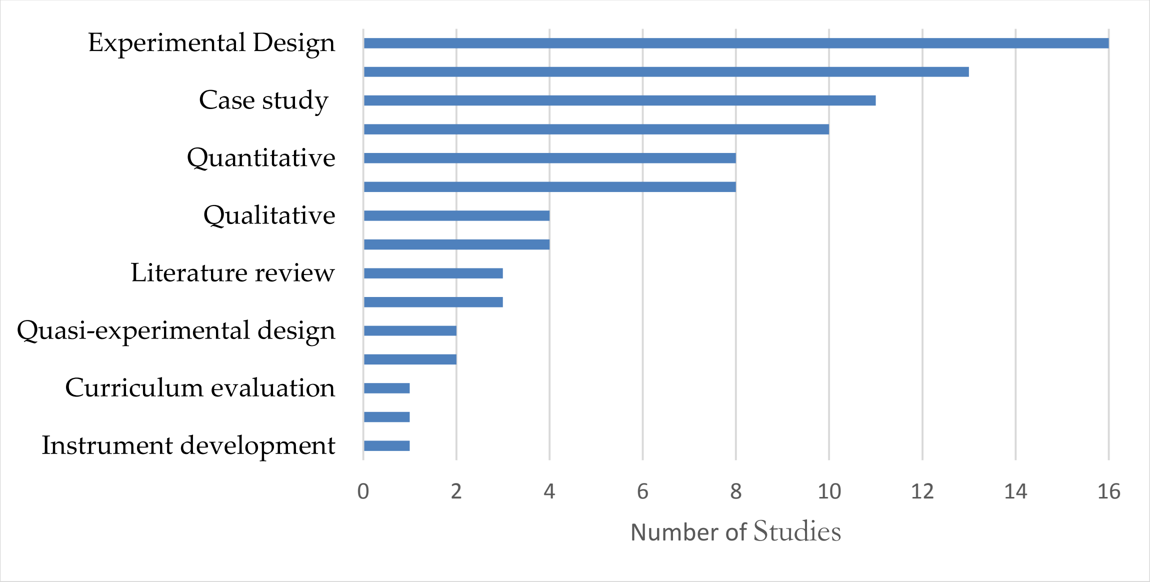

Figure 6. Grouping of research studies by research design

Target Group

When the studies were explored in terms of the research target group, it was noted that higher education students, especially undergraduates, were considerably more prevalent than other groups (see Table 2), which was followed by studies that addressed different K-12 age groups, and studies on 5th-8th Grade groups. There were three separate studies observed with one for kindergarten, one for professors, and one for parents as their target group of study, which are not shown in Table 2.

Table 2. Grouping of research studies by target group

Target Group	Number of Studies
1st to 4th Grade	4
5th to 8th Grade	13
9th to 12th Grade	4
K-12 (mixed grades)	15
K-12 + Higher Education	2
Higher Education	33
Teachers	4

Sample Size

It is important for any research to select a valid representative sample of the population in order to reach generalisable conclusions. However, almost half of the studies were conducted with 100 or fewer participants (see Figure 7), which may constitute a general threat in terms of the validity and generalisability of the research findings. However, several studies reached a wider target population with larger samples.

