Informatics in Education, 2024, Vol. 23, No. 1, 223–278

DOI: 10.15388/infedu.2024.08

Interdisciplinary Integration of Computational Thinking in K-12 Education: A Systematic Review

Sabiha YENI[1], Nataša GRGURINA[2], Mara SAELI[3],
Felienne HERMANS[4], Jos TOLBOOM[5], Erik BARENDSEN[6]

[1]*The Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands*
[2]*Teacher Education, Faculty of Behavioral and Social Sciences, University of Groningen, Groningen and Netherlands Institute for Curriculum Development, Amersfoort, The Netherlands*
[3]*Institute for Science Education, Radboud University, Nijmegen, The Netherlands*
[4]*Faculty of Science, Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands*
[5]*Netherlands Institute for Curriculum Development, Amersfoort, The Netherlands*
[6]*Institute for Science Education, Radboud University, Nijmegen and Department of Computer Science, Open University, The Netherlands*
e-mail: s.yeni@liacs.leidenuniv.nl, n.grgurina@rug.nl, mara.saeli@ru.nl, f.f.j.hermans@vu.nl, j.tolboom@slo.nl, erik.barendsen@ru.nl

Received: March 2023

Abstract. There is an increasing interest in the integration of computational thinking (CT) in the K-12 curriculum. By integrating CT into other disciplines, the aim is to equip students with essential skills to navigate domain-specific challenges. This study conducts a systematic review of 108 peer-reviewed scientific papers to analyze in which K-12 subjects CT is being integrated, learning objectives, CT integration levels, instructional strategies, technologies and tools employed, assessment strategies, research designs and educational stages of participants. The findings reveal that: (a) over two-thirds of the CT integration studies predominantly focus on science and mathematics; (b) the majority of the studies implement CT at the substitution level rather than achieving a transformation impact; (c) active learning is a commonly mentioned instructional strategy, with block-based languages and physical devices being frequently utilized tools; (d) in terms of assessment, the emphasis primarily lies in evaluating attitudes towards technology or the learning context, rather than developing valid and reliable assessment instruments. These findings shed light on the current state of CT integration in K-12 education. The identified trends provide valuable insights for educators, curriculum designers, and policymakers seeking to effectively incorporate CT across various disciplines in a manner that fosters meaningful skill development with an interdisciplinary approach. By leveraging these insights, we can strive to enhance CT integration efforts, ensuring the holistic development of students' computational thinking abilities and promoting their preparedness for the increasingly interdisciplinary domains of digital world.

Keywords: computational thinking, integration, interdisciplinary, K-12 curriculum, contextualized computing, pedagogical issues.

1. Introduction

The problems we face in the world are complex and require the embedding of multiple disciplines, concepts, and skills to solve them. Tedre *et al.* (2018, p. 177) consider training for computational problem-solving “a truly interdisciplinary undertaking” that increasingly addresses computing for other disciplines (Guzdial, 2019). This idea is captured in the term Computational Thinking (CT), which was coined by Papert (1980, p. 182) who regarded it as “procedural thinking”. Wing (2006), who promoted and popularized the notion of CT, defined it as “[…] the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent [...]. These solutions can be carried out by any processing agent, whether human, computer, or a combination of both” (Cuny *et al.*, 2010) cited in (Wing, 2011, p. 11). The emphasis in this study is on individuals performing a thought process, not on the production of artefacts or evidence. CT is used in the design and analysis of problems including their solutions, combining an attitude and a skill set which is necessary not just for computer scientists, but for every active member of the 21st century.

Following Wing’s seminal paper (2006), several definitions of CT were developed emphasizing various aspects of this CT problem-solving process. The Computational Thinking Task Force of the Computer Science Teachers Association (CSTA) in the US views CT as a problem-solving process that puts emphasis on the construction of a computational solution for a given problem, after that problem has been expressed in computational terms. By this definition, programming is considered to be an essential aspect of CT (Force, 2011). In line with this reasoning, Brennan and Resnick (2012) define CT along three dimensions: computational concepts (i.e., sequences, loops, events, parallelism, conditionals, operators and data), computational practices (i.e., being incremental and iterative, testing and debugging, reusing and remixing, and, abstracting and modularizing), and computational perspectives (i.e., expressing, connecting and questioning) (Brennan and Resnick, 2012). Selby (2013, p. 5) found common ground in various definitions of CT and describe CT as “a focused approach to problem solving, incorporating thought processes that utilize abstraction, decomposition, algorithmic design, evaluation, and generalizations”.

The CT problem-solving process in a particular discipline can be viewed as a three-step procedure (Kallia *et al.*, 2021). First, a problem or question is expressed in computational terms such as data or processes, thus allowing for the use of computing to solve it. Second, a computational solution is constructed, either by using existing applications or by devising new algorithms and writing new programs. Essential to the nature of CT is that this solution should be executable. Finally, the computational solution is interpreted in terms of the original subject matter, thus providing the solution to the original problem or answering the question. This view of CT highlights the interplay going on between the problem domain – outside Computer Science (CS) – and the construction and use of a computational solution, usually within CS.

In this literature review, we focus on CT in context where these three steps of problem solving are employed to solve problems in a particular discipline outside of CS. We chose to view CT in the spirit of the original Wing's definition of CT as thought process mentioned earlier. Other definitions (Brennan and Resnick, 2012; Force, 2011; Selby, 2013) provide indicators to characterize particular instances of CT in the reviewed studies. This allows us to capture various occurrences of CT, whether using a computer or unplugged.

Not surprisingly, CS courses or programming education are often considered a natural habitat for the development of CT skills (Bers *et al.*, 2014; Davies, 2008; Gouws *et al.*, 2013; Grover, 2011; Walden *et al.*, 2013; Weintrop and Wilensky, 2013). Indeed, Papert (1980) believed that by learning to program, children would develop computational problem-solving skills that they could apply within other contexts. Results in empirical research on this expected transfer are inconclusive, however. It appears that so-called high-road transfer, involving sophisticated applications of the problem-solving skills to other subject matter, requires explicit instruction (Perkins and Salomon, 1989). This suggests that introducing a separate computing or programming module in the curriculum is not sufficient: CT should (also) be learnt within the other subjects. While there are already numerous examples of CT playing a role in the learning of the subject matter in various disciplines outside of the CS education (Hambrusch *et al.*, 2009; Qin, 2009; Ruthmann *et al.*, 2010; Sengupta *et al.*, 2013), Voogt *et al.* (2015) believe that much more attention is needed to ways of incorporating CT skills within existing school subjects such that students can develop creative ways to apply CT within the respective disciplines.

In many countries, the position of digital literacy – and CT in particular – is being discussed. While there are many initiatives to integrate CT into the curriculum (Bocconi *et al.*, 2016), it is not always clear whether the integration concerns actual integration of CT with other disciplines. That is the case, for example, in the Netherlands, where the curriculum in K-9 education is about to be reformed (Curriculum of the Netherland, 2019) and CT is expected to get a prominent position and be taught within other subjects. Another example is the United States where the Next Generation Science Standards (NGSS) (Council *et al.*, 2012) introduce CT as a core practice of science education. These examples are illustrative for a significant and growing attention for the CT from the industry, politicians, researchers and educators alike across the globe. The motivation to embed CT into curriculum comes from the fact that computing concepts and practices have become an integral part of current professional practice and CT is a necessary skill desired by twenty-first century economies (Aydeniz, 2018). Computational approaches are vital for interdisciplinary practices because how these disciplines are practiced in the professional world is rapidly changing (Foster, 2006). In recent years, scientific fields have been supported with computational practices, for example in Bioinformatics, Computational Statistics, Chemometrics and Neuroinformatics (Weintrop *et al.*, 2016). Bringing computational tools and practices into K-12 classrooms gives learners a more realistic view of what these fields are, and better prepares students for professional careers in these disciplines (Grover and Pea, 2018). This sense of authenticity and real-world applicability is important in the effort to motivate diverse and meaningful participation in computational and scientific activities (Weintrop *et al.*, 2016).

In order to facilitate embedding of CT in context across various disciplines of the school curriculum we strive to gain a thorough understanding of the current state of affairs of the embedding of CT in various school subjects in primary and secondary education. This systematic review was conducted to investigate current CT embedded studies in curriculum subjects other than CS, in order to gain good insights into exactly how CT is embedded, taught and assessed. This way, we identify both CT integration practices as well as gaps in the knowledge of how to embed CT into the school curriculum. We examined related subjects, CT-related and subject-related learning objectives (LO's), instructional strategies, assessment strategies, research trends, and ages of participants. In this way, we aimed to reveal the trends and provide a complete interpretation of CT embedded studies.

We have found literature reviews and mapping studies concerned with the embedding of CT in various disciplines other than computer science in the broad sense (Barcelos *et al.*, 2018; de Araujo *et al.*, 2016; Grover and Pea, 2013; Hsu *et al.*, 2018; Lockwood and Mooney, 2017; Lye and Koh, 2014; Martins-Pacheco *et al.*, 2019; Moreno-León and Robles, 2016; Pöllänen and Pöllänen, 2019; Sullivan and Heffernan, 2016). A number of these studies have a narrow focus on programming (Lye and Koh, 2014; Moreno-León and Robles, 2016; Pöllänen and Pöllänen, 2019) or physical devices only (robotics) (Sullivan and Heffernan, 2016), or embedding CT in a particular discipline (Barcelos *et al.*, 2018) or only on teaching and assessment (de Araujo *et al.*, 2016; Martins-Pacheco *et al.*, 2019) and a number of studies do not limit their focus to K-12 education exclusively (Hsu *et al.*, 2018; Lockwood and Mooney, 2017). With the explosive growth of the research into CT, the Grover and Pea's study (2013) which reviewed the CT integration into K-12 is becoming increasingly outdated. Additionally, the selection criteria for the research included in these studies are not always clear. Also, a recent systematic review study (Kampylis *et al.*, 2023) focuses on the analysis of different CT integration/implementation approaches: (i) embedding CT across the curriculum as a transversal theme/skill set; (ii) integrating CT as a separate subject; and (iii) incorporating CT skills within other subjects.

The aim of this review study is to understand the current state of affairs of the embedding of CT in the non-computing school subjects in primary and secondary education. Our systematic literature review differs from the literature reviews and mapping studies mentioned above in three ways: (1) our strict focus on K-12 education, (2) the requirement that the CT-related LO's are embedded into subject-related LO's other than those associated with CS, and finally, (3) our broad and inclusive search process with clear selection criteria.

2. Methodology

This study aims to provide a comprehensive overview of current literature relevant to CT embedded in non-computing school subjects and to gain insight into exactly how CT is embedded, taught and assessed. A systematic procedure was used to choose the publications in order to guarantee the reliability of the data. To increase the reliability of the data, the PRISMA 2020 statement was implemented, which adheres to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses guidelines (Page *et al.*, 2021). Fig. 1 shows the order of steps in the PRISMA 2020 workflow, comprising the Identification, Screening, and Included stages, together with the related number of records maintained at each stage. The following section describes the research questions, the search criteria, the inclusion criteria for documents, and the selection process.

2.1. *Research Questions*

This systematic literature review (SLR) analyzes the integration of CT in context across various disciplines of the K-12 school curriculum other than CS education and CS-related topics. In order to gain thorough understanding of the research targeting this issue, we pose the following research questions:

RQ1: What are the CT-related and subject-related learning objectives and what are the integration levels of the CT integrated studies?
RQ2: What instructional strategies and technologies are being employed in the CT integrated studies?
RQ3: What assessment strategies are being employed in the CT integrated studies?
RQ4: What are the ages/grades of participants involved in the CT integrated studies?
RQ5: What are the research designs and sample sizes of the CT integrated studies?
RQ6: What is the evidence for the effectiveness of the CT integrated studies?

2.2. *Inclusion Criteria*

CS and CT are gaining increasing attention in education and CS education research follows in its wake. We are interested in embedding CT in other disciplines taught in K-12 and where the purpose of CT is to aid the learning of another discipline, rather than CS education. As this SLR serves as a part of a practice-oriented research project on the embedding of CT in an educational setting outside CS courses or programming education, we were not interested in teaching programming, games or robotics for their own sake. Rather, when programming was involved, we were interested in the cases of Explicative Programming where programming is a means to support the understanding of another discipline (Repenning and Basawapatna, 2021). We, therefore, searched for documents describing the embedding of CT in specific subject matter: both theoretical papers describing design principles for such integration, and empirical papers describing examples of such integration. We searched for the papers that were published between 2006 – the year when Wing published her seminal paper (Wing, 2006) introducing Papert's term CT into broad use – and 2019, and met the following criteria which are explained in Table 1.

Table 1
Inclusion & Exclusion Criteria

We included the papers that:	We excluded the papers that:
• Describe a study with respect to a learning objective integrating CT content – such as computational concepts, practices and perspectives or algorithmic design – with another school subject or discipline. • Have an empirical or a theoretical focus when elaborating the interpretation of CT for a specific school subject. • Have been peer reviewed. • Are written in English.	• Contained a learning objective integrating CT content only with CS, gaming or robotics and no other discipline or school subject. • Contained only a description of a tool without a specific mention of integration of CT content with subject matter of other school subjects.

