© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1718009

Exploring Computational Thinking Integration in Mathematics: A Review

NRIAGU CHUKWUNONSO OGONNA[1], EGBUNIWE OBIAGELI NANCY[2], OKETAYO ABIMBOLA MUJIDAT[3], ARIH SARAH NKEMDILIM[4]
[1,2,3,4] National Mathematical Centre, Abuja. Nigeria.

Abstract - Computational Thinking (CT) encompasses cognitive processes that enable problem definition and solution through computational tools. CT is closely tied to mathematical foundations and it's recognized as a necessary skill in mathematics and science education. This paper presents a systematic literature review examining how mathematics education is influenced when students engage in CT-focused instructional activities. The reviewed studies primarily involved software-based learning tools, explored diverse mathematical topics and approaches to integrating computational thinking into mathematics classroom. The findings indicate that computational thinking activities were concentrated in secondary school and university education, driven by the use of CT to mitigate conceptual difficulties in mathematics. It not only enhances mathematical understanding but also fosters interdisciplinary problem-solving. Studies consistently validates the role of computational thinking makes learning more interactive and improve mathematics skills. Despite the challenges related to assessments, teacher training, and curriculum design. Strengthening these areas is essential for the sustainable integration of CT as a core feature in mathematics education.

Keywords: Computational Thinking, Mathematics Education, Problem-Solving Skills, Software Tools

I. INTRODUCTION

The integration of computational thinking into mathematics education has become a focus of research, reflecting the growing recognition that problem-solving in the digital age requires both mathematical reasoning and computational approaches. Early studies, such as Kafai et al. (1998), on how game design could serve as an interactive environment for nurturing mathematics among students and teachers. This area of research has expanded, with scholars emphasizing the importance of programming as a literacy for children and a foundation for interdisciplinary learning (Kafai and Burke, 2014; Moreno-León et al., 2016).

In Wing (2006), Computational Thinking (CT) refers to the cognitive processes of abstraction and decomposition that support problem solving using computational resources and algorithmic strategies. It is broadly defined as the ability to formulate problems and solutions in ways that can be executed by a computer and has been linked to mathematics through shared practices of abstraction, decomposition, and algorithmic reasoning (Kale et al., 2018; Lye and Koh, 2014). Studies have shown that programming environments and digital game-plays support learning of computational thinking, and mathematical concepts (Kazimoglu et al., 2012; Ke, 2014). Robotics and game-based learning further enhance students' self-efficacy, and computational skills, while simultaneously strengthening their mathematical reasoning (Leonard et al., 2016).

Research has argued that the similarity with mathematical skills is clear; abstraction through analogy, generalization, and decomposition are essential for mathematical problem solving (Polya 2004; Hemmendinger 2010). The interdisciplinary nature of computational thinking is evident in efforts to embed mathematics throughout computer science curricula (Krone et al., 2011) and to design integrated STEM learning environments that connect programming with physics and mathematics (Lin et al., 2019; Pruski and Friedman, 2014). Scholars such as Lambic (2011) and Olteanu (2022) highlight how programming software and computational approaches can make mathematical reasoning more practical, understandable and accessible. More recent work has examined interactions between computational thinking skills and heuristics in mathematical problem solving, underlining the potential for CT to enrich traditional mathematical methods (Lehmann, 2025; Polya, 2004).

At the primary and secondary school levels, computational thinking has been introduced through tools like Scratch, which support the learning, development of informatics concepts and mathematical sciences (Lewis and Shah, 2012; Nordby et al., 2024). These approaches illustrate how programming can serve as a bridge between abstract mathematical ideas and concrete applications, fostering interdisciplinary connections. As Kitchenham (2004) outlines, systematic reviews provide a rigorous framework for synthesizing such diverse findings, to enhance understanding of intersection of learning of computational thinking and mathematics in schools. Hood (2005) noted that achieving widespread fluency in information technology requires implementation in the education system. However, fluency means more than mastering techniques; it also involves organizing thinking to solve concrete problems.

When introducing CT activities in education, it is important to consider their impact on learning other subjects, competency is the ability with knowledge to act effectively in a situations. Prior work has examined relationships between mathematics and computer science in educational settings (Ke 2014; Krone, Sitaraman and Hallstrom 2011; Pioro 2006; Ralston 2005; Taylor, Harlow and Forret 2010). It is reasonable to expect that integrating CT activities could benefit mathematics teaching and learning.

Many CT activities linked to mathematical skills have been reported in recent years, with the need for a detailed analysis of learning and its results. This study presents a systematic literature review aimed at identifying how relationships between mathematics and computational thinking have been demonstrated through instructional activities.

This review was guided by the three research questions:

1. How are computational thinking and mathematics connected, and who are the target audiences?
2. Which computational thinking concepts and mathematical topics are addressed in these connections?
3. What research methods and procedures are employed to investigate the integration of computational thinking and mathematics?

Section 2 is the theoretical background of computational thinking and discusses its connections to mathematics. Section 3 describes the review method. Results are presented in Section 4 and discussed in Section 5. Section 6 provides conclusion.