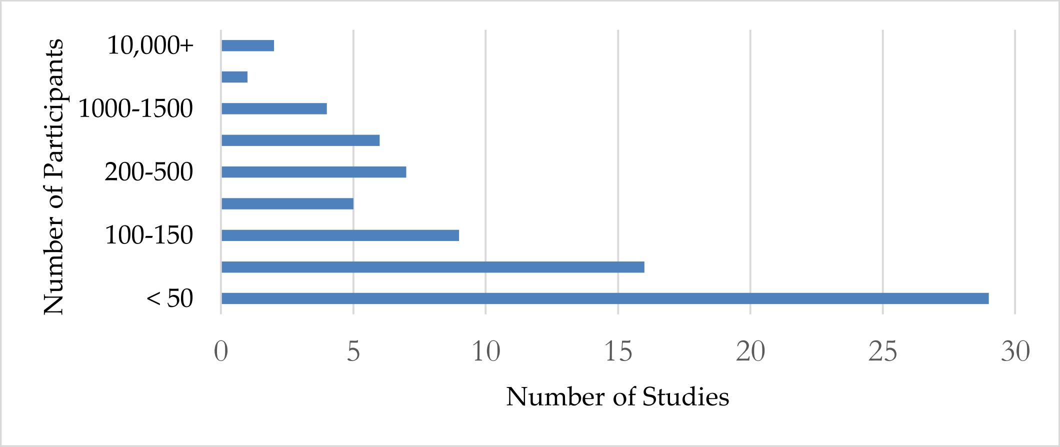

Figure 7. Grouping of research studies by sample size

Data Sources

For any research study, the source of data is important in terms of many aspects such as triangulation, participant honesty, accessibility, and being free from bias. For most of the research carried out on computer science education, the students formed the primary data source (69%). Of the other studies, teachers were the data source for 10% of the articles, with documentation used in 8% of studies, and research databases in 5% of cases. Only 5% of the studies used more than one type of data source, which raises questions as to the reliability of the findings. Additionally, 3% of the studies had professors or parents as data sources.

Data Collection Methods

When the studies were categorised in terms of their data collection methods, a variety were observed (see Figure 8). Survey, Questionnaire, and Achievement tests were the most commonly used data collection instruments for quantitative measures. In terms of qualitative measures, Interviews, Document analysis, Recordings, and Observations were also used for the purposes of data collection. There were also single studies that utilised PISA results, Usability tests, and Attitude scales as their data collection instruments, although these are not included in Figure 8. Realising that most of the studies focused on the academic performance of students, the common use of instruments to grade and explore the learning outcomes is not unsurprising.

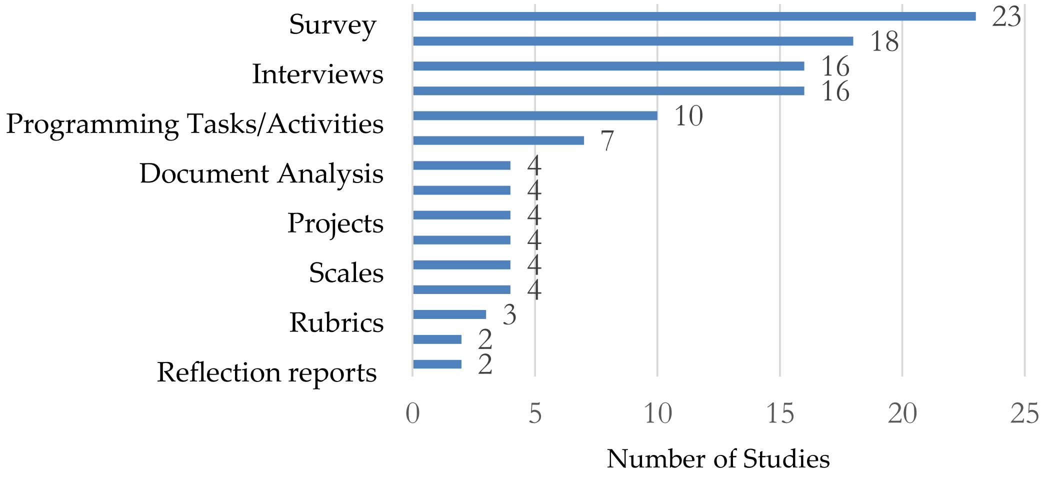

Figure 8. Grouping of research studies by data collection methods

Data Analysis Methods

Mostly relying on Descriptive analysis, a variety of statistical data analysis methods were observed in the examined research studies (see Figure 9). Methods used in only one of the studies were nonparametric tests, Multi-dimensional scaling and cluster analysis, Path model, Principal component analysis, Meta-analysis, and Construct validity analysis. On the other hand, although not a significantly large proportion of the qualitative studies, Coding schemes and Curriculum evaluation were also among the reported data analysis methods.