2.3. *Search Process*

We performed a systematic search of six databases: ACM, IEEE, Scopus, Web of Science, PsycINFO and Eric. In order to do so, we first had to establish the inclusion criteria for the papers. Since the embedding of CT in school subjects takes place at the intersection of CT and those subjects, our trial searches focused on the combination of CT with (a list of) specific subjects. This strategy, however, did not lead to satisfactory results for several reasons. First, the term 'Computational Thinking' was too narrow and needed to be supplemented with related concepts such as 'coding' and 'unplugged' (i.e., coding related activities done without the use of technology). Second, it was virtually impossible to list all the school subjects in all the different educational systems as described in the papers contained in these databases. We then decided to focus on activities that are typical for an educational setting, and since these are always situated in the context of teaching a particular school subject, we would capture these subjects along. Finally, we decided to consider only the studies focusing on K-12 education. These considerations led to the following criteria the papers needed to address:

- Both computational thinking and related concepts.
- Educational issues related to any aspects of instruction – and no more than that, so no papers on e.g., educational policies.
- K-12 education.

Each of these criteria led to a separate set of search terms resulting from several iterations of trial runs and finetuning. The documents found through the search in every iteration were checked against an existing list of documents that served as a control list. This process resulted in three sets of search terms, each corresponding to one of the criteria mentioned earlier:

Set 1. programming OR coding OR computational* thinking OR algorithmic thinking OR algorithmic skill* OR unplugged.

Set 2. context OR traject* OR curriculum OR integrat* OR framework* OR scaffold* OR pedagog* OR assess* OR measur* OR evaluat* OR instruct* OR progress* OR didactic* OR infus*.

Set 3. school OR K-12 OR K12 OR K-9 OR K9 OR K-6 OR K6 OR secondary education OR primary education OR elementary education.

This search resulted in 2935 titles (Fig. 1). Many of these papers had indeed nothing to do with computing, but were found because of the search terms programming and coding which are at the core of CS and closely linked to CT, and are also used in a number of disciplines possibly having some relation with education but not related to CS (e.g., mental health programing, content coding). After removing duplicates, the remaining papers were divided among the researchers who each performed a quick scan of titles in their share of the papers. This resulted in 1535 papers remaining. In the second round, the researchers read the titles and the abstracts of their share of the papers. This resulted in 741 papers remaining possibly relevant for the goal of this study. After that, the researchers swapped their shares of the papers and proceeded to read the essential parts of the reported studies, resulting in 273 papers left. During this round of reading, a number of documents were eliminated because they turned out not to be written in English after all, to be posters, abstract of keynotes or theses, or to be unfindable. The researchers discussed together any paper that did not obviously meet the inclusion criteria. In the next step, the 273 documents were subjected to close scrutiny and all the papers

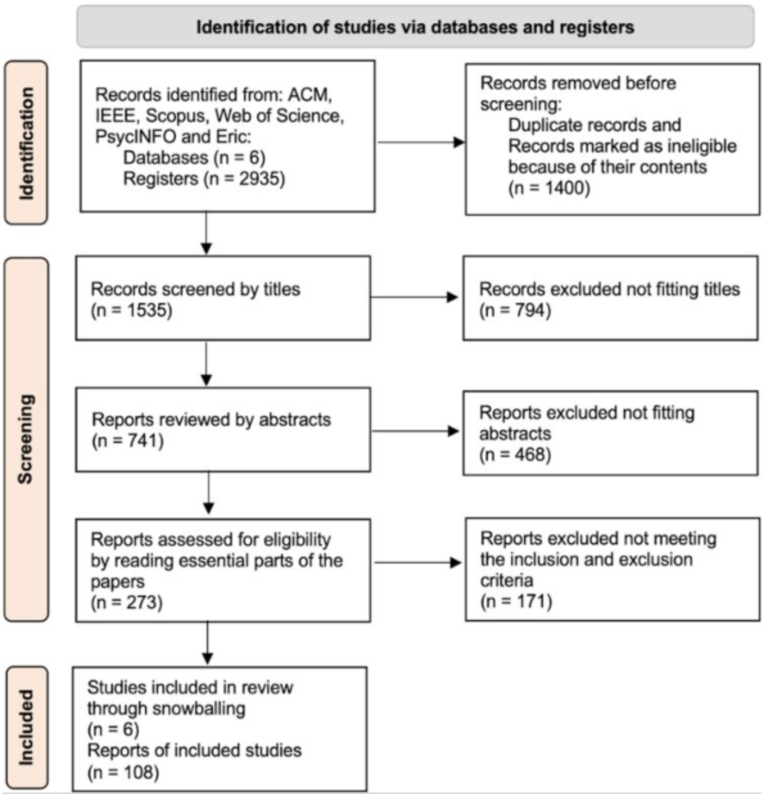

Fig. 1. The literature review process of the study (PRISMA 2020).

which did not clearly fit the inclusion criteria were eliminated. Again, all of the papers not unequivocally meeting the inclusion criteria were discussed. This resulted in a final set of 102 papers. In addition to these 102 papers found through the database search, we identified another six papers through snowballing, making a total of 108 papers that were subjected to detailed analysis.

3. Coding

In this section, we describe the categories to help understanding how we coded the reviewed papers. With our research questions in mind, we first categorized the papers using the following coding categories: target discipline/subject, CT-related LO's, subject-related LO's, the CT integration levels of studies, instructional strategies, technologies and tools, assessment goals and instruments, ages of participants and scale, research methodologies, and finally, effectiveness of studies. We coded deductively for the five categories using the predetermined coding categories (which were derived from the existing theories or studies) listed in sections learning objectives (3.2), integration level (3.3), instructional strategies (3.4), participants (3.7) and research methodologies (3.8). Within the other categories, we analyzed by inductive coding (coding categories were derived directly from the data) and these are the subjects/disciplines themselves (3.1), technologies and tools (3.5), assessment goals and instruments (3.6) and finally, effectiveness of studies (3.9). In an axial coding process (Cohen *et al.*, 2007), the codes were grouped and merged where necessary. While coding data, for the reliability of the qualitative analysis procedure, we adhered to the stability and accuracy aspects. For the stability aspect of coding (Krippendorff, 2018), each theme was coded exclusively by the first or second author. They interpreted the data in the same way and their behaviors remained the same for each theme over time. It helped to eliminate different interpretations of multiple raters that affect decision-making in systematic review. For the accuracy of the coding procedure (Krippendorff, 2018), the coding procedure was conducted according to the pre-agreed code scheme to clarify codes of each theme (especially for the deductive coding part). During the coding, if there was a need for refinement of it, they were resolved via discussion to reach a consensus between researchers. The coding scheme by categories and their definitions for each theme is explained in this section.

3.1. *Target Subject*

We analyzed the papers by inductive coding to distinguish the subjects of the papers into six main categories: art (including art and music), science (including science, biology, physics and chemistry), mathematics (including mathematics and geometry), language arts (including journalism, writing, English, Dutch, Spanish etc.), social sciences (including social science, history, geography and philosophy), and finally various/other (including multidisciplinary approaches where various disciplines come together to address problems).

Table 2
Learning Objectives related to CT

Categories	Definitions	CT subskills
Concepts	The concepts designers use as they program: syntactic, semantic and schematic knowledge commonly used in programming	Sequences, loops, parallelism, events, conditionals, operators, and data
Practices	Knowledge and skills to solve problems during the process of thinking and practices	Abstraction, decomposition, algorithmic thinking, evaluation and generalization (Selby, 2013)
Perspectives	The perspectives designers form about the world around them and about themselves: understandings of personal, social and technological relationships around them	Expressing, connecting, questioning the technology world through computation

3.2. *Learning Objectives*

We categorized the intended learning objectives deductively. This section consists of two parts: the first one focuses on the LO's related to CT and the second one focuses on the subject area. We classified the CT-related LO's according to the triad of Brennan and Resnick (2012): Concepts, Practices, and Perspectives. For practices, the Selby's classification (2013) was used. If a particular study focuses on creating a positive change in learners' attitudes towards technology world and computation, we classified it as 'perspectives' (Table 2). These are categories that in isolation help us analyze the teachers' and researchers' focus. However, in practice there is no sharp division between the subskills. Subject-related LO's are classified according to each subject: art, math, science, language arts, social sciences and others.

3.3. *CT Integration Levels*

CT integrated education can be defined as the approach to teaching the content of CT and other disciplines, bound by CT practices within an authentic subject context to enhance student learning. To clarify the CT integration levels, we adapted the SAMR model (Puentedura, 2010) (Fig. 2). The SAMR framework categorizes four levels of technology integration: Substitution, Augmentation, Modification, and Redefinition. However, the notion of CT includes more than the notion of technology and is an important aspect of the discipline of CS. For this reason, we defined the levels of the adapted CT integration model variously ranging from disciplinary to transdisciplinary approaches (Tress *et al.*, 2005) and included different CT applications (Malyn-Smith *et al.*, 2018; Waterman *et al.*, 2020; Weintrop *et al.*, 2016). We employed the SAMR-model in a two-dimensional matrix that indicates CT integration levels of studies. As we move down the vertical columns, they refer to studies that move from enhance-

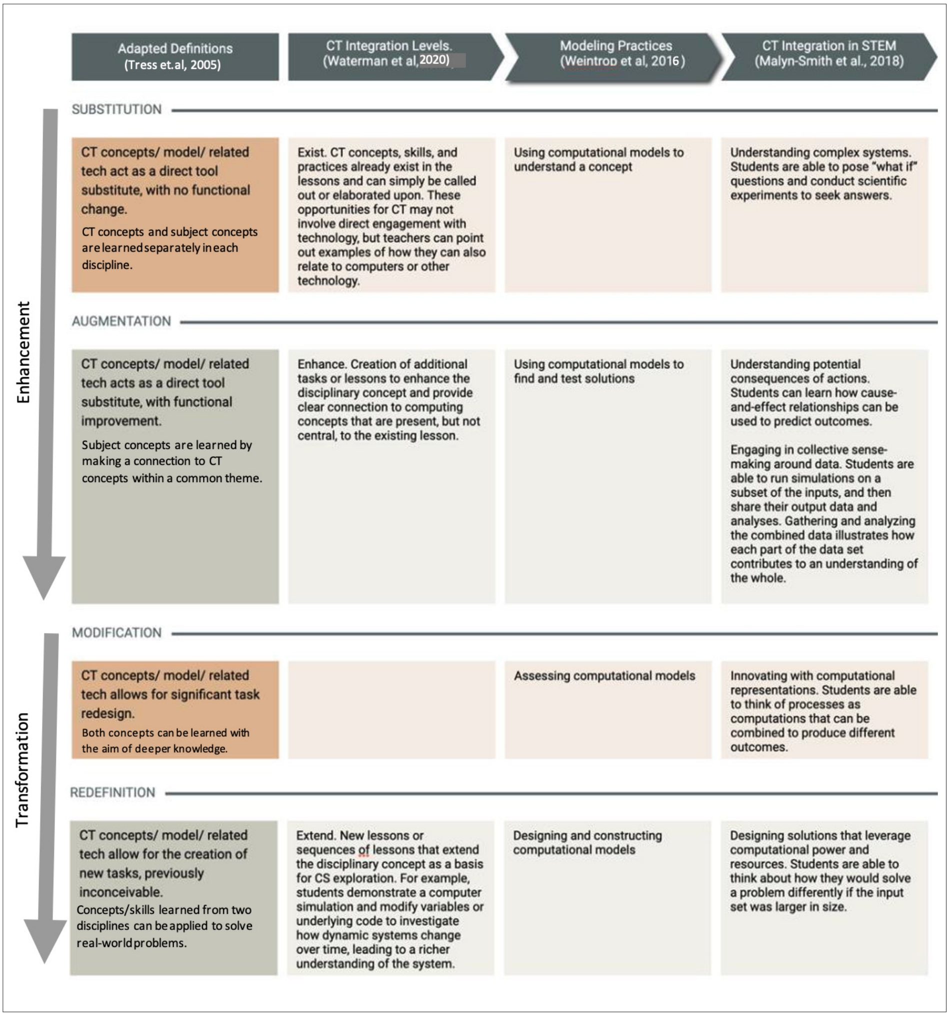

Fig. 2. CT Integration Model (Adapted from the SAMR model).

ment to transformation. If we move horizontally, we can see different definitions and practices of CT integration. The first column contains the adapted definitions with disciplinary perspectives (Tress *et al.*, 2005), while the second column categorizes the CT integration levels as exist, enhance and extend (Waterman *et al.*, 2020). The third column reflects the different levels of integration in modeling applications (Weintrop *et al.*, 2016). The last column presents the different levels of CT integration in STEM integration studies (Malyn-Smith *et al.*, 2018).

We categorized the integration levels of the reviewed studies deductively into four main categories according to the adapted model: substitution, augmentation, modification, redefinition from a disciplinary integration perspective. At the substitution stage, CT concepts/models or related technology are directly substituted for a more traditional one and CT concepts/skills are learned separately in each discipline (disciplinary). Already existing CT concepts and skills were introduced in the subject lessons with no or limited functional change and CT concepts/skills are not closely linked to subject content. At the augmentation stage, CT concepts/models or related technology act as a direct technology substitute with functional improvement. Teachers enhance the disciplinary content by making a connection to CT concepts within a common theme (multi-disciplinary). At the modification stage, the levels of framework convert from enhancement to transformation. There is an actual change in the design of the lesson and learning outcomes. CT concepts/models or related technology allows for significant task redesign and closely linked CT and subject concepts or skills can be learned with the aim of deeper knowledge and skills (interdisciplinary). In the redefinition stage, CT concepts/models or related technology allows for the creation of new tasks, also knowledge and skills learned from two disciplines (CS and subject) can be applied to solve real-world problems (transdisciplinary).