II. THEORETICAL BACKGROUND

Creswell and Miller (2000) provided a methodological foundation by addressing the perennial challenge of validity in qualitative inquiry. It has become a benchmark for ensuring credibility in interpretive research, offering a framework that continues to guide contemporary studies in social sciences and education. Trochim and Donnelly (2006) synthesized both qualitative and quantitative approaches, underlining the importance of mixed methodological approach.

In Wing (2006); computational thinking articulated its significance as a universal skill to be on the same level as numeracy and literacy. It catalyzed a global movement to influence educational policy reforms to integrate computational thinking into the curricula. Papert's *Mindstorms* (1980) serves as a precursor to these debates, introducing constructionist learning through programming and demonstrating how children can develop powerful ideas by engaging directly with computational tools. His vision laid the foundation for subsequent explorations of digital tools for training.

Applied perspectives on computational thinking are evident in Hoji et al (2013) demonstrated how Octaves tool can be leveraged to teach forces and torques in vocational education. Their study exemplifies the practical integration of mathematics, engineering, programming, and the potential of computational tools to enhance conceptual understanding. Hood (2005) extend this conversation into the school context, advocating for fluency in information technology as a critical component of modern education. Their work aligns with broader initiatives to embed digital literacy into the curricula.

Polya (2004), originally from 1945 of "How to Solve It" remains an influential text in mathematical education, offers heuristic strategies that go beyond disciplinary and computational approaches to problem-solving. The literature also reflects enduring contributions to problem-solving and teacher proficiency. The need for teachers, educators to adapt to evolving demands in assessment and professional development in the twenty-first century. Pioro (2006) added an interdisciplinary dimension by exploring connections between mathematics and computer science, developments in research mathematics inform teaching practices and inspire innovative approaches to curriculum design.

2.1 Computational Thinking

(Wing, 2006) argued that computer science skills should be taught from the early age in schools, illustration of computational thinking on principles in computer science. Following this, several computing curricula were revised to include skills associated with computational thinking in developed countries. The importance of integrating computational and mathematical thinking into education can't be over-emphasized.

2.2 Computational Thinking and Mathematics Skills

Computing has been closely tied to mathematics. Early models like the Turing Machine and λ-calculus were developed to show the feasibility of automating calculations. Computing methods have links to natural sciences, engineering, and mathematics (Denning 2005). Experimental methods support heuristic algorithm design, engineering methods with software development, and mathematical deduction that provides the basis for proving correctness.

Although there is growing recognition that some computer science skills should be developed in primary education, studies show these skills are often taught together with other subjects such as biology, literature, arts, and mathematics (León, Robles and Román-González 2016). Since computational thinking is a practical approach to problem solving, it can be applied across the subjects. Demonstrating its benefits has been a focus of several studies and identifying of similar skills between computational thinking and mathematics is shown later in this review.

Studies such as (Barcelos and Silveira (2012, 2013)) show that CT and mathematics share similarities like logical reasoning and problem solving. Davenport et al. (2014) and Benton et al. (2017) demonstrated how programming and mathematics can be taught together, strengthening conceptual understanding that bridge the gaps in learning.

Practical applications of CT in mathematics classrooms often involve robotics and game-based learning. Robotics enhances collaboration in mathematics, as robotics learning activities improve CT skills in gender and different age groups (Ardito, et al 2014; Atmatzidou and Demetriadis 2016). Mobile games could enhance measurement estimation, blending CT with mathematical reasoning (Arroyo et al. 2017). These approaches illustrate how technology-driven activities make abstract mathematical concepts more accessible.

III. METHODOLOGY

We followed the guidelines of Kitchenham (2004) and Wohlin et al. (2012), which outlined three stages: planning, conducting, and reporting the review.

The initial search using ACM, IEEE, and Google Scholar found limited review with the same objective. The research questions were stated in Section 1. The review process specified the databases, search strings, inclusion and exclusion criteria. Four databases were searched: ACM, IEEExplore, SpringerLink, and ScienceDirect.

Two inclusion criteria were applied:

1. The study indicates a relationship between computational thinking and a mathematics topic, skill, or literacy.
2. The study describes a teaching activity and reports evaluation results.

To improve the search, we required that a term with the prefix "math" appear in the title, abstract, or keywords. The final search string was (("computational thinking" AND (abstract:math) OR document_title:math OR keyword:math")).

Exclusion criteria were applied: if the article was a tutorial, demonstration, panel, proposal, or interview. If the article focused on computational thinking or computer science education but did not connect it to mathematics and article was outside the scope.

Searches were run, retrieving 200 articles for further screening. After screening titles and abstracts, 58 articles met at inclusion criterion and the others were excluded. Table 1 summarizes the results.

Table 1: Statistics for retrieved articles

Database	Number of Articles
ACM	23
IEEE Xplore	2
ScienceDirect	9
SpringerLink	8
Other Sources	16

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1718009

Articles were divided into two groups. Group 1 involve teaching activities with reported evaluations. Group 2 show strong emphasis on theoretical and analytical discussions of computational thinking and mathematics.