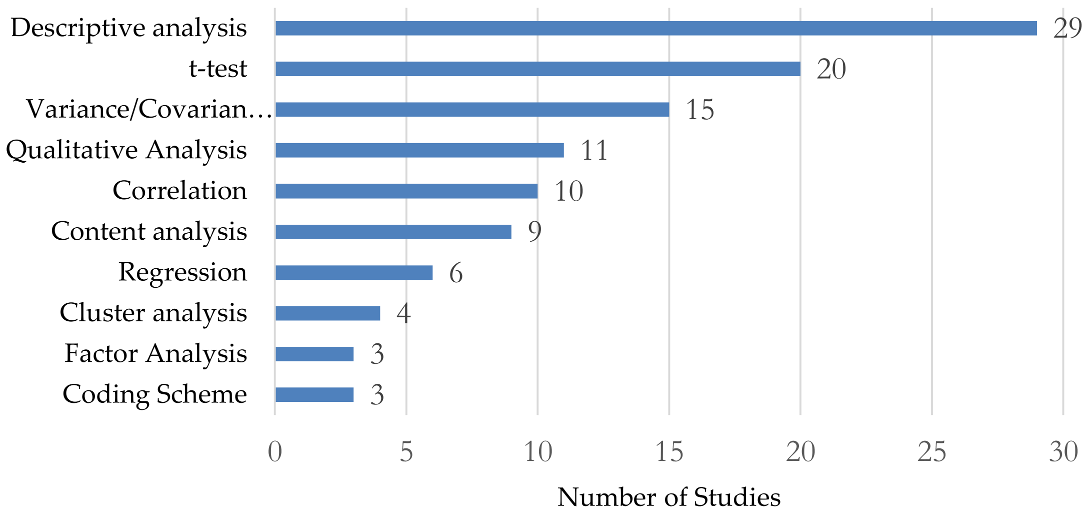

Figure 9. Grouping of research studies by data analysis method

Major Findings

The major findings of the research studies were classified according to the Darmstadt Model, based on their contribution to the field (see Figure 10).