3.4. *Instructional Strategies*

We applied deductive coding to characterize the instructional strategies and we created 19 different categories for instructional strategies: experiential learning (Kolb, 2014), problem-based learning (Wood, 2003), collaborative learning (Dillenbourg, 1999), game-based learning (Akbar *et al.*, 2018), direct instruction (McCoy-Parker *et al.*, 2017), model-based learning (Jacobson *et al.*, 2015), design-based learning (Ke, 2014), scaffolding (Basu *et al.*, 2017), use-modify-create model (I. Lee *et al.*, 2011), project-based learning (Barron *et al.*, 1998), authentic learning (Mingo, 2013), inquiry-based learning (Brown, 2017), exploratory learning (Israel and Lash, 2019), storytelling (W. Q. Burke, 2013), active learning (Burleson *et al.*, 2018), independent learning (Moore, 1973), place-based education (Cunningham and Lansiquot, 2018), embodied learning (Daily *et al.*, 2014) and finally, situated learning (Anderson *et al.*, 1996). Some studies were categorized as ‘undefined’. If the instructional strategy used in the study was specified by the authors of the paper, the paper is categorized according to the authors’ statement. If there is no description about instructional strategy, we have examined the instruction process and classified the appropriate strategy. We categorized some papers for more than one strategy.

3.5. *Technologies and Tools*

We applied inductive coding to characterize the technologies and tools and there are seven main categories: block-based programming tools (e.g. Scratch, Alice etc.), text-based programming tools (e.g. Python, Logo etc.), physical devices (e.g. Arduino, Lego Mindstorm etc.), modeling tools (e.g. NetLogo, StarLogo etc.), unplugged activities (e.g. paper-and-pencil activities, etc.), games (e.g. video games, puzzles etc.) and various (e.g. Google Documents, Spreadsheets, ToonDoo etc.).

3.6. *Assessment Strategies*

We distinguish four goals for assessment inductively in CT-embedded studies: program comprehension and CT skills, subject-related LO's, attitude and satisfaction, and other. We categorized the instruments into five main categories: tests (e.g. quizzes and surveys), performance tasks (e.g. homework, projects, portfolios), self-assessment (e.g. self-reflection, interviews, observations), assessment rubrics, and automated assessment. We coded some assessment instruments with several categories: it is common to use rubrics in combination with project work or automated assessment tools in combination with quizzes or surveys.

3.7. *Educational Stages of Participants*

The studies described in papers were classified according to the participants' educational stages and number of participants. The categories are: preschool (PS), primary (P1 includes Grade 1–4 and P2 includes Grade 5–6), secondary (S1 includes Grade 7–8, S2 includes Grade 9–10 and S3 includes Grade 11–12), adults and various. The adult category includes teachers, experts and administrators. The focus point of some studies are adults, and we didn't exclude them since our priority lies with charting the full breadth of theory and practice of embedding CT in K-12 education, and by excluding the papers including adults, we could have jeopardized that intention and that way compromised the objectivity of the selection process. On the other hand, this research doesn't have a core research question regarding the professional development of teachers, so we didn't include keywords about teachers or professional development in the search process.

3.8. *Research Methodology and Sample Size*

Research designs of the papers were classified according to Creswell's Educational Research Design classification (Creswell, 2012), since it provides different methods to conduct research to meet certain education-related research objectives. There are eight categories: experimental, quasi-experimental, survey, case, narrative, mixed method, action, and theoretical. If the research design of the paper was specified by the authors of the paper, the paper is categorized according to the authors' statement. If there is no description about research design, we have examined the methodology and data collection parts of the studies and classified the appropriate research design. Regarding the sample size, the studies with less than 30 participants, were categorized as small scale, those with 31 to 100 participants as medium scale, and those with more than 100 participants as being large-scale (Meulen *et al.*, 2021).

3.9. *Effectiveness of CT Integration Studies*

In order to investigate the cause-effect relationship in the studies and to broaden the evidence for the effectiveness of the studies, experimental and quasi-experimental studies were analyzed in this section. It is important to emphasize here that we consider non-experimental studies essential to the development of the field, but experimental research processes are usually well-structured and they include a control group which is crucial for the strong internal validity of studies (Creswell, 2012). This is in line with the aim of this section which tries to reflect unbiased reliable cause-effect relations of studies. There are 19 experimental and quasi-experimental studies. Of these studies, we included only studies (12) conducted with students in a formal education setting. We also excluded one study that did not specify the number of participants.

4. Findings

The main goal of this review study is to gain insights over exactly how CT is embedded into non-CS subjects in the curriculum, and how CT is taught and assessed to understand CT integration practices and gaps in the knowledge. In this section, we presented the findings related to the reviewed papers according to their learning objectives, integration levels, related subjects/courses, instructional strategies, technologies/tools, assessment goals and instruments, and finally participants.

4.1. *Target Subject*

To answer RQ1, we analyzed the subjects of the papers and learning objectives. Based on the analysis, the subjects that most frequently embedded CT are science (71) and mathematics (34). Social sciences (17), language arts (12) and art (9) are the least popular subjects among the embedded studies (Table 3).

Regarding the relation between the subjects and the grades of students (Table 3), mathematics is the subject where the primary school students are the largest group of participants. Science subjects are also very common for CT integration by all age groups, among which the largest group is secondary school students and adults (teachers, experts, and administrators). Social science subjects have most frequently been related to secondary school students. There are only 2 papers for preschool students and these papers focus on mathematics and science subjects.

With respect to the relation between the subjects and the dates of publication, we examined the papers published between 2006–2019. However, the first papers on CT embedded into non-computing subjects as reviewed in this study were published in 2009. As indicated in Table 3, most of the social science subjects-related papers were published in the recent years, 2017–2019. Also, the number of science and mathematics related papers has an increasing trend as well. For science subjects in particular, there has been a striking increase in the number of papers published between 2017 and 2019.

Table 3
Frequency of CT embedded studies by subjects*, grades** and publishing year

Category	Subject	PS	P	S	A	09–12	13–16	17–19	F	Total
Art	Art		2	4	2		3	4	7	9
	Music			2			2		2	
Science	Science	1	10	15	15	4	8	22	34	71
	Physics		4	9		1	7	6	14	
	Chemistry			4	1		3	3	6	
	Biology		7	9	1		8	9	17	
Mathematics	Math	1	16	8	6	2	10	17	29	34
	Geometry		3	3	1		2	3	5	
Language Arts	Language		3	3	2		4	3	7	12
	Journalism		1	1		1			1	
	Writing		3	2		2	1	1	4	
Social sciences	Social Science			6	3		2	6	8	17
	Geography		2	4	2	1		5	6	
	History		1	1			1		1	
	Philosophy			2	1		1	1	2	
Various	Multiple disciplines		1	2	6	1	4	4	9	9
Total									152	152

*PS: Pre-school, P: Primary, S: Secondary, A: Adults; publishing year 2009–2012, 2013–2016, 2017–2019.
**Some studies involve students at more than one education level, so there is a difference between grade total frequencies and year total frequencies.

4.2. *Learning Objectives*

To answer RQ1, we analyzed the learning objectives and the integration levels of studies by using adapted CT integration model (Fig. 2).

4.2.1. *CT-related Learning Objectives*

We classified CT-related LO's into three categories: concepts, practices and perspectives, and some of them include two or three of these dimensions. There were 42 studies (39%) which focus on **the concepts** dimension of CT. The benefits of a programming environment for learning mathematics were examined by Rodríguez-Martínez *et al.* (2020): they teach sequences, loops, events and conditionals as CT concepts. Similarly, Chang (2019) focused on the same CT concepts for digital storytelling activities. The example studies which focus on teaching different CT concepts in different subjects contained the following concepts: sequencing, loops, and conditional logic concepts in mathematics (Israel and Lash, 2019); variables, operators, control statements, loops, arrays, strings, functions in art-related examples (Jawad *et al.*, 2018); basic object-oriented programming concepts such as objects, classes and attributes in physics (Kanaki and Kalogiannakis, 2018). Twenty-one of 108 studies (19%) focus only on **the practices** dimension of CT. They examined how students learnt skills required to solve problems during the process of thinking such as algorithmic thinking and abstraction, confrontation (defining and understanding a problem encountered), decomposition, pattern recognition, abstraction, algorithm, automation, and analytical skills were analyzed in language arts (e.g., to read and understand basic Spanish words), geometry (e.g., to reason about two dimensional shapes and angles), financial arts (e.g., to define and calculate stock price factors such as current yield, total return) (Kale *et al.*, 2018); algorithmic thinking, decomposition, evaluation, abstraction and data representation skills were developed while exploring life on Mars in Science class, in which students design robots and test them using a simulated Mars environment (Baek *et al.*, 2019).

Forty studies (37%) focus on both concepts and practices together. One of these studies reports that students' skills related to code developing, modeling, and sequencing were improved while they were learning how energy is transferred from producers to consumers during science class (Lytle *et al.*, 2019). Another study showed that students develop a better understanding of force topics during physics course, while simultaneously acquiring CT concepts and practices (Hutchins *et al.*, 2018).

Only five studies (5%) focus on **concepts, practices and perspectives together**. Regarding the perspectives dimension, they examined how students develop understanding of themselves and their relations with the technological world. As an example, in the scope of a science course, the use of electronic textiles was examined for the introduction of the key computational concepts (input/output, digital/analog, variables, sequences, conditionals, loops, operators) and practices (reusing and remixing code, test and debug code) while broadening perceptions about computing (relevance of computing, ability to see oneself as a computer scientist) (Kafai *et al.*, 2014).

4.2.2. *Subject-related Learning Objectives*

In this section, CT embedded studies were analyzed in terms of subject-related LO's. They were grouped according to the subject areas: mathematics, science, language arts, social sciences, arts.

Mathematic and geometry related objectives and topics in CT embedded studies were summarized as follows: addition, fractions (Kong and Li, 2016); numbers, addition/subtraction, different calculation strategies (Djurdjevic-Pahl *et al.*, 2016; Messer *et al.*, 2018); numerical calculation, analytical reasoning, quantitative reasoning (Ke, 2014); coordinate graphing, mathematical functions (Friend *et al.*, 2018); calculating the least common multiple and the greatest common divisor (Rodríguez-Martínez *et al.*, 2020); identifying/creating geometric shapes, angles, symmetry and mirroring, functions and simultaneous equations, formulas for e.g. percentages, areas, calculator application (Niemelä *et al.*, 2017; Urban, 2015); cartesian coordinates, probability, measurement and number sense (Gadanidis *et al.*, 2017); operations, algebraic thinking, number and operations' counting, cardinality, fractions, the number system, expressions and equations (Harrison *et al.*, 2018); binary numbers and cartesian coordinates (Mensing *et al.*, 2013); shapes, angles, patterns, problem solving, symmetry, ratio (An and Park, 2012); shapes, addition and subtraction, mental math strategies, coordinate geometry, repeating patterns, fractions, area and perimeter relationships, symmetry and transformations, probability (Gadanidis, 2017); properties of polygons, fractions, multiplication, number sense through number stories, area and volume, algebraic thinking, operations (multiplication and division) (Israel and Lash, 2019); mathematical problems (Lévano *et al.*, 2016; Pires *et al.*, 2019); mathematical patterning (Miller, 2019); polygons and tessellations (Förster *et al.*, 2018; Foerster, 2016); fractions and ratios, geospatial concepts (coordinate estimation based on location) (Nugent *et al.*, 2009); two dimensional shapes (angles, areas etc.) (Kale *et al.*, 2018; Terwilliger *et al.*, 2019; Valentine, 2018).

Science and physics related objectives and topics in CT embedded studies were summarized as follows: simple electronic circuit, circuit diagram (Jacobson *et al.*, 2015; Kafai *et al.*, 2014; Litts *et al.*, 2017; Nugent *et al.*, 2019); forces and motion (velocity, speed, time, acceleration, vectors) (Basu *et al.*, 2018; Hutchins *et al.*, 2018); similarities and differences between digital and analog waves, wave amplitude and frequency, modern sonography (Lehmkuhl-Dakhwe, 2019; Towhidnejad *et al.*, 2014); types of machines, their characteristics and their uses in daily life (Pinto-Llorente *et al.*, 2018); weather prediction (temperature, wind, precipitation) (Yadav *et al.*, 2018); the electromagnetic forces, magnetic fields, electrical circuits (Gendreau Chakarov *et al.*, 2019); physics of flight and aircraft controls, unmanned aerial systems(Khan and Aji, 2018); gravity, forest fire (wind and tree density, fire prevention techniques) (Klopfer *et al.*, 2009); distance, measurement, gravity, center of mass (Krishnamoorthy and Kapila, 2016); mechanics, electromagnetism (Gero and Levin, 2019); calculating the gravitational constant, conservation of energy and momentum (Weintrop *et al.*, 2016); making children interested in physics, their familiarization with various physical phenomena (states of matter, some physical properties of matter and some physical laws) (Folgieri *et al.*, 2019); relative motion, accelerated motion (free fall), projectile motion, and a combination of projectile and relative motion (Lopez and Hernandez, 2015; Louca *et al.*, 2011); electricity unit (Garneli and Chorianopoulos, 2018; Hadad *et al.*, 2020); forces, motion, Newton's laws, vectors, physics concepts such as kinematics, and the difference between vectors and scalars (Liu *et al.*, 2014); kinematics (Basu *et al.*, 2014, 2015, 2016, 2017; Merkouris and Chorianopoulos, 2018) geology, meteorology, astronomy, and energy (Peel *et al.*, 2015); exploring life on mars (Baek *et al.*, 2019).