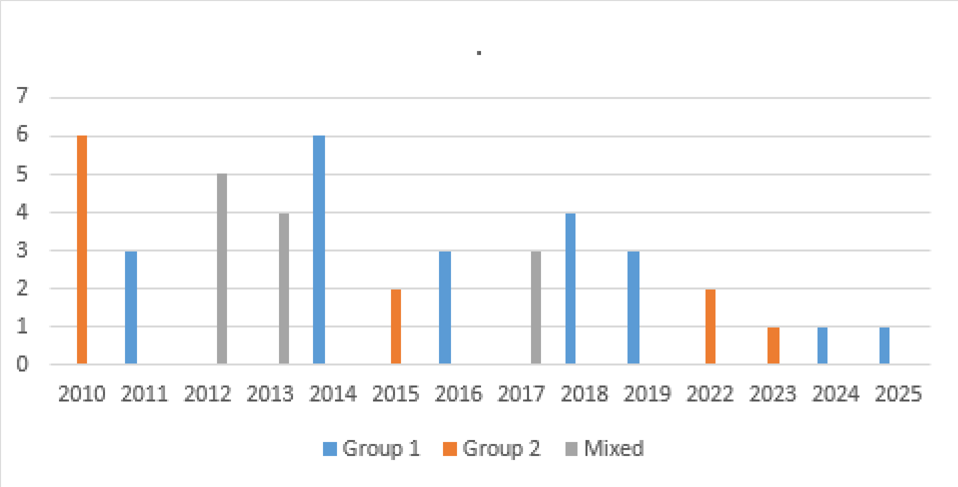

Figure 1: Distribution of References by Year and Group Classifications

Figure 1 shows the distribution of articles over the years, categorized into Group 1 (Teaching activity and evaluation), Group 2 (Theoretical/analytical work), and Mixed classifications reflecting the evolving role of computational thinking in mathematics education..

IV. RESULTS

In each of the groups, the intended audience was identified. Most of the databases consulted provided articles in English, the United States of America educational terminology was applied. The results were summarized in Table 2.

Table 2: Educational Level of Target Audience

Educational Level	No. of Articles	%	Sample Studies
High School / Secondary School	20	36%	Kazimoglu et al. (2012); Leonard et al. (2016); Moreno-León et al. (2016); Sengupta et al. (2013); Winthrop et al. (2016); Wing (2008); Nordby et al. (2024); Lehmann (2025)
Higher Education (University/College)	18	32%	Malan & Leitner (2007); Wilson & Shrock (2001); Porter et al. (2013); Saeli et al. (2011); Tan et al. (2014); Olteanu (2022)
Teacher Training / Professional Development	8	14%	Yadav et al. (2017); Kale et al. (2018); Lee et al. (2019); Mahadev & Conner (2014)
General / Cross-level (Theoretical, Curriculum, Policy, Textbooks)	9	16%	Kafai & Burke (2014); Lye & Koh (2014); Rosen (2019); Sinclair & Patterson (2018); Springer (2023)
Other (ICT in society, interdisciplinary STEM not tied to level)	3	2%	Umar & Musa (2022); Stozhko et al. (2015); Pruski & Friedman (2014)

High school/Secondary School has more articles which reflects more on need for programming, computer science courses, mathematics and sciences hence show the growing importance of computational thinking in schools' curricula. Teacher training is important for CT integration in the classroom.

4.1 Software Tools

Some of the activities relied on software tools, with 35 studies using software and 7 using unplugged

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1718009

methods. Visual programming were most common, with Scratch and variants used in 9 studies. Eight studies used custom learning objects or specialized software, such as Octave or MATLAB for estimation (Arroyo et al. 2017). An interactive gallery of graphical artifacts (Wilkerson-Jerde 2014), and a simulation tool (Alab, Magan and Garci 2013). Below is the summary of the software tools used.

Table 3: Software Tools in Educational Activities

Software Tool / Category	No. of Articles	%	Studies
Scratch	2	6%	Malan & Leitner (2007); Kazimoglu et al. (2012)
Agent-based programming	1	3%	Sengupta et al. (2013)
Specific-purpose software tool	4	12%	Tan et al. (2014, Laboratory Animal System); Park & Mills (2014, LMS); Moreno-León et al. (2016, Code to Learn); Stozhko et al. (2015, project-based learning tech)
Robotics + Simulation	2	6%	Leonard et al. (2016, robotics/game design); Nordby et al. (2024, robotics in math classrooms)
Java, Python	2	6%	Seng & Yatim (2014, OOP with games); Olteanu (2022, programming + reasoning)
MATLAB	1	3%	Pruski & Friedman (2014, integrative teaching with Matlab)
Other (general programming tools, ICT, textbooks)	23	64%	Kafai & Burke (2014, MIT Press); Wing (2008, CT perspective); Rosen (2019, textbook); Umar & Musa (2022, ICT in math); Sinclair & Patterson (2018, programming + geometry)

Scratch and specific-purpose tools like LMS, Code to Learn, and project-based learning platforms appear in multiple teaching-focused studies. Robotics simulation is becoming popular in year 2016 to date. MATLAB and Java/Python are not frequently used. The majority fall under general programming environments, ICT applications, and theoretical works that are not tied to a single tool.