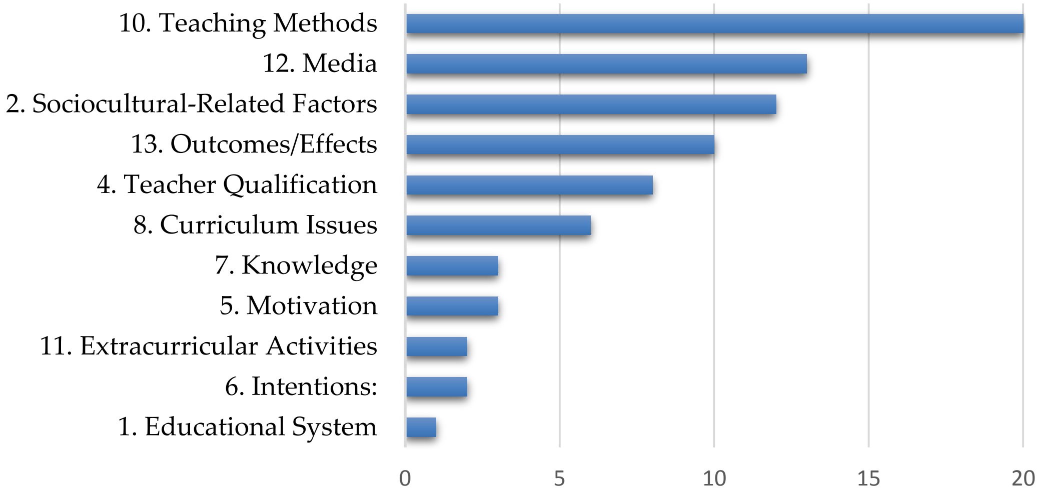

Figure 10. Grouping of research studies by major findings

The category that took most researchers attention was 'Teaching Methods'. All of the findings revealed the positive effect of different teaching methods on learning programming. Use of paper programming strategy and unplugged CT activity, situated and spiral learning-by-design approach, 'constructionist-oriented' instructional process, 'deep' study approach, intervention based on metacognition, collaborative robotics projects, sub-goal labelled worked example, arcade and educational games, model for concept-based learning, integrated designs for the cultivation of Computational Thinking, and the use of concept maps as an educational tool were reported to be either effective or caused significantly better learning of programming. In another study, a variety of pedagogical strategies were also recommended by teachers based on their individual experience as: unplugged type activities, contextualisation of tasks, collaborative learning, developing computational thinking, and scaffolding programming tasks. It can be seen that numerous ways were put forward to accomplish the task of teaching programming with some premising research findings.

The second most covered area of findings related to the category of 'Media', which generally investigated the effect of using different software on learning outcomes. Block-based programming (mostly Scratch and Alice), text-based programming experience, learning to program robots with developmentally appropriate tools, and the use of animations and algorithm visualisation tools were found to be mostly effective in reaching expected learning outcomes. However, one study reported that 'programming in Scratch platform did not cause any significant differences in the problem solving skills of primary school students (Kalelioglu & Gulbahar, 2014, p. 33), whereas another study reported on the low usage of various teaching aids in programming classes as an obstacle to learning.

Studies that focused on 'Sociocultural-Related Factors' were observed to have dealt with variables such as like skills prior to and after having learnt programming, learning styles and knowledge map construction (Shaw, 2017), different generation and non-immigrant students, the performance of problem-solving ability, having prior programming experience (Kim, 2018; Veerasamy, D'Souza, Lindén, & Laakso, 2018), students' self-explanation quality, number of code edits (Liu, Zhi, Hicks, & Barnes, 2017), computational thinking, spatial and reasoning ability where all variables were found to have some effect on either the learning outcomes of learner characteristics except learning styles, and knowledge map construction. Two studies reported no gender difference in terms of academic achievement in computer science, whereas problem-solving ability was associated positively with performance at the concrete operational stage in one study (Kožuh, Krajnc, Hadjileontiadis, & Debevc, 2018), and significantly correlated with students' self-explanation quality, number of code edits, and prior programming experience in another (Liu et al., 2017). Yet, another study reported significant correlations of at least moderate intensity between computational thinking with spatial ability, reasoning ability, and problem-solving ability.

The 'Outcomes/Effects' category showed diverse research interests among the studies. Effects of persistence levels on self-efficacy beliefs (Lin, 2016), level of knowledge on specific subjects, increasing accessibility to CS resources, a measurement model, a model for building CS content as a scaffolder for higher-level learning in transdisciplinary settings, providing teachers with more computer-based training (Coleman, Gibson, Cotten, Howell-Moroney, & Stringer, 2016), and early access to computer science lessons were among the topics explored. There were also some research recommendations put forward based on literature reviews.

For the aspect of 'Teacher Qualification', several studies reported on the challenges that teachers face such as: isolation (Yadav, Gretter, Hambrusch, & Sands, 2016), lack of adequate computer science background (Yadav et al., 2016), limited professional development resources (Yadav et al., 2016), and limited knowledge of and experience with computer science. Menekse (2015) cited not only limited collaboration between educational organisations to develop computer science teachers' professional development, but also limited duration and lack of clear focus on discipline-specific pedagogical content knowledge for existing professional development programmes.