Biology related objectives and topics are as follows: protein synthesis translation process (Peel and Friedrichsen, 2018); natural selection (variation and differential survival)(Xiang and Passmore, 2015); movements of human heart, pulsimeter (Cakir and Guven, 2019); population dynamics and evolutionary change (Wagh *et al.*, 2017); ecosystem stability (Swanson *et al.*, 2018); cell (cell membrane, nucleus, cytoplasm, and vacuoles) (Leonard *et al.*, 2015); the spread of epidemic diseases (Cateté *et al.*, 2018); food webs (Lytle *et al.*, 2019); developing a heart rate remote monitoring system (Yu and Guo, 2018); epidemiology (spread of disease), ecology (I. Lee *et al.*, 2011); sequencing of genomes (Malyn-Smith *et al.*, 2018); life cycle of a tree (Israel *et al.*, 2015); the virus contagion model (which shows how a virus could spread, the forest fire model (which simulates forest fire), and the ecosystem model (which describes food pyramids in ecosystems) (Koh *et al.*, 2013); radioactivity, black holes and DNA sequencing (Orton *et al.*, 2016); assembling the full DNA sequence (Weintrop *et al.*, 2016); ecology (Basu *et al.*, 2014, 2015, 2016, 2017; Merkouris and Chorianopoulos, 2018).

Chemistry related objectives and topics are the following: periodic table (Boulden *et al.*, 2018); the structure of chemical compounds (Malyn-Smith *et al.*, 2018); matter (Peel *et al.*, 2015); pH neutralization of different water samples (Akbar *et al.*, 2018);exploring the relations between macroscopic properties of gases (pressure, volume, and temperature) (Weintrop *et al.*, 2016); ion transport across a cell membrane (Musaeus and Musaeus, 2019); the components of rocks and minerals in rock cycles (Dong *et al.*, 2019).

Language arts related objectives and topics in CT embedded papers were summarized as follows: reading and understanding basic Spanish words (Kale *et al.*, 2018); writing an essay (following directions and using literary elements), identifying symbols and explaining in a multi-paragraph essay how the symbols developed through the entire piece of literature, identifying elements of poetry (e.g., rhyme, repetition, etc.) in shorter pieces of literature (i.e., song lyrics) (Nesiba *et al.*, 2015); poetic thinking in English subject (Jenkins, 2015); reading of narrative and expository texts (Dong *et al.*, 2019); researching, developing and editing a story, refinement, fact-checking (Wolz *et al.*, 2011); blogging (Mensing *et al.*, 2013); storytelling (Chang, 2019; Parker, 2012); writing workshop sessions, defining a particular element of effective composition (such as characterization, foreshadowing, setting a scene) planning, drafting, revising, editing, publishing (Q. Burke, 2012).

Social sciences related objectives and topics in CT embedded studies were summarized as follows: digital humanities (Chen *et al.*, 2019); sustainability (Giordano and Maiorana, 2013); water wastage (Fronza *et al.*, 2016); weight issues (bullying due to their weight and appearance), energy wastage (unmonitored usage of appliances), the harmful effects of pollution (Noushad *et al.*, 2017); exploring life in the Middle Ages (Dong *et al.*, 2019); global social effects of climate changes (Weitze, 2017); geospatial concepts (coordinate estimation based on location) (Nugent *et al.*, 2009); ethical discussions (Fontenot *et al.*, 2013; Pardo, 2018); climate change and water shortage (Park and Park, 2018); local environmental and civic issues (Litts *et al.*, 2019).

Art related objectives and topics in CT embedded studies were summarized as follows: choreographing a dance (Leonard *et al.*, 2015); designing of e-textiles (Litts *et al.*, 2017); understanding artistic elements in paintings, learning biographical and historical contents of Spanish painters, increasing cultural and artistic competence to understand paintings, improving the ability to understand artistic expressions from different eras, analyzing historical and artistic content in paintings (Sáez-López *et al.*, 2016); weaving with a loom (V. R. Lee and Vincent, 2019); using music subject to motivate learning to program (Aaron *et al.*, 2017).

4.2.3. *Integration of CT and Subject related Learning Objectives*

The distribution of CT learning objectives by the subjects is presented in Fig. 3. The studies which include CT concepts (C) and practices (PR) are the ones which focus on programming concepts and problem-solving process together. The number of studies which include C & PR related LO's is the highest in science. As an interesting finding, the number of C-related studies in all other subjects except science is higher than C & PR related or PR related studies. This data can be interpreted as, apart from the science subject, mostly programming concepts were addressed in CT integration studies, and less emphasis was put on the problem-solving process (practices). However, this situation is different for science, as the number of studies including C & PR related LO's is higher than the number of studies in which only programming concepts were taught.

Another point related to the embedding of CT skills into subjects is teaching to use CT practices which focus on problem-solving process. The numbers of studies related to CT practices is lower than the number of C-related studies (Fig. 3). We presented the examples of how CT practices (Selby, 2013) can be applied and learned in context (Table 4). The common trend is that they are based on constructionist approach, students are active and try to build their own projects while investigating domain-specific content learning and also exercising CT practices. The point that makes the activities different from each other is related to the use of technology. They can be classified as no-tech, low-tech and high-tech activities. It means that students can learn CT without a tech/machine. All examples include a mental activity (CT practices) for formulating and solving a problem to find a solution, but the solution can be carried out by a human or machine, in line with the Wing's view of CT (Wing, 2006).

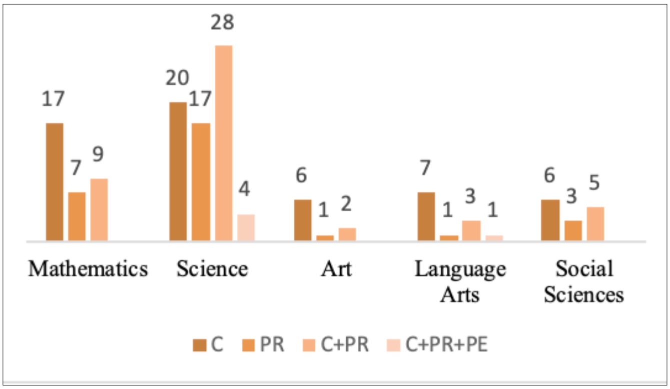

Fig. 3. CT-related Learning Objectives by Subjects.

Table 4

CT Practices by subjects in CT integration studies

Math	Science	Social Sciences	Language Arts	Arts & Music
Abstraction				
Students work on number puzzles and simple code crackers. The aim of both is to put numbers into a wider context, connecting them with language (letters, words) and art (shapes) (Djurdjevic-Pahl *et al.*, 2016).	Students recognize which resource from the lives caused CO2 to be produced from the most to the least (Park and Park, 2018).	The use of latitude and longitude to express locations demonstrates abstraction of the source data – that is, the physical characteristics of the location (say, a red cup placed under a tree) (Hammond *et al.*, 2019).	Students summarize a piece of literature. Students were provided a simple three-step: 1) Selection: Highlight or write down important sentences 2) Rejection: Discard the sentences that are not crucial 3) Substitution: Convert the highlighted sentences into your own words without altering the main ideas or introducing your own opinions and biases (Nesiba *et al.*, 2015).	Students generate their own designs for e-textiles, focusing first on their chosen aesthetic and later on the logistics of circuitry and coding (Kafai *et al.*, 2014).
Algorithmic Thinking				
Students develop algorithms to solve math problems (Pires *et al.*, 2019).	Within the game, students follow step-by-step procedures to conduct scientific experiments in virtual labs to measure pH in water and soil samples. Students follow instructions for flying a drone to embark on a virtual exploration of the region (Akbar *et al.*, 2018).	Geocaching is a form of a game, in which students must solve a problem (locating the assigned targets) by using the tools (GPS unit, list of tar-gets' coordinates, and their own geospatial understandings and orientation) for completing the target and arriving at the correct finish point. The entire process of navigating to the targets is an enaction of algorithmic control (Hammond *et al.*, 2019).	Students create a detailed step-by-step storyboard while illustrating of a scene of the play/story (Nesiba *et al.*, 2015).	Students choreographed and rehears-ed their cell- inspired dances in the physical realm with partners, they made their own choreographic notes on paper (Leonard *et al.*, 2015).
Decomposition				
Students decompose the mathematic problems to smaller parts (Pires *et al.*, 2019).	Related the water shortage topic, students decompose the problem into factors in saving water and discuss what kind of water resource there are and how much could save (Park and Park, 2018).	Teachers made a historical puzzle game for students to explore life in the Middle Ages. Students research events and note dates in the timeline, discover specific characteristics and break down the logical components in the puzzle games (Dong *et al.*, 2019).	Students write outlines of composition and identify arguments. Students analyze the context of a text by defining: the protagonist and antagonist; the setting of story; the conflict and the resolution (Dong *et al.*, 2019).	

Continued on next page

Table 4 – continued from previous page

Math	Science	Social Sciences	Language Arts	Arts & Music
Generalization				
Students assess different approach/ solutions to a mathematic problem (Weintrop *et al.*, 2016).	Students generalize code solutions for ion transport across a cell membrane (Musaeus and Musaeus, 2019).	Students learn to use latitude and longitude to express locations as they walk towards or away from the Equator. They can generalize to use latitude and longitude to express different locations (Hammond *et al.*, 2019).	Students relate the literary skills to what they are learning in other subject areas (Nesiba *et al.*, 2015).	Students weave with loom and draw woven pattern on paper (Lee and Vincent, 2019).
Evaluation				
Students debug and detect systematic errors while coding two-player fraction game (Kong and Li, 2016).	Related the water shortage topic, students test the efficiency of some natural materials for making rainfall or dirty one cleaner and choose most efficient materials in order to get the clean water (Park and Park, 2018).	Students test their products (a mobile app that shows how people use and waste water) and check whether the features behave correctly or not (Fronza *et al.*, 2016).	Students create digital stories as learning an effective composition. During the editing phase, many of the edits were spelling and grammar in characters' dialogue or trouble- shooting the programmed behavior of a coded sprite (Q. Burke, 2012).	Students iteratively test and debug their e-textile designs, developing solutions to address them (Kafai *et al.*, 2014).

4.3. *CT Integration Levels*

To answer RQ1, we classified the CT integration levels of studies according to adapted CT integration model. We categorized 40 studies as being at the "substitution" level (Fig. 4). In the substitution level studies, CT concepts exist and can be simply called out, but they may not involve direct engagement with subject related objectives or CT practices/perspectives. For example, Aaron *et al.* (2017) aimed to use Sonic-Pi for enabling students to learn programming by creating music. In this study, music is used to motivate students to learn programming, there is no specific explanation about musical LO's for CT integration.

We categorized twenty-eight of 108 studies as being at the "augmentation" level. For example, Burke (2012) aimed to teach a particular element of effective composition such

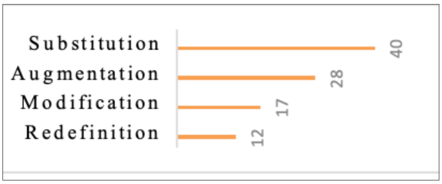

Fig. 4. CT Integration Levels of Studies.

as characterization, setting a scene and planning, drafting, revising, editing, publishing and programming concepts together. Students created stories while learning programming and improved their language related skills by establishing clear connection to the computing concepts.

We categorized seventeen of 108 studies as being at the "modification" level (Fig. 4). Wagh *et al.*(2017) asked students to explore NetLogo models of population dynamics and evolutionary change. Students engaged in guided investigations to develop questions using the models, develop hypotheses, run the models to make observations and record data, and formulate evidence-based explanations for their explorations. Each student created code modifications and explained how they expected code changes would impact model outcomes.

We classified twelve of 108 studies as being at the "redefinition" level. Xiang and Passmore (2015, p. 317) initiated student activities with a scenario, "According to biologists, the fur color of the ancestors of white polar bears was likely to be brown. These brown ancestral bears were gradually adapted into the arctic environment, which is often full of ice and snow, and evolved into white polar bears over time." They explained to the students that they should put all components of the natural selection model into their program procedures and then use their simulations to examine whether a group of brown ancestral bears would evolve into a group of white descendent bears (polar bears) over time. All students successfully constructed their simulation in this round with guidance from the instructors. In the second programming round, students were free to select the adaptation scenario in which they were most interested. For example, a group of students decided to simulate the adaptation of butterfly coloration. They redefined a computational model previously not available. Finally, we classified eleven of 108 studies as "Other", because most of them are theoretical studies that could not be associated with any of the four levels.

4.4. *Instructional Strategies*

To answer RQ2, we analyzed the instructional strategies (Fig. 5) and technologies/tools (Fig. 7). Some papers have used more than one instructional strategy or tool.

Experiential learning (25) and problem-based learning (24) are the most frequently used strategies. Examples of experiential learning strategy include: students created their own electric circuits with Arduino (Kafai *et al.*, 2014); students weaved with a loom and coded similar patterns with Scratch (Lee and Vincent, 2019); many CT related hands-on topics (plant irrigation or reading distance alarm) were taught in STEM education (So, 2018); teachers were encouraged to create learning activities that allowed students to practice coding (Dong *et al.*, 2019); students independently experimented with a robot and a visual programming environment to gain hands on learning experience in the STEM concepts (Krishnamoorthy and Kapila, 2016).