4.2 Research Strategies for Evaluation

For quantitative studies, Trochim and Donelly (2006) recommend random assignment, but quasi-experimental designs are common in education research. For qualitative studies, Creswell and Miller (2000) suggest prolonged engagement and detailed description.

Table 4: Research Approaches and Methods

Methodology	No. of Articles	Articles %	Selected Studies
Quasi-experiment	6	11%	Kazimoglu et al. (2012); Leonard et al. (2016); Seng & Yatim (2014); Tan et al. (2014); Lehmann (2025); Nordby et al. (2024)
Report (descriptive, curriculum, framework)	12	22%	Wing (2008); Moreno-León et al. (2016); Weintrop et al. (2016); Sinclair & Patterson (2018); Springer (2023)
Qualitative case study	8	15%	Malan & Leitner (2007); Porter et al. (2013); Park & Mills (2014); Stozhko et al. (2015); Yadav et al. (2017)
Qualitative (general analysis, interviews, document analysis)	15	27%	Kafai & Burke (2014); Kale et al. (2018); Lee et al. (2019); Olteanu (2022); Lye & Koh (2014)
Experiment (controlled, comparative)	14	25%	Wilson & Shrock (2001); Saeli et al. (2011); Taylor et al. (2010); Lambic (2011); Pruski & Friedman (2014)

Qualitative approaches (general and case studies) reflects the analysis, interviews, and classroom case studies. Experiments especially in higher education contexts (CS courses, math applications). Quasi-experiments though fewer but notable, especially in recent years.

Table 5: Qualitative Approaches and Methods

Instrument	No. of Articles	%
Pre- and post-tests	10	18%
Classroom observation	8	14%
Survey	12	22%
Interview	9	16%
Artifacts development (student work, code, projects)	7	13%
Others (documents, textbooks, policy analysis)	8	14%

Surveys with 22% are the most common instrument, in the studies. Pre-test and post-tests were used in quasi-experiments and controlled studies to measure learning improvement. Classroom observations and interviews in qualitative case studies. Artifacts development are used to analyze student-created code, projects, and outputs on problem-solving.

4.3 Mathematical Topics and Skills

The activities addressed a wide range of mathematical topics, on planar geometry, algebra, Numerical Operations etc. Mapped contents that improve skills identified by Barcelos and Silveira (2013). The recognition of relationships and patterns was common in algebra, statistics, and modeling. Building descriptive and representative models was linked to modeling on studies on geometry. Visual programming tools consistently addressed Cartesian coordinates and analytic geometry (Lewis and Shah 2012; Taylor et al. 2010). 3D environments sometimes required spatial geometry (Taylor et al. 2010). Physics topics, especially kinematics, appeared on some studies due to their basis in mathematics and science (Sengupta et al. 2013; Hoji, Vianna and Felix 2012). Few studies used mathematical modeling, despite its relevance (Buteau and Muller 2017; Oliveira 2012).

The grouping of studies that develop similar topics, we identified Math topics are shown in Table 6.

Table 6: Mathematical Topics in Computational Thinking Studies

Math Topic	No. of Articles	%	Studies
Euclidean Geometry	6	11%	Sinclair & Patterson (2018); Nordby et al. (2024); Sengupta et al. (2013); Tan et al. (2014)
Algebra	5	9%	Olteanu (2022); Lee et al. (2019); Kazimoglu et al. (2012); Lehmann (2025)
Numerical Operations	4	7%	Wilson & Shrock (2001); Lambic (2011); Taylor et al. (2010)
Mathematical Modeling	7	13%	Weintrop et al. (2016); Leonard et al. (2016); Pruski & Friedman (2014); Sengupta et al. (2013)
Others (general CT integration, curriculum, ICT, problem-solving frameworks)	32	60%	Wing (2008); Kafai & Burke (2014); Lye & Koh (2014); Moreno-León et al. (2016); Springer (2023)

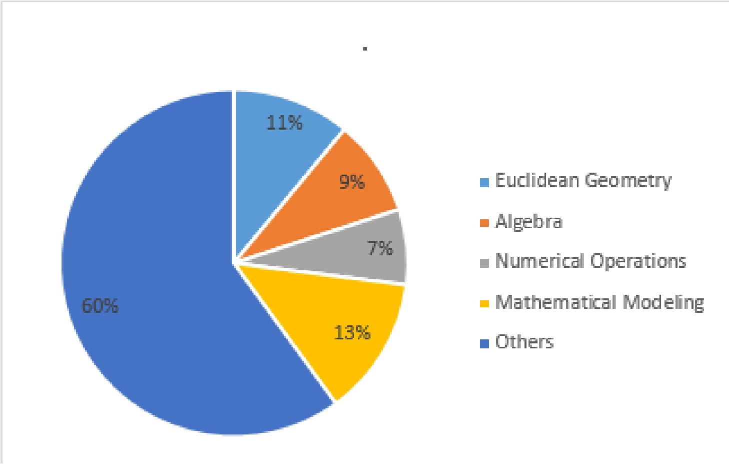

Figure 2: Distribution of CT Problem-Solving on Mathematics

V. DISCUSSION

The analysis of 58 studies reveals that computational thinking activities targeting mathematics learning are distributed across multiple educational levels, with the highest concentration in high school/ secondary school (32%) and University education (36%). This distribution reflects two trends; the integration of programming and computational concepts into school curricula, and the use of CT as an educational strategy to solve conceptual difficulties and improve students' problem-solving skills. The relatively low proportion of studies focused on teacher training and professional development (14%) suggests that the preparation of educators remains a problem to a wider classroom adoption, consistent with earlier findings on the challenges of scaling Computational Thinking (Yadav et al., 2017; Kale et al., 2018).