Under the 'Curriculum Issues' category, researchers efforts were proven to be successful on developing, implementing, and evaluating various curriculum for different target groups like teachers (Kucuk & Sisman, 2018) and students (Kynigos & Grizioti, 2018) based on different approaches like robotics and 3D spaces (Kucuk & Sisman, 2018; Kynigos & Grizioti, 2018). One study stated that "Despite the recent revived interest in programming for K-12, little studies have been conducted to inform the researchers and educators on implementing suitable curriculum for the group of students" (Lye & Koh, 2014, pp. 59-60).

Having three studies under the "Knowledge" category, the authors of the researchers were noted as searching for defining constructs in order to reveal perceptions of parents toward programming (Kong, Li, & Kwok, 2018), dimensions for assessment of computational thinking (Zhong, Wang, Chen, & Li, 2016), and students use of computational thinking concepts in a Story-Writing-Coding context (Price & Price-Mohr, 2018). Hence, all of the studies were types of assessment of cognitive knowledge and abilities.

Research under the "Motivation" category revealed facts about students in STEM fields, indicators of student engagement (Benotti, Martinez, & Schapachnik, 2018), and the importance of taking more units in the computer science subject in order to increase student motivation (Lee, 2015). Thus, considering which factors can affect motivation levels. There were also two studies grouped under "Extracurricular Activities" based on challenges for solving tasks, and two studies under the "Intentions" category that focused on competence areas for computer science education (Zendler, Klaudt, & Seitz, 2014). There was also one article under the category of "Educational System", where the authors (Aleksić & Ivanović, 2016) investigated the use of programming languages at different universities. They concluded that "Universities from Central and Eastern-European countries mainly based their study programmes on teaching C and C++ programming languages, while programmes of the Scandinavian universities were mainly based on Java" (p. 177). The researcher's also added that while most of programmes highly support object-oriented paradigm of programming, introductory programming subjects were mainly based on imperative paradigm. Additionally, there were no research studies conducted that were categorised as regarding 'Policies' or 'Examination/Certification'.

Discussion

When the studies were examined, most of the research articles between 2013 and 2018 were published in the Journal of Educational Computing Research and Informatics in Education. In terms of publication years, while the distribution is fairly equally distributed, 2018 saw almost twice as many studies published. The reason for this may be the increased interest in the area due to curricular studies in computer science education and changes to national education policies (Bocconi et al., 2016).

Although computer science education and K-12 keywords were systematically searched; the keywords entered for the articles gives an idea of the research topics published for computer science education. Whilst a very comprehensive field, it was observed that the keywords used in computer science education research focused on programming and computational-thinking skills; which reflects the nature of computer science education research at the K-12 level. It can also be said that at the K-12 level; teaching methods, teaching tools and evaluation methods constitute the most researched topics.

As to which countries are undertaking research on computer science education, conclusions can be drawn based on examining the countries where the published research has taken place. It was seen that studies were conducted in America, Europe, and Asia. This result is supported by the Developing Computational Thinking in Compulsory Education report by Bocconi et al. (2016), in which it was stated that since 2012, many countries have carried out curriculum developments in order to integrate computer science education and computational thinking into compulsory education. Therefore, developments and changes in national education are naturally reflected in the published research.

It is possible to understand what kind research studies are conducted from examining the research purposes of the published studies. When the studies were examined in general, it can be seen that the research mostly consisted of studies aimed at revealing cause and effects. Moreover, this situation is fully supported by pedagogical issues. By considering the purposes of the published studies, computer science education research topics are can be listed as teaching methods, programming environment and context, and hardware and software that can impact student-related variables. On the other hand, there are also some notable issues with regards to areas of very little published research. For instance, the lack of macro-level studies such as on policy and education systems, curricula, and extracurricular activities are noteworthy examples. In addition, the current study found no research that had focused on examination and certification.

It can be concluded that researchers are generally more interested in the effects of different teaching methods and media for teaching programming, as well as in student characteristics. As educators, having an effective learning environment supported by appropriate pedagogical approaches is the most significant aspect of the teaching-learning process. Hence, reaching such a finding is not unsurprising. However, having diverse teaching methods where all of them have a large or small effect but are somehow successful is quite perplexing. This critique can also be applied to the findings about media; that whatever media may be used, students are still likely to learn from it.