Out of 108 reviewed studies, 24 studies used a problem-based learning strategy include examples of students who focused on the development of understanding for fundamental circuitry and programming by using wearable textiles (Nugent *et al.*,

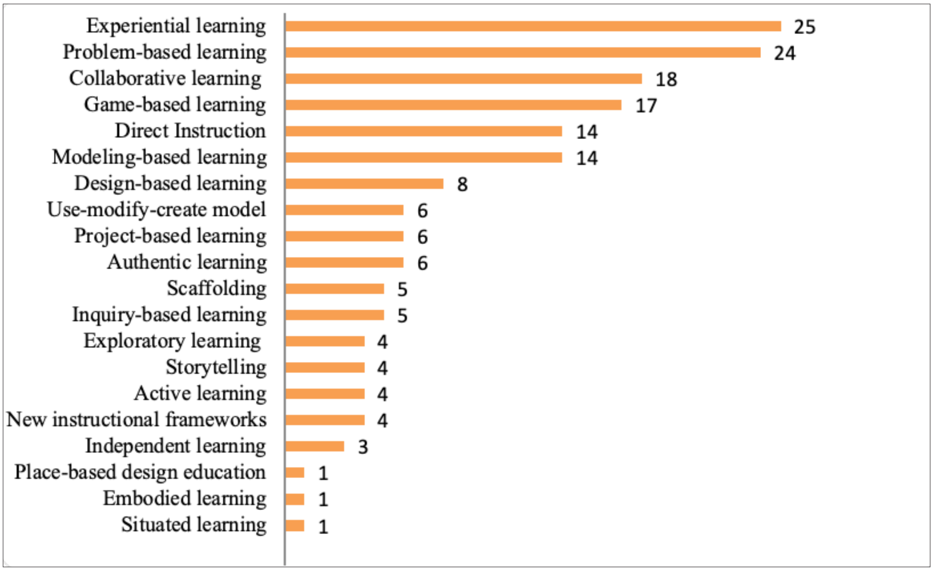

Fig. 5. Instructional strategies in CT integration studies.

2019); a basic framework was proposed for sequencing structure for learning and problem-solving activities in circuit activities and electricity models by using NetLogo tool (Jacobson *et al.*, 2015); students were familiarized with programming then students were guided through social problem solving framework to identify issues that affected them and they produced an app related to the problem by using App Inventor (Noushad *et al.*, 2017).

Eighteen studies used a collaborative learning strategy and nine of them also used it in conjunction with experiential learning. Seventeen studies used a game-based learning strategy, usually being based on game-making activities for improving subject related and CT related LO's. Fourteen studies used a direct instruction strategy partially, meaning that the direct instruction strategy was not used exclusively in any reviewed study. Direct instruction was only used together with other teaching strategies. For example, teachers' presentations, hands on activities, students' presentations, problem solving activities were used together for teaching physics of flight, aircraft controls and unmanned aerial systems via programmable robots (Khan and Aji, 2018); teachers applied teaching and didactic activities to activate previous knowledge of students and help students with difficulties in the development of each activity, teachers also created game samples, generated mathematical problems, and stimulated the competences of students (Lévano *et al.*, 2016).

A modeling-based learning strategy was used in fourteen studies (they were explained in detail in 4.4.1). Eight studies used design-based learning and in five of them it is related to game design (Garneli and Chorianopoulos, 2018; Harrison *et al.*, 2018; Ke, 2014; Kong and Li, 2016; I. Lee *et al.*, 2011). Use-modify-create model was used in six of studies (Baek *et al.*, 2019; Israel and Lash, 2019; I. Lee *et al.*, 2011; Lytle *et al.*, 2019; Malyn-Smith *et al.*, 2018; Musaeus and Musaeus, 2019), five of them were science related subject, and only one study is related to mathematic. A project-based learning strategy was used in six studies to teach LO's related to CT and subject (Almeida *et al.*, 2018; Fronza *et al.*, 2016; Koh *et al.*, 2013; Sáez-López *et al.*, 2016; Urban, 2015; Yu and Guo, 2018). The instructional strategy of six studies was based on authentic learning (Amador and Soule, 2015; Noushad *et al.*, 2017; Orton *et al.*, 2016; Parker, 2012; Towhidnejad *et al.*, 2014; Yu and Guo, 2018). Scaffolding strategy/approach were used in five studies explicitly (Basu *et al.*, 2016, 2018; Cateté *et al.*, 2018; Freudenthal *et al.*, 2013; Hammond *et al.*, 2019). An inquiry-based learning strategy was applied in five studies (Aaron *et al.*, 2017; Burgett *et al.*, 2015; Peel *et al.*, 2015; Wagh *et al.*, 2017; Yu and Guo, 2018). Among the reviewed studies, there were four different instructional frameworks. These are the EIMA instructional framework (Lehmkuhl-Dakhwe, 2019), the 5E learning model (Cakir and Guven, 2019), the 4C framework (Pinto-Llorente *et al.*, 2018) and Zones of Proximal Flow strategy (Basawapatna *et al.*, 2013).

In Table 4, we presented sample studies related to the teaching of CT practices, and in this section, we have explored in more detail the instructional strategies used to teach different CT learning objectives (concepts, practices, perspectives) (Fig. 6). It is observed that reviewed studies preferred to use different type of instructional strategies to teach different type of CT learning objectives. As seen in Fig. 6, partially "new and innovative teaching approaches" such as use-modify-create (83%), modeling-based learning (79%), inquiry-based learning (60%) were mostly used in teaching CT practices which focus on problem solving skills (such as abstraction, decomposition, algo-

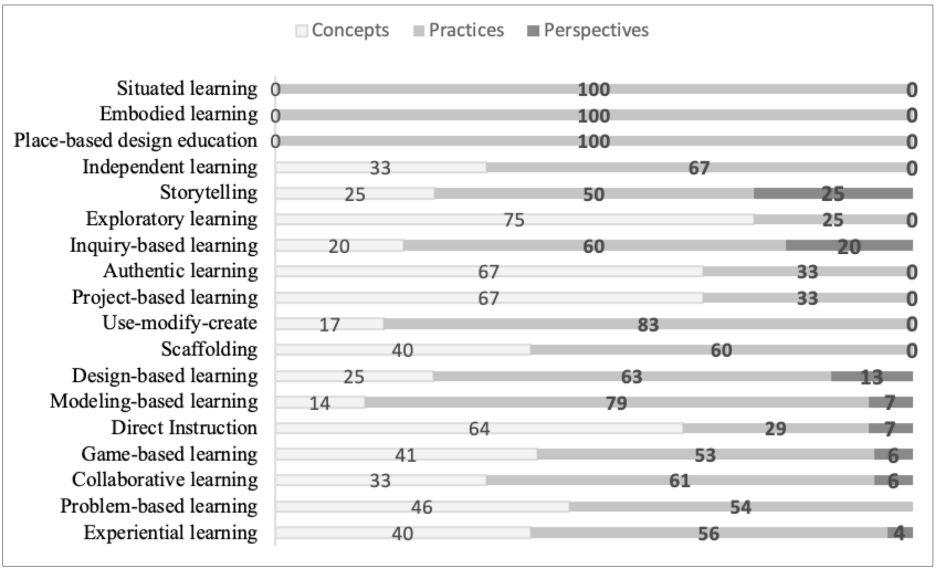

Fig. 6. Percentages of Instructional Strategies Used to Teach Different CT Goals.

rithmic thinking, evaluation and generalization etc.). On the other hand, better-known and more traditional teaching methods such as direct instruction (64%), project -based learning (67%), etc. were mostly used to teach CT concepts which focus on programming knowledge (such as sequences, loops, events, conditionals etc.), since probably teachers are already familiar with these instructional approaches. In addition, it is seen that subject teachers mostly use exploratory learning (75%) while teaching CT concepts, and it brings to mind the idea that the lack of programming/CT knowledge of teachers can affect their instructional strategy choices.

4.4.1. *Instructional Strategies Used in Higher Transformation Level CT Integration*

In order to understand higher-level CT integration practices, we investigated the transformation level CT integration studies (Fig. 2) which mainly focus on problem-solving process (CT Practices) instead of programming concepts. We examined the instructional strategies and approaches of 26 study in this category.

Out of 26 studies, more than half of them (16) used "**model-based learning**" or "**use-modify-create approach (UMC)**" to teach CT integrated subject goals. While teaching domain subject (kinematics, ecology, epidemic diseases, etc.) students engaged in model building, simulation, model checking and verification activities (Basu *et al.*, 2017; Cateté *et al.*, 2018; Hutchins *et al.*, 2018; Louca *et al.*, 2011). First, students read domain specific sources, summarized them and found common/different properties of data (abstraction), and second, students observed the model behavior in the form of simulations to explore the deep domain specific problems such as effect of environmental factors on disease spread or ion transport across a cell membrane and then modify or create their own models. Model-based learning is usually combined with UMC approach and students build CT and content knowledge by using, modifying, and creating code in the models (I. Lee *et al.*, 2011; Lytle *et al.*, 2019; Malyn-Smith *et al.*, 2018; Musaeus and Musaeus, 2019). From the perspective of the teacher's role, subject teachers taught unplugged activities related with the domain and for plugged activities, the research team aided in terms of technology/CT to help the teacher (Cateté *et al.*, 2018), teachers were trained on the plugged materials (Lytle *et al.*, 2019) or research team developed the plugged materials (models/simulations/ games) by themselves in collaboration with subject teachers (Basu *et al.*, 2017).

Out of 26 studies, almost half of them (12) used "**scaffolding approach**" together with other instructional strategies to deal with the open-ended nature of learning environment which makes it hard for students to interpret all data and achieve the learning goals (Basu *et al.*, 2016, 2017, 2018; Hammond *et al.*, 2019; Lytle *et al.*, 2019). To address this challenge, the scaffolding approach is an effective instructional approach that involves providing support to learners as they work on CT tasks that are initially beyond their current level of understanding/skill. They used different scaffolding approaches such as breaking down tasks, providing examples, guided practice and feedback. For example, in a language arts lesson, students were supported with a structured template to analyze a poem using CT practices such as decomposition, pattern recognition, and abstraction. The template included prompts such as "Identify the repeating patterns in the poem," "Break down the poem into its individual components," and "Identify the overarching themes and concepts present in the poem."(Nesiba *et al.*, 2015). Also, modeling-based learning was used together with the scaffolding teaching approach to teach the epidemics curriculum focused on modeling the spread of epidemic diseases such as the flu by using modeling tools such as Cellular (Cateté *et al.*, 2018). Additionally, in one of the reviewed experimental studies, it's found that students who received scaffolding built more accurate models, used modeling strategies effectively, adopted more useful modeling behaviors, showed a better understanding of important science and CT concepts, and transferred their modeling skills better to new scenarios (Basu *et al.*, 2017).

Out of 26 studies, nine of these transformation level CT integrated studies used "**collaborative learning**" explicitly and encouraged collaboration among students to support each other's learning. Peer support was used to help the students with different level of understanding on CT, as students can share their understanding and skills with each other. For example, in a geography class, students were involved in a scaffolded geocache game, in which students must solve a problem (locating the assigned targets) by using the tools (GPS unit, list of targets' coordinates, and their own geospatial understandings and orientation) to reach a win state (completing the target and arriving at the correct finish point) and students were encouraged to peer-to-peer collaboration to support each other's learning, and they were advised to decompose the task by having one student focus on latitude and the other focus on longitude (Hammond *et al.*, 2019). Musaeus *et al.* (2019) described that agent-based modeling is a valuable way for students to learn CT in different subjects, and furthermore students worked in pairs to support peer-to-peer collaboration. In another study, students worked in groups to develop their models, which subsequently underwent a whole class collaborative evaluation two or three times. From those evaluations, students took ideas about how to improve their models and went back to their small groups to implement the desired changes (Louca *et al.*, 2011).

4.5. *Technologies and Tools*

To answer RQ2, we analyzed the technologies and tools used in the reviewed studies (Fig. 7). From the 108 papers, we identified the most frequently used technology category as block-based programming (BBP) tools with 57 papers in total. Two main block-based programming tools are Scratch (34) and App Inventor (8) in this review study. Other BBP tools used were: Alice (2), Blockly (2), Snap (2), TurtleArt (1), Stagecast Creator (1), Looking Glass (1) and six studies reported as BBP/ visual programming without specifying any tool. Physical devices/ robotics (27) are the second most frequently used tools. Six of the studies reported Arduino as technology used and another six of them reported Lego. Seven studies reported robotics without specifying any tool. The other eight studies included different physical devices such as a GPS unit, Kinect, e-textiles, Makey Makey, graphic calculator, Ozobots, sphero robotic ball and a robotic puppet. We observe that text-based programming (TBP) tools are not commonly used in the CT embedded studies. We found only ten studies using text-

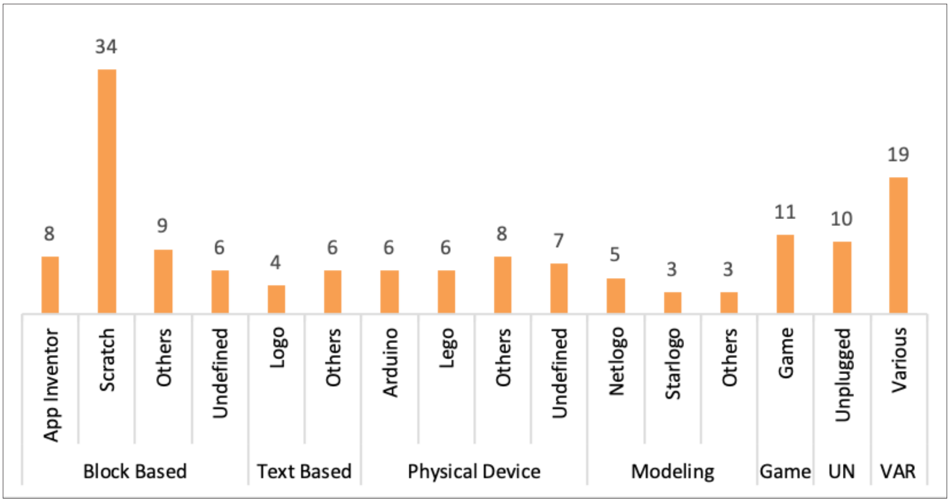

Fig. 7. Frequency of Technologies/Tools in CT studies.

based programming tools. These are Logo (4), Python (2), HTML/CSS (1), Sonic Pi (1) and studies reported as TBP without specifying any tool (2). Eleven of CT studies used modeling tools for CT embedding: NetLogo (5) and StarLogo (3), Cellular (2) and SURGE Gameblox (1). Eleven studies used games (e.g., video games, puzzles etc.). Ten studies were classified as "unplugged" and 19 as "various". Some example tools in the "various" category are Google Documents, Spreadsheets and ToonDoo, and the new platforms such as Moral machine platform (MIT), CTSIM, C2STEM, DISSECT (Folk *et al.*, 2015).