The instructional tools, visual programming environments such as Scratch and domain-specific platforms dominate the empirical studies, accounting for 18% of tool-specific applications. Their occurrence aligns with the need for low entry points that allow students to express mathematical ideas through code and visual feedback. Robotics and simulation environments have gained attention since 2016, indicating a shift toward contextualized problem solving. In contrast, general-purpose languages like MATLAB, Java, and Python are less common on students and teacher training exercises with computational thinking, likely due to high prerequisite knowledge on the software. The large category of other tools (64%) encompasses theoretical papers, curriculum frameworks, and ICT applications not tied to a single environment, suggesting that conceptual, classroom interventions could shape the curriculum.

Qualitative approaches, including case studies, interviews, and document analysis, constitute 42% of the studies, reflecting the exploratory nature of majority of the research on Computational Thinking and mathematics. Controlled experiments and quasi-experiments account for 36%, with a noticeable increase in quasi-experimental designs in recent years. This pattern indicates a gradual movement toward stronger inference, though randomization remains rare due to the constraints of the education system. In terms of data collection, surveys (22%) and pre/post-tests (18%) are the most frequently used instruments for measuring assessments, while artifact analysis of student code and projects (13%) remains underutilized despite its potential to reveal procedural understanding and representation (Brennan & Resnick, 2012).

The studies address a broad range of topics, but three areas were commonly used were mathematical modeling (13%), Euclidean geometry (11%), and algebra (9%). Modeling tasks frequently link computational simulation to real-world scenarios, supporting the development of representational translation skills identified by Barcelos and Silveira (2013). Geometry and algebra activities could be used on visual programming to make Cartesian coordinates, transformations, and functional relationships. The relative scarcity of studies on numerical operations and the limited use of mathematical modeling as a primary strategy suggest opportunities for expanding the scope of CT integration.

These findings indicate that computational thinking is being operationalized in mathematics education through a combination of accessible programming environments, inquiry-based tasks, and different assessment strategies.

VI. CONCLUSION

Computational thinking encompasses skills from computer science that should be developed from early age in schools. Its relationship to traditional subjects and the outcomes of integrated strategies is necessary. This systematic review synthesized empirical evidence on how computational thinking activities influence mathematics learning across in, primary, secondary and university education, and teacher training programs. The educational interventions span different mathematics topics. The results show that Computational Thinking is being implemented through software tools, visual programming and simulation being the most predominant in practice-oriented studies.

The review is characterized methodologically by a combination of qualitative, experimental, and quasi-experimental designs. Recent studies demonstrate improved rigor on assessments, reliance on single data sources remains common, which reduce the validity of inferences about learning processes. Teacher training and professional development are underrepresented, despite their critical role in sustaining classroom implementation. Addressing these gaps will strengthen the base for integrating computational thinking into mathematics education aimed at preparing students and policymakers for a digital society.

Recommendations and Future research

1. Embed computational thinking across subjects from the primary level onwards, ensuring pupils develop these skills alongside mathematics, science, and literacy.
2. Prioritize professional development programs that equip educators with both conceptual understanding and pedagogical strategies for computational thinking.
3. Develop interdisciplinary activities that connect computational thinking with other subjects, improving problem-solving skills.
4. Expand research and practice to include early childhood education and teacher education, addressing currently underrepresented groups.
5. Conduct extended investigations to trace the development of computational skills over time and assess the sustained impact of integrated strategies.

REFERENCES

[1] Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832-835. https://doi.org/10.1093/comjnl/bxs074

[2] Ardito, G., Mosley, P., & Scollins, L. (2014). We, robot: Using robotics to promote Collaborative and mathematics learning in a middle school classroom. (Report). Middle Grades Research Journal, 9(3), 73.

[3] Arroyo, I., Woolf, B. P., Burel, C., & Muldner, K. (2017). A mobile game for measurement estimation: Design, evaluation, and lessons learned. *International Journal of Artificial Intelligence in Education, 27*(2), 448–475. https://doi.org/10.1007/s40593-016-0117-6

[4] Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75(B), 661-670. https://doi.org/10.1016/j.robot.2015.10.008

[5] Barcelos, T. S., & Silveira, I. F. (2012). Towards shared competencies between mathematics and computer science: An analysis of computational thinking. In *Proceedings of the 2012 IEEE Frontiers in Education Conference*, 1–6. IEEE. https://doi.org/10.1109/FIE.2012.646237

[6] Barcelos, T. S., & Silveira, I. F. (2013). An analysis of curriculum guidelines for computational thinking in K-12 education. In *Proceedings of the 2013 IEEE Frontiers in Education Conference*, 1587–1593. IEEE. https://doi.org/10.1109/FIE.2013.6685111

[7] Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved?