From decades of research on the effect of technology on learning, it is known that technology-rich or computer-based environments may or may not enhance learning, but generally a negative impact would not be expected. Thus, rather than investigating the effect of a certain piece of software, learning environment, or tool, it would perhaps be more useful to explore the appropriateness of these media according to learner age groups from a cognitive and constructive point of view.

One other issue may be the order of the software and tools used from kindergarten through to high school; starting with block-based then switching to text-based, or starting with robots and then switching to text-based etc. More empirical evidence is needed of which tool should be used at what age, and the same goes for pedagogy too. Which teaching methods should be used for specific age groups so as to increase academic performance? Studies addressing these issues should be conducted with participants of different age groups. Among those investigated in the current research, only 17 studies were observed that undertook a comparison of different age groups.

Studies that attempt to define and assess the concepts of the field should be continued until an agreed upon framework has been reached. Theories as didactical approaches are quite important, since they form a base for the academic research work of the future. Curriculum issues are also important in order to frame the knowledgebase for students of different ages. Hence, studies should be conducted by targeting different age groups. Personal constructs are what carries graduated students on to a computer science-related profession; therefore, studies that reveal the interest, cognitive, and problem-solving abilities of students are needed in order to better support their decisions about their future career.

Research studies to fulfil the premise of "Policies", "Examination/Certification", and "Extracurricular Activities" should be reaching thousands of participants in order to properly suggest a policy or to improve the effectiveness of being "qualified" by way of examination. However, having a strong voice in this area is not easy, so it is unsurprising that no more than a few studies are published; or none, as in the case of the current study's sample. This situation does not imply any lack of necessity for such studies, but underlines the importance of handling issues of a wider audience and reaching sound conclusions. The

Darmstadt model also has a dimension called "range of influence". Findings of a single study may not be true for a different content or culture, which makes research questionable as to its international validity and reliability. In order to reach sound conclusions at the policy level, collaborative research studies should be planned first at the national level and then internationally.

When the findings of the current study are investigated based on the Darmstadt Model, it can be concluded that there were no studies conducted at the school level or above in terms of range of influence. Almost all of the studies assessed were in the range of classroom or student influence. Although not directly related with the scope of this research, it is worth noting that some dimensions of the Darmstadt Model can be extended to a broader scope in order to address issues of personal constructs, assessment, as well as didactical aspects. The category of 'Motivation' can be extended to 'Personal Constructs' in order to cover variables such as attitude, interest, prior knowledge, and self-regulation. The category 'Knowledge' can be extended to 'Theory and Knowledge' in order to cover studies about the definition and framework for emerging concepts and theories. Finally, the 'Examination/Certification' category can be extended to 'Assessment and Evaluation' in order to broaden the scope to the assessment and evaluation of issues such as academic performance, academic success, thinking ability, and cognitive aspects. Hence, the Darmstadt Model can be thought of as a framework to guide systematic literature reviews about the area of computer science education.

Conclusion

Although the current study was limited to 87 journal articles, it provides insight into the computer science education field by highlighting certain important issues for future research studies in the field. It is clear that there is a need to broaden the research about the teaching and learning of computer science at the K-12 level by also trying to improve the range of influence. Hence, encouraging teamwork rather than only as individuals may fulfil the premise of broadening the impact of future studies. Adding to the theory of knowledge, reaching proven competencies, standards, curriculums, teaching and assessment strategies, professional development approaches and instructional media and content is as important as making them valid and reliable by being culture and context independent for educational systems in the sociocultural context.