We also analyzed the frequency of the use of specific technologies for particular age groups (Fig. 8). Results show that the block-based programming is the most popular technology/tool used by all target groups except preschool. The largest group was sec-

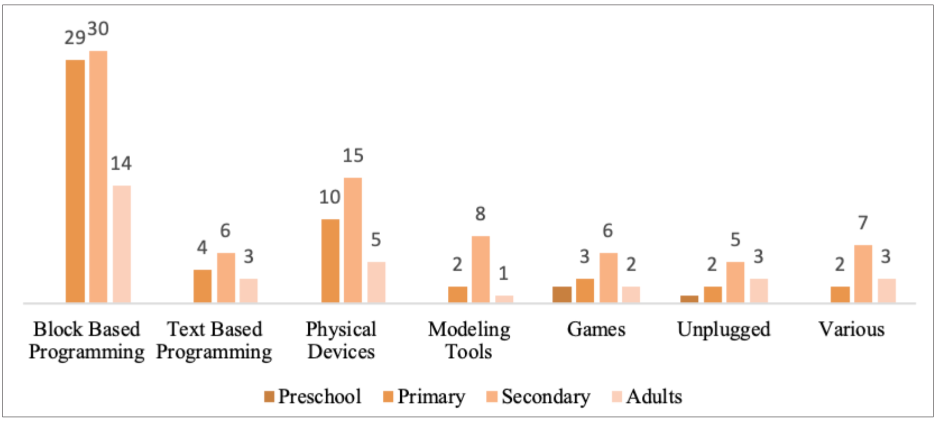

Fig. 8. The relationship between age and technology/tool.

ondary school students, followed by primary school students and adults. We observed that the text-based languages, which are thought to be more difficult to learn, were also used by primary school students.

4.6. *Assessment Strategies*

To answer RQ3, we analyzed the assessment strategies. Eighteen of the 108 papers did not report any assessment instruments. Only six of the 108 papers explicitly report performing an assessment of CT in the integrated lessons, it refers to the overall approach used to evaluate student learning. The other 84 papers reported using assessment instruments for the purpose of data collection. A number of papers reported using more than one instrument, resulting in a total of 186 instruments (Fig. 10). Fig. 9 presents the numbers of studies reporting assessment for programming comprehension and CT skills, subject related goals, and attitude and satisfaction – as well as numbers of combinations of any two of these goals and all three of them – regarding a total of 80 studies. Finally, Table 5 presents the numbers of assessment instruments encountered per assessment goal.

Formative and summative assessment: Apart from data collection instruments, only six of the reviewed papers explicitly report performing assessment of CT in the integrated lessons. Basu *et al.*(2015) developed a modeling-based learning environment fo-

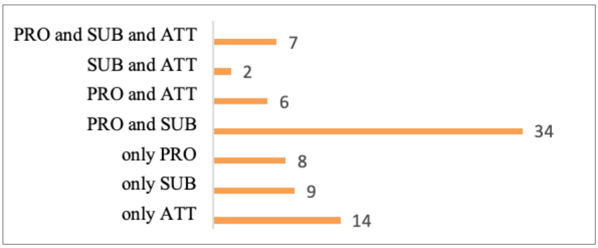

Fig. 9. Number of studies per assessment goal
(ATT: Attitude, SUB: Subject related, PRO: Programming/CT related).

Table 5
Numbers of assessment instruments per assessment goal

	Tests (Quizzes, surveys)	Performance Tasks (Homework, projects, portfolios)	Self-Assessment (Self-reflection, interviews, observations)	Assessment rubrics	Automated Assessment
Programming Comprehension/CT skills	26	27	29	7	4
Subject related objectives	26	24	28	6	1
Attitude/Satisfaction	21	0	14	2	0
Others	9	4	27	1	0

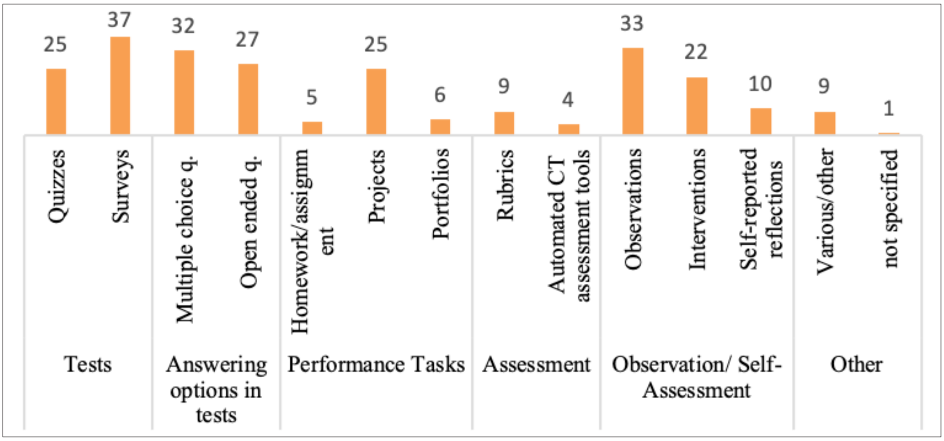

Fig. 10. Number of assessment instruments combination of assessment goals.

cused on specific science topics, and accompanying **multiple choice quizzes** to test the learning of these science concepts and CT skills. Then, they developed a collaborative learning-by-modeling environment for high school physics classrooms and accompanying **rubrics** to assess the synergistic learning of physics and CT concepts and practices through **observations and analysis of students' projects** in addition to multiple choice quizzes. They provide students with formative assessments tasks too (Hutchins *et al.*, 2018). Nesiba *et al.* (2015) used **surveys** to evaluate students' understanding of CT concepts after a lesson unit integrating teaching of CT and English literature. Louca *et al.* (2011) constructed a framework to analyze and evaluate subject related concepts and CT aspects of computational models of physics phenomena constructed by students. Grgurina *et al.* (2018) **provide a generic practical assignment and accompanying rubrics** based on SOLO taxonomy (Biggs and Tang, 2011) to assess the development and use of agent-based models of phenomena from various disciplines within secondary education CS course. Finally, So (2018) reports assessing **the projects** made by primary teachers where they use microcontrollers for STEM experience; however no details of the assessment are provided. Finally, it can be summarized that multiple forms of assessment (including tests, performance tasks, observations, rubrics, surveys etc.) were used together to assess multiple dimensions of CT and it helps to provide a more comprehensive picture of students' learning.

Programming comprehension and CT skills: 55 studies with 93 data sources focus on the assessment of the learning of these skills. In 29 instances, assessment is based on **observations, interviews, self-reported reflection** or other sources, sometimes in combination with each other or with the analysis of projects or portfolios. These forms of assessment are used, for example, to "elicit student thinking processes when approaching an assigned problem" (Cateté *et al.*, 2018), for the analysis of moments of notice (Hadad *et al.*, 2020), to have learners themselves describe what they had learned (Klopfer *et al.*, 2009), to observes the participants engagement with the design process and their reflection upon it (Litts *et al.*, 2017), or to examine teachers' views on integration of CT (Rich *et al.*, 2019; Yadav *et al.*, 2018). Secondly, in 27 instances, courses used **performance tasks** (including homework/assignments, projects or portfolios). Examples include a debugging assignment (Kafai *et al.*, 2014) or just a suggestion for an assignment (Kong and Li, 2016); projects where, for example, the functionality of mobile apps produced by students was checked (Noushad *et al.*, 2017) or to assess both students' learning and the effectiveness of the project itself (Urban, 2015). Finally, this assessment goal includes 26 instances of **tests (including quizzes and surveys)**: for example, to assess the learning of CT concepts (Burgett *et al.*, 2015; Rodríguez-Martínez *et al.*, 2020), through e.g. examination of algorithms generated by students (Basu *et al.*, 2015) or having students report their improvement in programming skills themselves (Giordano and Maiorana, 2013).

Subject matter skills: 52 studies used 85 data sources to examine the gains in learning of the subject matter. Twenty-six of these data sources were **quizzes and surveys**, for example to assess the mathematical abilities, spatial awareness and working memory (Messer *et al.*, 2018), or to test how students use and understand models (Musaeus and Musaeus, 2019). Eleven of these studies report using multiple choice questions to test a single science concept (Basu *et al.*, 2015), coordinate graphing, spatial skills, and functions (Friend *et al.*, 2018) or English reading (Nesiba *et al.*, 2015). Nineteen of these studies report using open-ended questions: for example, to test the "understanding of science concepts and CT skills as well as the ability to solve problems by combining multiple fundamental concepts" (Basu *et al.*, 2015), or to assess geometry learning (Förster *et al.*, 2018).

There were twenty-four cases of **performance tasks (including assignments, project and portfolios)** reported. They were assessed, for example, by evaluating the models (Louca *et al.*, 2011); through the use of Bag of Words using a "vector-distance metric to measure the dissimilarity between a student's model and the expert model" (Basu *et al.*, 2015); field observations, video recordings, artifact analysis and interviews (Q. Burke, 2012); self-reported reflection in the form of science diaries (Cakir and Guven, 2019); day summaries (Harrison *et al.*, 2018); students' notes (Leonard *et al.*, 2015); student-generated diagrams and handwritten comments (Litts *et al.*, 2017); or students' experiences during lessons (Peel and Friedrichsen, 2018); through debugging assignments (Kafai *et al.*, 2014) or evaluation of written essays (Nesiba *et al.*, 2015). In one case, an observation of **a group discussion** was used to assess the gains in learning of subject matter (Leonard *et al.*, 2015). To assess the gains in learning of subject matter, **rubrics** were used five times (Cakir and Guven, 2019; Chang, 2019; Grgurina *et al.*, 2018; Hutchins *et al.*, 2018; Terwilliger *et al.*, 2019); and **an automated CT assessment tool** once to assess not only the gains in learning of subject matter, but program comprehension and CT skills as well (Q. Burke, 2012).

Subject matter skills in combination with program comprehension and CT skills: 41 studies examine both the gains in learning of subject matter skills and the learning of CT skills or program comprehension (Basu *et al.*, 2015; Cateté *et al.*, 2018; Chang, 2019; Dong *et al.*, 2019; Grgurina *et al.*, 2018; Harrison *et al.*, 2018; Hickmott *et al.*, 2018; Israel and Lash, 2019; Jaipal-Jamani and Angeli, 2017; Jenkins, 2015; Kafai *et al.*, 2014; Ke, 2014; Klopfer *et al.*, 2009; Lai and Lai, 2012; Leonard *et al.*, 2015; Litts *et al.*, 2017; Louca *et al.*, 2011; Merkouris and Chorianopoulos, 2018; Miller, 2019; Moreno-León and Robles, 2016; Musaeus and Musaeus, 2019; Nesiba *et al.*, 2015; Nugent *et al.*, 2019; Orton *et al.*, 2016; Pardo, 2018; Peel and Friedrichsen, 2018; Rich *et al.*, 2019; Rodríguez-Martínez *et al.*, 2020; Swanson *et al.*, 2018; Terwilliger *et al.*, 2019; Towhidnejad *et al.*, 2014; Urban, 2015; Valentine, 2018; Wagh *et al.*, 2017; Weitze, 2017; Xiang and Passmore, 2015; Yadav *et al.*, 2018; Yu and Guo, 2018).

Attitude and satisfaction: Twenty-nine studies used 37 assessment instruments to examine the attitude and satisfaction of the study participants. Examples include exploring (increased) interest in and perception of STEM and CS (Burgett *et al.*, 2015; Kafai *et al.*, 2014; Urban, 2015) and interest to pursue a CS degree (Jawad *et al.*, 2018); affective aspects such as enjoyment in the activity (Djurdjevic-Pahl *et al.*, 2016), motivation (Lévano *et al.*, 2016; Nugent *et al.*, 2009), ownership (Lytle *et al.*, 2019), perceived confidence (Lehmkuhl-Dakhwe, 2019; Wolz *et al.*, 2011) and difficulties (Almeida *et al.*, 2018); and finally, the attitude toward the camp they participated in (Khan and Aji, 2018). Regarding the attitude and satisfaction of the teachers, they were asked about their difficulties (Lytle *et al.*, 2019), attitude on CT in teaching their (other) discipline (Gadanidis *et al.*, 2017) and the quality of the teaching program (Liu *et al.*, 2014). Seven studies report measuring the combination of learning gains in subject matter, programming comprehension and CT skills, and attitude and satisfaction as well (Q. Burke, 2012; Kafai *et al.*, 2014; Ke, 2014; Nugent *et al.*, 2009, 2019; Orton *et al.*, 2016; Urban, 2015).