[8] and what is the role of the computer science education community? *Inroads*, 2(1), 48-54. https://doi.org/10.1145/1929887.1929905

[9] Belanger, C., Christenson, H., & Lopac, K. (2018). Confidence and common challenges:

the effects of teaching computational thinking to students ages 10-16. St. Catherine University, MN. https://sophia.stkate.edu/maed/267

[10] Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. Digital Experiences in Mathematics Education, 3(2), 115-138. https://doi.org/10.1007/s40751-017-0028-x

[11] Borrego, M., & Newswander, L. K. (2010). Definitions of interdisciplinary research: Toward graduate-level interdisciplinary learning outcomes. The Review of Higher Education, 34(1), 61-84. https://doi.org/10.1353/rhe.2010.0006

[12] Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA)*. Vancouver, Canada.

[13] Burke, Q., & Kafai, Y. B. (2010). Programming & storytelling: opportunities for learning about coding & composition. *Proceedings of the 9th International Conference on Interaction Design and Children*, 348-351. https://doi.org/10.1145/1810543.1810611

[14] Burns, R., Pollock, L., & Harvey, T. (2012). Integrating hard and soft skills: software engineers serving middle school teachers. *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, 209-214. ACM.

[15] Cai, W., & Sankaran, G. (2015). Promoting critical thinking through an interdisciplinary study abroad program. *Journal of International Studies*, 5(1), 38-49. https://doi.org/10.32674/jis.v5i1

[16] Cassel, L. N. (2011). Interdisciplinary computing is the answer: now, what was the question? *ACM Inroads*, 2(1), 4-6. https://doi.org/10.1145/1929887.1929888

[17] Coenraad, M., Mills, K., Byrne, V. L., & Ketelhut, D. J. (2020). Supporting teachers to integrate computational thinking equitably. *Proceedings of 2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology*. https://doi.org/https://doi.org/10.1109/RESPECT49803.2020.9272488

[18] Creswell, J. W., & Miller, D. L. (2000). Determining validity in qualitative inquiry. *Theory into Practice, 39*(3), 124–130. https://doi.org/10.1207/s15430421tip3903_2

[19] Czerkawski, B. C., & Lyman, E. W., III. (2015). Exploring issues about computational thinking in higher education. *TechTrends*, 59(2), 57-65. http://dx.doi.org/10.1007/s11528-015-0840-3

[20] Dagdilelis, V., Satratzemi, M., & Evangelidis, G. (2004). Introducing secondary education students to algorithms and programming. *Education and Information Technologies*, 9(2), 159-173. https://doi.org/10.1023/B:EAIT.0000027928.94039.7b

[21] Davenport, J. H., Wilson, D., Graham, I., Sankaran, G., Spence, A., Blake, J., & Kynaston, S. (2014). Interdisciplinary teaching of computing to mathematics students: Programming and discrete mathematics. MSOR Connections, 14(1), 1-8. https://doi.org/10.11120/msor.2014.00021

[22] 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. https://doi.org/10.1016/j.compedu.2011.08.006

[23] Denning, P. J. (2005). Is computer science science? *Communications of the ACM, 48*(4), 27–31. https://doi.org/10.1145/1053291.1053309

[24] Djurdjevic A., Pahl, C., Fronza, I., & El Ioini, N. (2017). A pathway for introducing computational thinking to K-12 education. In *Proceedings of the 2017 IEEE Global Engineering Education Conference (EDUCON)*, 1509–1516. IEEE. https://doi.org/10.1109/EDUCON.2017.7943027

[25] Draganoiu, R., Moldoveanu, A., & Braescu, A. (2017). The Math of Programming-Interdisciplinary Approach. The International Scientific Conference eLearning and Software for Education, Carol National Defense University, 2, 473-481. https://doi.org/10.12753/2066-026X-17-152

[26] Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? Schulte (Ed.), *Proceedings of the 9th Workshop in*