On the other hand, all research studies should be suggesting ideas for those who may benefit from the results; after all, that is one of the primary reasons why research is undertaken and published. All studies should be contributing to at least one stakeholder in the field with at least one of its findings. But is that true for all research studies? What if one develops a new curriculum and proved its effectiveness based on research and did not then subsequently share the curriculum itself? What if one implements a teaching strategy or instructional media and reports its success, but never reveals the details? How can such research studies benefit others or contribute to the field? If there is anything found to have a positive impact on the process or the product of study, then the details and its documentation should be publically shared so that replicated studies may be conducted, either to validate or to falsify the original study, and thereby strengthen the literature of the field. Hence, there should be a knowledgebase of curricula, competencies, learning outcomes, assessment scales and inventories, achievement tests and tasks etc. In the case where a researcher adds to this base rather than producing a new curriculum, inventory, or survey, the findings of research studies are more likely to merge in order to present a more valid and reliable picture. Although the role of systematic literature studies here serves a similar purpose, there should be a higher level of systematisation in order to distribute what is known from research studies to the wider academic world. Hence, organisations, conferences, and journals will be in a better position to provide researchers with future research topics in order to form a reliable knowledgebase on specific dimensions, and thereby prevent loss of time and effort in producing studies with little or no contribution to the field.

Research trends are changing according to technology, culture, and societal needs, which currently seems to be moving more towards computational thinking, teacher education and professional development, pedagogical aspects, gender and diversity issues, stem/steam approaches, as well as physical computing and robotics. However, social, economic, and cultural barriers surrounding computing should also form an important focus for research, with studies carried out in collaboration with an interdisciplinary approach at the international level that subsequently informs the wider academic community (Blikstein, 2018). "Only in this way can we achieve the hoped for scale and sustainability, and realise the ultimate vision of generations of researchers, practitioners, and policy makers that have been trying, for the last 50 years, to bring CS to all students" (Blinkstein, 2018, p. 35).

Acknowledgement

The data used in this study was confirmed by the researchers that it belongs to the years before 2020.

Authorship Contribution Statement

Yasemin Gülbahar: *Conceptualization, design of the work, literature search, data analysis, data interpretation, writing - review and editing.*

Filiz Kalelioğlu: *Literature review, data collection, data analysis, data interpretation, writing and manuscript revisions.*

References

Aleksić, V., & Ivanović, M. (2016). Introductory programming subject in European higher education. *Informatics in Education*, *15*(2), 163-182.

Benotti, L., Martinez, M. C., & Schapachnik, F. (2018). A tool for introducing computer science with automatic formative assessment. *IEEE Transactions on Learning Technologies*, *11*(2), 179-192.

Blikstein, P. (2018). *Pre-College computer science education: a survey of the field.* Mountain View, CA: Google LLC. Retrieved from https://goo.gl/gmS1Vm.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice (No. JRC104188).* Luxembourg: Publications Office of the European Union. Retrieved from http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf.

Code Advocacy Coalition. (2018). *2018 State of computer science education, policy and implementation*. Retrieved from https://code.org/files/2018_state_of_cs.pdf.

Coleman, L. O., Gibson, P., Cotten, S. R., Howell-Moroney, M., & Stringer, K. (2016). Integrating computing across the curriculum: The impact of internal barriers and training intensity on computer integration in the elementary school classroom. *Journal of Educational Computing Research*, *54*(2), 275-294.

Gretter, S., Yadav, A., Sands, P., & Hambrusch, S. (2019). Equitable learning environments in k-12 computing: teachers' views on barriers to diversity. *ACM Transactions in Computing Education*, *19*(3), 1-16.

Hubwieser, P. (2013). The Darmstadt model: A first step towards a research framework for computer science education in schools. In Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages. In I. Diethelm & R. T. Mittermeir (Eds.), *Proceeding of the 6th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (ISSEP'13)* (pp. 1-14). Berlin, Germany: Springer.