Other aspects: Thirty-four studies report measuring other aspects, for example beliefs and practices within the project (Aaron *et al.*, 2017), reflection and feedback on user interface or teaching materials (Akbar *et al.*, 2018; Cateté *et al.*, 2018), personal experiences and reflections (Boulden *et al.*, 2018), evaluation of lesson plans (Dong *et al.*, 2019; Israel and Lash, 2019), assessing collaboration (Fields *et al.*, 2015), working memory (Messer *et al.*, 2018), teachers' perspectives on integration of CT (Rich *et al.*, 2019; Yadav *et al.*, 2018) and teachers' experiences during lessons (Peel and Friedrichsen, 2018; Urban, 2015; Yu and Guo, 2018). Ten of these studies report measuring these other aspects exclusively (Akbar *et al.*, 2018; Boulden *et al.*, 2018; Folgieri *et al.*, 2019; Freudenthal *et al.*, 2013; Gadanidis, 2017; Goncharenko *et al.*, 2019; Hestness *et al.*, 2018; Israel *et al.*, 2015; V. R. Lee and Vincent, 2019).

4.7. *Educational Stages of Participants*

To answer RQ4, we analyzed educational stages of participants (Fig. 11). Fifty-eight (45%) of the papers have participants in secondary school level; 43 (34%) of the papers have participants in primary school level. Adults were involved in 25 of 108 studies (19%), and although adults are mostly teachers, there are also studies with experts and administrators. The most frequently included target audience grades in the CT context stud-

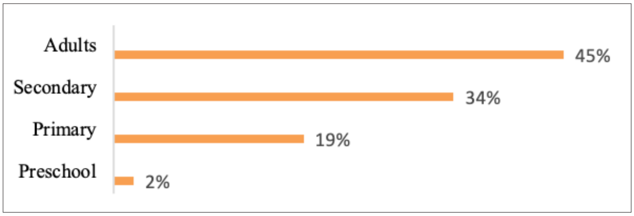

Fig. 11. Educational stages of participants.

ies are the last grades of primary school / Grade 5–6 (28) and the first grades of secondary school/ Grade 7–8 students (23). The least frequently mentioned grades of participants for CT integration studies are preschool (2) and the last grades of secondary school /Grade 11–12 (6). In four studies, only primary school level (P1&2) was stated and no grade or age was specified. Similarly, in 16 studies only secondary school level (S1&2&3) was cited, no grade/age was specified. Finally, in 4 studies, only K-12 level was cited.

4.8. *Research Methodology and Sample Size*

To answer RQ5, we analyzed the research methods (Fig. 12) and scales of the papers (Fig. 13). Out of the 108 studies reviewed, we categorized 39 studies as a case study. Common examples of case studies are teachers'/students' experiences with CT integration to different subjects (Basu *et al.*, 2018; Parker, 2012; Yadav *et al.*, 2018) or a

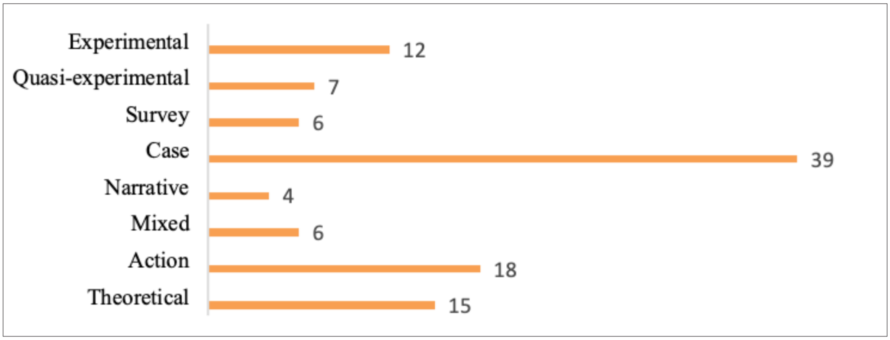

Fig. 12. Frequency of research methods.

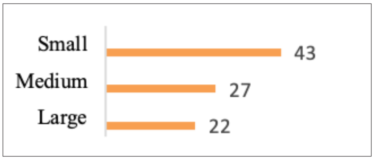

Fig. 13. Sample size of studies.

description of CT integrated a new workshop/course/activity (Cakir and Guven, 2019; Fronza *et al.*, 2016; Pardo, 2018; So, 2018). Observations (Cateté *et al.*, 2018; Hadad *et al.*, 2020), interviews (Boulden *et al.*, 2018; Rich *et al.*, 2019; Wagh *et al.*, 2017) and artifact analyses (Chang, 2019; Dong *et al.*, 2019; Litts *et al.*, 2017) were the main data collections forms in these qualitative studies.

We categorized nineteen studies as experimental and quasi-experimental studies; an intervention was implied in these studies and the most common experimental design found were time-series experiments (within group designs) in CT integration studies. For example, Jaipal-Jamani and Angeli (2017) explore the effect of robotics on elementary preservice teachers' CT skills by implementing one group pre-test post-test design in their experimental study. In quantitative designs, researchers collect data related to "performance of the CT/Subject" or "attitude towards CT" or "CT self-efficacy" by using achievement tests, questionnaires or checklists (Messer *et al.*, 2018; Pires *et al.*, 2019; Rodríguez-Martínez *et al.*, 2020; Wolz *et al.*, 2011).

We categorized twenty-four studies as combined research design: mixed method (6) and action research (18) designs were conducted in these studies. Sixteen theoretical studies contain new or established abstract principles related to a specific field of knowledge. They do not contain research or present experimental data. For example, Lehmkuhl-Dakhwe (2019) suggest an instructional framework for integrating computing into science context in 4th–12th grades. Similarly, Ragonis (2018) describe an academic course for pre-service teachers for facilitating their students' learning CT process in the context of their own discipline. In another theoretical study, Weintrop *et al.* (2016) proposed a definition of CT for mathematics and science in the form of a specified taxonomy. A model for CT integration in third grade geography lesson was suggested by Hammond *et al.* (2019). Noa (2018) developed CT integrated course content and lesson materials particularly for K-12. Greenberg (2017) provided a theoretical information to applying the Pythagorean Theorem to robot building.

Out of the 108 studies, six are literature review studies related to CT integration: Scratch tools for teaching physics (Lopez and Hernandez, 2015); robotic kits for learning STEM disciplines (Sullivan and Heffernan, 2016); games for integrating CT into science (Clark and Sengupta, 2019); integrating CT with a Music context (Bell and Bell, 2018); CT for US Common Core Standards in language, art and mathematics context (Mensing *et al.*, 2013); and a systematic review of CT in secondary education (Lockwood and Mooney, 2017).

Regarding the sample size (scales) of papers, 43 papers included less than 30 participants (small scale), 27 papers included between 31 and 100 participants (medium scale), and 22 papers include more than 100 participants (large scale). 16 papers did not mention the sample size (Fig. 13).

4.9. *Effectiveness of CT Integration Studies*

To answer RQ6, we analyzed the effectiveness of 12 experimental and quasi-experimental studies in this section (Table 6), as follows:

Table 6

Effectiveness of Experimental and Quasi-experimental studies

Ref	Grade	Scale	Subject	Subject LO	CT LO	Intervention / Ins Str / Tool	Assess Tools	Results
Quasi Experimental								
(Rodríguez-Martínez *et al.*, 2020)	6th	EG=24 CG=23	Math	The least common multiple and the greatest common divisor	Sequences, iteration or loop, event handling and conditionals	Problem based learning EG: They used Scratch as a tool to solve the different tasks proposed. CG: They worked in a classic paper- and-pencil environment.	The CT Test (pre-post) Mathematical Knowledge Test (MKT)(pre-post)	+ No differences in CT were detected between the EG and CG. + No significant differences were shown in the CG between MKTpre and MKTpost. + On the other hand, the tests revealed a significant improvement in the EG between MKTpre and MKTpost.
(Lytle *et al.*, 2019)	6th	EG=95 CG=65	Science	Food Webs: students learn about how energy is transferred from producers to consumers.	Modelling, sequential conditionals, control condition, developing code	EG: Use-modify-create strategy CG: Create-create-create strategy	Student-perceived difficulty (scale), Student-perceived ownership teacher-perceived difficulty, Class observations	+UMC sequencing provides students a natural progression to learn CT within a science course, while giving students more ownership over the artifacts they create. +Teachers told that UMC curriculum was easy to teach, and that it promoted student engagement and exploration.
(Nugent *et al.*, 2019)	6th-8th	EG:808 CG:618	Science	Circuitry design, microcontrollers, electronic textiles, engineering design	Problem solving and the interdisciplinary thinking, programming	EG: The e-textiles instruction CG: No treatment	Knowledge of circuitry design, programming and engineering design Self-efficacy in making a wearable e-textile product	+The wearable technology instruction resulted in significantly higher scores in student knowledge of programming and circuitry, as well as STEM self-efficacy, compared with a control group.
(Pires *et al.*, 2019)	5th	EG:14 CG:14	Math	Mathematic problems	decomposition, pattern, abstraction, and algorithm	EG: Gamified didactic activities CG: (not mentioned)	Mathematical performance test	+The EG had a significant improvement in mathematical performance after the applications of gamified didactic activities involving CT. +The rate of learning evaluation in the test group is higher than the control group.

Continued on next page

Table 6 – continued from previous page

Ref	Grade	Scale	Subject	Subject LO	CT LO	Intervention / Ins Str / Tool	Assess Tools	Results
(Garneli and Chorianopoulos, 2018)	8th	Simu-lation Gr=17 Video game Gr=17	Science	Electric Circuit	CT concepts, practices and perspectives	Simulation group: They represented the function of a basic electric circuit by creating a simulation. Video game group: They represented the same function by creating a video game.	CT skills pre-post test CT concepts were examined by Dr Scratch Student motivation Observation and informal interviews with students	+ Video game construction resulted in projects with higher CT skills and more primitives, as measured through projects' code analysis. + The video-game context seems to better motivate students for future engagement with computing activities.
(Lai and Lai, 2012)	5th	96	Science	Observation of the Sun and The Weather	Logical thinking, problem solving	Students were encouraged to engage in science content and think about how they can represent this specific science content in a Scratch.	Logical thinking test (pre-post) Problem solving test (pre-post) Subject learning questionnaire (post)	+The results of the logical thinking test and the problem-solving test indicate that 5th graders had better performance in logical thinking and problem-solving skills by designing Scratch. +The students' Scratch projects indicate that students were able to represent their understanding of the science concepts cold front and warm front.
(Sáez-López *et al.*, 2016)	5th and 6th	EG:107 CG:32	Art History	Artistic elements in paintings, biographical and historical contents, cultural and artistic competence	CT concepts (sequence, looping, conditions, etc.) CT practices (experimentation and iteration)	EG: Project based learning using Visual programming language (Scratch) CG: (not mentioned)	Visual Blocks Creative Computing Test (to measure computational concepts and computational practices.) Questionnaire and structured observation (to measure students' learning and attitude)	+An active pedagogical approach using a Visual Programming Language (Scratch) significantly improves several elements: learning programming concepts, logic, and computational practices. +Students and observers point out that working with visual programming through projects provides fun, motivation, enthusiasm, and commitment from the student.
Experimental								
(Basu *et al.*, 2017)	6th	EG=52 CG=46	Science	Kinematics and ecology	CT concepts, Modelling and programming concepts	EG: CTSIM with adaptive scaffolding modelling. Students build and test their models. CG= CTSIM without adaptive scaffolding model	Kinematic and Ecology test (pre-post) CT skill test include questions to predict program outputs and develop algorithms. Modeling performance metrics	+ Students who received scaffolding built more accurate models, used modelling strategies effectively, adopted more useful modelling behaviours, showed a better understanding of important science and CT concepts, and transferred their modelling skills better to new scenarios.

Continued on next page

Table 6 – continued from previous page

Ref	Grade	Scale	Subject	Subject LO	CT LO	Intervention / Ins Str / Tool	Assess Tools	Results
(Jenkins, 2015)	8th	EG=13 CG=14	English	Poetic thinking	Sequences, loops, events	EG: Microworld based intervention: Poem generator activity	CT pre-post test Poetic thinking pre-post test	+Learners who received the microworld-based intervention in teaching practice made a higher improvement in CT and poetic thinking than their counterparts who did not.
(Messer *et al.*, 2018)	KG (5- 6 age)	41	Math	Simple mathematical calculations-counting, addition, subtraction	Sequence of several commands, paths, maze solving and program debugging	1st group: Ipad Programming group (Bee-bot ipad app) 2nd group: Unplugged programming group (pictures of bee-bot levels)	Number skills math test (pre-post) Computation math test (pre-post) Spatial awareness test. (pre-post)	+All three groups increased their mathematical abilities and spatial awareness. + There were no significant increases in working memory, and
(Moreno-León and Robles, 2016)	4th and 5th	4th EG=13 4th CG=15 5th EG=19 5th CG=18	English	Vocabulary and grammar exercise from regular English books	• Abstraction • Parallelism • Synchronization • Logical thinking • Flow control • User interactivity • Data representation	EG: Programming activities with scratch CG: Traditional /without programming	English knowledge pre-posttest Motivation survey Source code analyzes	+The results show that the groups working with programming activities improved academic results and motivation more than the groups using traditional resources. +Most students felt that coding was a positive influence, not only for learning English, but for developing other important skills as teamwork and learning to learn. +CT skills are developed with maturity (5th grade students' development higher than 4th grade students)
(Nesiba *et al.*, 2015)	12th	EG=66 CG=42	English	Writing an essay, identify symbols, identify elements of poetry, write critical and comparative analyses of literature	Algorithms (spatial reasoning), Abstraction (data analysis, analogy, and prompt summarization)	EG= DISSECT lesson modules (Discovering Science through CT Project) (Google documents, ToonDoo, Blogger etc.) CG= Control group (no information)	CT test with multiple choice questions Subject performance test (include reading test questions and essay writing)	+The students in the DISSECT classes performed better on the CT assessment than their peers in the control classes. +Regarding the reading section, all classes performed approximately the same, on the writing section the DISSECT classes scored higher than the control classes.