Primary and Secondary Computing Education, 60-69. ACM
[27] Feaster, Y., Segars, L., Wahba, S. K., & Hallstrom, J. O. (2011). Teaching CS unplugged in the high school (with limited success). *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, 248-252. ACM.
[28] Fofang, J. S., Weintrop, D., Walton, M., Elby, A., & Walkoe, J. (2020). Mutually supportive mathematics and computational thinking in a fourth-grade classroom. In Gresalfi, M. and Horn, I. S. (Eds.), the Interdisciplinary of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS), Nashville, Tennessee: International Society of the Learning Sciences, 3, 1389-1396.
[29] Fojtik, R. (2014). Design patterns in the teaching of programming. Procedia - Social and Behavioral Sciences, 143, 352-357. https://doi.org/10.1016/j.sbspro.2014.07.493
[30] Gadanidis, G. (2017). Artificial intelligence, computational thinking, and mathematics education. *The International Journal of Information and Learning Technology*, 34(2), 133–139. doi:10.1108/ijilt-09-2016-0048
[31] Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). Computational thinking in mathematics teacher education. Contemporary Issues in Technology and Teacher Education, 17(4), 458-477.
[32] Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in K-12. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, 57-62. ACM.
[33] Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38-43. https://doi.org/10.3102%2F0013189X12463051
[34] Guzdial, M., & DiSalvo, B. (2013). Computing education: Beyond the classroom. Computer, 46(9), 30-31. https://doi.org/10.1109/MC.2013.306
[35] Guzdial, M. (2004). Programming environments for novices. Computer Science Education Research, 127-154. Routledge Falmer.
[36] Hemmendinger, D. (2010). A plea for modesty. *ACM Inroads, 1*(2), 4–7. https://doi.org/10.1145/1805724.1805726
[37] Hoji, E. S., Vianna, W. B., & Félix, T. A. (2013). Computer-aided teaching of math in electromechanics vocational course. *International Journal of Engineering Pedagogy (iJEP), 3*(S2), 39–45. https://doi.org/10.3991/ijep.v3iS2.2445
[38] Hood, L. (2005). The mathematics and science education system: Fluency with information technology in K–12. National Academies Press
[39] Hurley, P. J. (2012). A Concise Introduction to Logic (11th ed.). Boston, MA: Wadsworth.
[40] Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. Computers and Education. https://doi.org/10.1016/j.compedu.2018.07.004
[41] Husain, H., Kamal, N., Ibrahim, M. F., Huddin, A. B., & Alim, A. A. (2017). Engendering Problem solving skills and mathematical knowledge via programming. *Journal of Engineering Science and Technology*, 12(12), 1–11.
[42] Isbell, C. L., Stein, L. A., Cutler, R., Forbes, J., Fraser, L., Impagliazzo, J., Proulx, V., Russ, S., Thomas, R., & Xu, Y. (2010). (Re)Defining computing curricula by (re)defining computing. *Proceedings of ITiCSE*, ACM Press, 123–130. https://doi.org/10.1145/1595496.1595520
[43] Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. Computers & Education, 82, 263-279. https://doi.org/10.1016/j.compedu.2014.11.022
[44] Jehlička, V. (2010). Interdisciplinary relations in teaching of programming. *Proceedings of the 2010 International Conference on Applied Computing*. World Scientific and Engineering Academy and Society (WSEAS), 33-38.
[45] Jonassen, D. H. (2000). Computers as mindtools for schools: Engaging critical thinking (2nd ed.). Upper Saddle River, NJ: Prentice Hall.
[46] Jones, C. (2010). Interdisciplinary approach-advantages, disadvantages, and the future benefits of interdisciplinary studies. Essai, 7(1), Article 26. http://dc.cod.edu/essai/vol7/iss1/26

[47] Kafai, Y. B., Franke, M. L., Shih, J. C., & Ching, C. C. (1998). Game design as an interactive learning environment for fostering students' and teachers' mathematical inquiry. *International Journal of Computers for Mathematical Learning, 3*(2), 149–184. https://doi.org/10.1023/A:1009796009182
[48] Kafai, Y., & Burke, Q. (2014). Connected code: why children need to learn programming. MIT.
[49] Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018).
[50] Computational what? Relating computational thinking to teaching. *TechTrends*, 62(6), 574-584. https://doi.org/10.1007/s11528-018-0290-9
[51] Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning Programming at the Computational Thinking Level via Digital Game-Play.
[52] Ke, F. (2014). An exploration of designing integrated math and computer science learning environments. *Computers & Education, 73*, 111–120.
[53] https://doi.org/10.1016/j.compedu.2014.01.006
[54] Kitchenham, B. (2004). *Procedures for performing systematic reviews.* Keele University Technical Report TR/SE-0401. Keele, UK.
[55] Krone, J., Sitaraman, M., & Hallstrom, J. O. (2011). Mathematics throughout the CS curriculum. *Journal of Computing Sciences in Colleges, 26*(6), 123–131.
[56] Lambic, D. (2011). Presenting practical application of mathematics by the use of programming software with easily available visual components. Teaching Mathematics and Its Applications: *An International Journal of the IMA*, 30(1), 10–18. doi: 10.1093/teamat/hrq014
[57] Lee, I., Grover, S., Martin, F., & Pillai, S. (2019). Computational Thinking
[58] From a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology, Engineering, and Mathematics Education. Journal of Science Education and Technology, 29(1), 1–8. doi:10.1007/s10956-019-09803-w
[59] Lehmann, T. H. (2025). Examining the interaction of computational thinking skills and heuristics in mathematical problem solving. *Research in Mathematics Education, 27*(2), 269–290. https://doi.org/10.1080/14794802.2025.1234567
[60] Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O., Hubert, T. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology*, 25(6), 860–876. doi: 10.1007/s10956-016-9628-2
[61] Lewis, C., & Shah, N. (2012). Building up informatics concepts: Scratch as a tool for learning coordinate geometry. *Proceedings of SIGCSE 2012*, ACM Press, 483–488. https://doi.org/10.1145/2157136.215729
[62] Lin, Y.-T., Wang, M.-T., & Wu, C.-C. (2019). Design and Implementation of Interdisciplinary STEM Instruction: Teaching Programming by Computational Physics. Asia-Pacific Education Researcher, 28(1), 77-91. https://doi.org/10.1007/s40299-018-0415-0
[63] 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. https://doi.org/10.1016/j.chb.2014.09.012
[64] Mahadev, A., & Conner, S. (2014). An interdisciplinary approach to teaching a computer ethics course. *Journal of Computing Sciences in Colleges*, 29(5), 202-207. https://dl.acm.org/doi/10.5555/2600623.2600663
[65] Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum? Journal of Information Technology Education: Research, 15, 283–303. doi: 10.28945/3521
[66] Nordby, S. K., Mifsud, L., & Bjerke, A. H. (2024). Computational thinking in primary Mathematics classroom activities. *Frontiers in Education, 9*, Article 123456. https://doi.org/10.3389/feduc.2024.123456
[67] Oliveira Aureliano, V. C. (2013). A methodology for teaching programming for beginners. *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, 169-170. ACM.
[68] Olteanu C. (2022). Programming, mathematical reasoning and sense-making.