Hubwieser, P., Armoni, M., Brinda, T., Dagiene, V., Diethelm, I., Giannakos, M. N.,…Schubert, S. E. (2011). Computer science/informatics in secondary education. In L. Adams & J. J. Jurgens (Eds.), *Proceedings of the 16th Annual Conference Reports on Innovation and Technology in Computer Science Education—Working Group Reports* (pp. 19-38). New York, NY: ACM. doi:10.1145/2078856.2078859

Hubwieser, P., Armoni, M., & Giannakos, M. N. (2015). How to implement rigorous computer science education in K-12 schools? Some answers and many questions. *ACM Transactions in Computing Education, 15*(2), 1-12. DOI: http://dx.doi.org/10.1145/2729983.

Kalelioglu, F., & Gulbahar, 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.

Kim, S. (2018). ICT for children of immigrants: Indirect and total effects via self-efficacy on math performance. *Journal of Educational Computing Research*, *55*(8), 1168-1200.

Kong, S.-C., Li, R. K.-Y., & Kwok, R. C.-W. (2018). Measuring parents' perceptions of programming education in p-12 schools: scale development and validation. *Journal of Educational Computing Research*, *57*(5), 1260-1280.

Kožuh, I., Krajnc, R., Hadjileontiadis, L. J., & Debevc, M. (2018). Assessment of problem solving ability in novice programmers. *PloS one*, *13*(9), 1-21.

Kucuk, S., & Sisman, B. (2018). Pre-Service teachers' experiences in learning robotics design and programming. *Informatics in Education*, *17*(2), 301-320.

Kynigos, C., & Grizioti, M. (2018). Programming approaches to computational thinking: integrating turtle geometry, dynamic manipulation and 3d space. *Informatics in Education*, *17*(2), 321-340.

Lee, A. (2015). Determining the effects of computer science education at the secondary level on STEM major choices in postsecondary institutions in the United States. *Computers & Education*, *88*, 241-255.

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

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

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.

Menekse, M. (2015). Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. *Computer Science Education*, *25*(4), 325-350.

Özbey,S. & Köyceğiz-Gözeler,M. (2020). A study on the effect of the social skill education on the academic self respect and problem solving skills of the pre-school children. *International e-Journal of Educational Studies (IEJES), 4* (8), 176-189. DOI: 10.31458/iejes.727590.

Price, C. B., & Price-Mohr, R. M. (2018). An evaluation of primary school children coding using a text-based language (Java). *Computers in the Schools, 35*(4), 284-301.

Shaw, R. S. (2017). The learning performance of different knowledge map construction methods and learning styles moderation for programming language learning. *Journal of Educational Computing Research*, *56*(8), 1407-1429.

Stephenson, C., Gal-Ezer, J., Haberman, B., & Verno, A. (2005). *The new educational imperative: Improving high school computer science education. Using worldwide research and professional experience to improve U.S. Schools. Final Report of the CSTA Curriculum Improvement Task Force.* New York, NY: ACM CSTA. Retrieved from https://cse.sc.edu/~buell/References/StudentRecruiting/CSTA-WhitePaperNC.pdf.

The Royal Society. (2012). *Shutdown or Restart. The Way Forward for Computing in UK Schools*. Retrieved from https://royalsociety.org/-/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf

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

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). *Running on empty: The Failure to Teach K–12 Computer Science in the Digital Age.* Retrieved from https://runningonempty.acm.org/fullreport2.pdf.

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

World Economic Forum. (2016). *The future of jobs employment, skills and workforce strategy for the fourth industrial revolution. Global Challenge Insight Report.* Retrieved from http://www3.weforum.org/docs/WEF_Future_of_Jobs.pdf.

World Economic Forum. (2020). *The future of jobs report.* Retrieved from http://www3.weforum.org/docs/WEF_Future_of_Jobs_2020.pdf.

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education*, *26*(4), 235-254.

Zendler, A., Klaudt, D., & Seitz, C. (2014). Empirical determination of competence areas to computer science education. *Journal of Educational Computing Research*, *51*(1), 71-89.

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