International Journal of Mathematical Education in Science and Technology 53(8), 2046-2064.

[69] Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books

[70] Park, J. Y., & Mills, K. A. (2014). Enhancing interdisciplinary learning with a learning management system. *Journal of Online Learning and Teaching*, 10(2), 299-313. https://jolt.merlot.org/vol10no2/park_0614.pdf

[71] Pióro, M. (2006). Mathematics and computer science: Connections and challenges. *Journal of Telecommunications and Information Technology, 4*(3), 43–56

[72] Polya, G. (2004). *How to solve it: A new aspect of mathematical method* (Expanded ed.). Princeton University Press.

[73] Porter, L., Guzdial, M., McDowell, C., & Simon, B. (2013). Success in introductory programming: What works? Communications of the ACM, 56(8), 34-36. https://doi.org/10.1145/2492007.2492020

[74] Pruski, L., & Friedman, J. (2014). An integrative approach to teaching mathematics, computer science, and Physics with Matlab. *Mathematics and Computer Education*, 48(1), 6-18.

[75] Ralston, A. (2005). Do we need ANY mathematics in computer science curricula? ACM SIGCSE Bulletin*, 37*(2), 6–9. https://doi.org/10.1145/1067445.1067456

[76] Rosen K. (2019). Discrete Mathematics and Its Applications. 8th Edition, illustrated Publisher McGraw-Hill, ISBN1260091996, 9781260091991.

[77] Saeli, M., Perrenet, J., &Jochems, W. M.. (2011). Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education*, 10(1), 73-88. https://www.ceeol.com/search/article-detail?id=69618

[78] Seng, W. Y., & Yatim, M. H. M. (2014). Computer game as learning and teaching tool for object oriented programming in higher education institution. *Procedia - Social and Behavioral Sciences*, 123, 215-224. https://doi.org/10.1016/j.sbspro.2014.01.1417

[79] Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: a theoretical framework. *Education and Information Technologies*, 18, 351-380. https://doi.org/10.1007/s10639-012-9240-x

[80] Sinclair, N., & Patterson, M. (2018). The Dynamic Geometrisation of Computer Programming. Mathematical Thinking and Learning, 20(1), 54–74. doi: 10.1080/10986065.2018.1403541

[81] Springer (2023). Handbook of Digital Resources in Mathematics Education, Springer International Publishing, 1-38, 2023, Springer International Handbooks of Education, 978-3-030-95060-6

[82] Stozhko, N., Bortnik, B., Mironova, L., Tchernysheva, A., E. (2015). Interdisciplinary project-based learning: technology for improving student cognition. Research in Learning Technology, 23, Article 27577. https://doi.org/10.3402/rlt.v23.27577

[83] Taub, R., Armoni, M., & Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. ACM Transactions on Computing Education (TOCE), 12(2), Article 8. https://doi.org/10.1145/2160547.2160551.

[84] Taylor, M., Harlow, A., & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia - Social and Behavioral Sciences*, 8, 561–570. doi: 10.1016/j.sbspro.2010.12.078

[85] Thies, R., & Vahrenhold, J. (2013). On plugging unplugged into CS classes. In Proceedings of the 44th ACM Technical Symposium on Computer Science Education, 365-370.

[86] Trochim, W. M., & Donnelly, J. P. (2006). *The research methods knowledge base* (3rd ed.). Atomic Dog Publishing.

[87] Umar, A., & Musa, S. (2022). ICT and Learning of Mathematics in Nigeria. Journal of Mathematics Instruction, Social Research and Opinion, 1(3), 143-152. ISSN 2962-7842.

[88] Weintrop, D., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25, 127-147

[89] Wilkerson-Jerde, M. H. (2014). Construction, categorization, and consensus: Student-generated computational artifacts as a context for disciplinary reflection. *Educational Technology Research and Development, 62*(6),

767–791. https://doi.org/10.1007/s11423-013-9327-0

[90] Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.111821

[91] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Systematic literature reviews in software engineering. In *Experimentation in Software Engineering*, 45–54. Springer, Berlin. https://doi.org/10.1007/978-3-642-29044-2_4

[92] Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. P. J. Rich & C. B. Hodges (Eds.), Emerging research, practice, and policy on computational thinking, 205-220.