- Almost all experimental studies compared the effect of different media/technology on CT and the subject knowledge of students (such as “simulation versus video game” or “Scratch versus unplugged”). Further, some of them investigate whether there are differences between CT integrated lessons (using active learning strategies such as project-based learning with specified media such as Scratch) and traditional lessons (not specified as instructional strategies or media). The effect of the two different instructional strategy was compared in only one experimental study (use-modify-create versus create-create-create).
- Half of the studies (6) are small-scale studies where the number of students in the experimental group (EG) is less than 30, the average number of participants in EG is around 15–18. Almost all studies (10 out of 12) were conducted with students in the middle school grades (5th–8th). There is only one experimental study with 12th grade students (Table 6). Experimental studies with early elementary and high school students are absent.
- No experimental studies on social sciences have been found. Except for three language arts and one art history study, other studies (8) are related to science or mathematics subjects.
- Almost all of the studies (10) have set learning goals related to both CT concepts and CT practices (modeling, logical thinking, debugging, abstraction, decomposition, etc.).
- As an assessment tool, CT knowledge and subject knowledge were measured using pretests and posttests consisting of multiple-choice questions. In addition, some studies include variables such as motivation, attitude, or perceived difficulty measured by questionnaires and observations.
- The students in the experimental groups generally made better progress in terms of CT knowledge, subject knowledge, and motivation compared to the control groups (traditional education) in studies where CT was integrated into subject content using different technologies. In only one study (scratch vs. unplugged) (Rodríguez-Martínez *et al.*, 2020), there was no significant difference between the EG and CG in terms of CT.

5. Conclusion and Discussion

This section presents the key findings and discussion from the analysis of 108 papers published between 2006 and 2019, focusing on the integration of computational thinking in non-computing school subjects. The research questions addressed various aspects, including learning objectives and CT integration levels, instructional strategies and technologies/tools used, assessment strategies, educational stages of participants involved, research methodologies, and the effectiveness of CT integration.

5.1. *Learning Objectives and CT Integration Levels*

With regard to RQ1, the findings reveal that science subjects and especially mathematics dominate the integration of CT, with arts and humanities catching up in recent years. This indicates a growing interest in integrating CT into a broader range of disciplines. However, there is a need to develop more mature CT integration practices in other subjects. Notably, mathematics, including geometry, appeared to have a strong association with CT, likely due to the affinity between mathematical thinking and computational thinking, such as problem-solving, logical reasoning, and innovative thinking ability (Kallia *et al.*, 2021). Additionally, the early development of educational STEM activities that combined programming, computational modeling, and mathematics (Rodríguez-Martínez *et al.*, 2020) might have contributed to this prevalence. The advent of physical devices like the Arduino platform and block-based programming tools further bolstered the integration of CT in mathematics and science classrooms. Encouragingly, we observed a growing trend of CT integration in arts, humanities, and social studies, indicating a broader interest in extending CT to diverse disciplines. The successful integration of CT into social studies and other disciplines can improve the curriculum while also verifying Wing's argument that CT is everywhere and for everyone.

While examining the timeline of CT integration studies, we noted that the first papers on CT integration into non-computing subjects were published in 2009, three years after Wing's seminal paper introducing computational thinking (Wing, 2006). This suggests that the field of CT integration studies might have initially lagged behind CT studies in general. However, this gap has been closing over time, with an increasing number of publications on the subject. This trend aligns with the concept that CT is best learned in context (Grover, 2018), integrated into class subjects. Moreover, the emerging studies provide substantial evidence of the potential benefits of CT integration and its positive impact on curriculum enhancement.

An essential observation pertains to the specific focus of CT learning objectives in the reviewed studies. The prominence of studies emphasizing programming concepts alone indicates a preference for the technical aspects of CT rather than broader skills related to problem-solving processes and perspectives on the technology world. However, when examining science subjects, we noticed a greater emphasis on teaching both concepts and practices related to CT. This might indicate a higher level of maturity in the integration of CT within science disciplines. Additionally, the prevalence of assessment tools that primarily focus on computational concepts, overlooking practices and perspectives, further reinforces the need for a more holistic approach to CT assessment.

When analyzing the integration levels of CT studies, we found that a significant portion (40%) of them remained at the substitution level, where CT concepts were merely mentioned without direct engagement with subject-related objectives or CT practices and perspectives. This could suggest that the field of CT integration is still relatively immature, and there is a room for growth and development. Encouragingly, some studies reached the redefinition level, the highest stage of integration, allowing students to create their computational products or solutions to address subject-related or real-world problems. The new CT integration model with different CT integration levels through different integration perspectives (Malyn-Smith *et al.*, 2018; Tress *et al.*, 2005; Waterman *et al.*, 2020; Weintrop *et al.*, 2016) might help to develop higher transformation level (redefinition) integrated lessons in the future studies.

5.2. *Instructional Strategies and Technologies/Tools*

In relation to RQ2, the analysis of instructional strategies highlights the prevalence of experiential learning, problem-based learning, and collaborative learning, which align well with the constructivist and active learning approaches. Our results appear consistent with Hsu *et al.* (2018) who reported a meta-review for CT studies for 2006–2017 and found the most frequently used instructional strategies as project-based learning, problem-based learning and collaborative learning for CT instruction.

Notably, more than half of the higher transformation level CT integration studies employed model-based learning and use-modify-create approaches to teach CT integrated subject goals. These approaches involved students in model building, simulation, model checking, and verification activities, fostering a deeper understanding of both CT and the subject matter. Additionally, scaffolding and collaborative learning were frequently used alongside other instructional strategies to support students in the open-ended learning environments. These strategies appeared particularly beneficial in addressing the complexities of CT integration into non-CS subjects. These results might provide useful guidance for designing effective instructional practices to integrate CT into non-CS subjects.

An important observation relates to subject teachers' self-efficacy in programming, which seemed to influence their instructional strategy choices. The prevalence of exploratory learning approaches, particularly in teaching CT concepts, might reflect teachers' comfort with encouraging students to experiment with the learning process independently. This underscores the significance of providing professional development programs to support non-CS subject teachers in developing their understanding of computational thinking and effective integration practices. In the literature, several studies have shown that subject teachers often lack the required knowledge and skills to integrate CT effectively into their curriculum (Yadav *et al.*, 2016). Therefore, professional development programs are essential to help teachers gain the necessary knowledge and skills (Grover and Pea, 2018).

In recent years, from 2017 to 2019, the adoption of 18 different learning strategies for CT integration indicates an increasing interest in exploring diverse approaches to enhance students' performance in both subject-related and CT-related areas. Notably, efforts to develop CT-specific instructional frameworks and strategies, such as the Use-Modify-Create strategy (Lytle *et al.*, 2019), EIMA instructional framework (Lehmkuhl-Dakhwe, 2019), the 5E learning model (Cakir and Guven, 2019), the 4C framework (Pinto-Llorente *et al.*, 2018), demonstrate a growing commitment to refining and innovating instructional practices.

The analysis of technologies and tools used in CT integration studies revealed a prevalence of block-based programming and physical devices (including robotics). The popularity of block-based programming can be attributed to its user-friendliness and affordability, making it accessible to learners with varying levels of programming experience. The visual nature of block-based languages also attracts students by eliminating the need to focus on syntax, thereby promoting engagement and creativity, alongside promoting and developing CT (Kafai, 2016; Wei *et al.*, 2021). Additionally, modeling tools emerged as the third most frequently used category in secondary education, suggesting the positive contribution of CT to learning specific disciplinary concepts (Grgurina *et al.*, 2018).

Surprisingly, unplugged activities and text-based programming languages were used less frequently than block-based programming and physical devices. The limited use of text-based languages might be attributed to perceived complexity due to syntax rules. However, the development of easy-to-learn text-based gradual programming languages, such as Hedy (2021), offers promising prospects for addressing this challenge. It is also important to highlight the increasing recognition of unplugged activities as valuable means for developing computational thinking skills in students, as evidenced in recent research studies (Kuo and Hsu, 2020).

5.3. *Assessment Strategies*

With regard to RQ3, although only six studies directly reported assessments as formative or summative, most of the assessment stemmed from data collection methods, either qualitative or quantitative. This indicates the need for more dedicated efforts in developing standardized, valid, reliable and comprehensive measurement tools for assessing CT skills comprehensively across different contexts and settings. The scarcity of such assessment tools makes it challenging to compare and evaluate students' CT skills consistently. In a literature review about CT assessment (Poulakis and Politis, 2021) it was found that CT assessment faces unresolved issues, including the inability to cover all concepts and student age groups, the lack of scientific documentation and validation, and the dearth of tools to autonomously and efficiently assess CT. Furthermore, Tang *et al.* (2020) indicated that more reliability and validity evidence regarding CT assessment tools needs to be collected and reported in the future studies.

Self-assessment tools, performance tasks, and tests/quizzes (in frequency order) were the most commonly used instruments for evaluating CT learning objectives. Self-assessment, in particular, proved effective in assessing students' CT skills by encouraging them to reflect on their abilities and application of CT skills (Brennan and Resnick, 2012; Mendoza *et al.*, 2016). **Performance tasks** involved students in applying CT skills to solve subject-related problems, promoting deeper engagement with the subject matter studies (Barr and Stephenson, 2011; Kafai and Burke, 2017). **Tests and quizzes**, often used as summative evaluation tools, focused on assessing students' technical skills or conceptual understanding of CT. Similarly, Tang *et al.* (2020) found that traditional tests and performance assessments are often used to assess CT skills.

While self-assessment, performance tasks, and tests/quizzes are valuable assessment tools, they alone may not provide a comprehensive evaluation of students' computational thinking skills. CT skills **encompass multiple dimensions**, such as problem-solving, algorithmic thinking, and abstraction, thus making comprehensive assessment challenging (Martins-Pacheco *et al.*, 2019). **The combination of different assessment and data collection methods** was common in reviewed studies, aiming to present a more holistic picture of students' learning outcomes.

In light of the assessment goals, it is evident that CT integration studies often do not focus on evaluating programming comprehension and CT skills, subject-related objectives, or students' attitude and satisfaction. A significant number of studies emphasized **enhancing positive attitudes and satisfaction towards CT or STEM** disciplines, indicating the importance of fostering students' motivation and interest in the field. However, the relative lack of emphasis on developing standardized and reliable assessment instruments for CT-related and subject-related learning objectives are indicative of the immaturity of this field of study and the need for greater focus on comprehensive assessment practices.

5.4. *Educational Stages of Participants*

In relation to RQ4, looking at the student target groups in the reviewed studies, we see that while even preschool students are involved, the primary school students get a lot of attention, and secondary school students even more. There is a growing trend in the number of studies by starting with smaller age groups, a peak half way through K-12 and then a decline towards the end of secondary education. The most frequently included target audience in the CT integration studies are the last grades of primary school (Grade 5–6) and the first grades of secondary school (Grade 7–8). Another target group represented well – mentioned in 23% of the studies – consists of teachers, experts and administrators. They are the ones tasked with planning and implementing the teaching of the relatively new idea of CT in context.

5.5. *Research Methodology and Sample Size*

With regard to RQ5, we see that while the number of the papers is increasing rapidly, most of the research is still explorative and concerns small or medium scale studies. Almost half of the reviewed studies were labeled as small scale and were thus conducted with less than 30 participants. In another review study related to CT integration in mathematics, it is indicated that there are also small-scale research designs on self-reported attitudes or beliefs (Hickmott *et al.*, 2018). A large number of reviewed studies (67 out of 108) employ qualitative or mixed methods research design. There are relatively few theoretical studies which contain new or established abstract principles related to a CT integration field and they do not contain research or present experimental data. Furthermore, there are only 19 experimental and quasi-experimental studies. To identify the cause and effect of hypothesis and determine more in-depth ideas, it is revealed that experimental studies regarding CT integration are absent.

5.6. *Effectiveness of CT Integration Studies*

In relation to RQ6, to understand the effectiveness of CT integration, we analyzed the results of experimental and quasi-experimental studies. We found that almost all experimental studies compared the effect of different media/technology on CT and the subject knowledge of students. Moreover, there is a lack of experimental studies which compare the effectiveness of various instructional strategies. In 1983, Clark published a meta-analysis that examined the influence of media on learning and according to him, the media/technology is not the message, it is merely vehicle whereby instruction is delivered and the influence comes from instructional strategies. So, it is considered that more experimental studies exploring the instructional strategies of CT integration are needed to contribute to the field. Additionally, the students in the experimental groups generally made better progress in terms of CT knowledge, subject knowledge, and motivation compared to the control groups (traditional education). In only one study (scratch vs. unplugged comparison) (Rodríguez-Martínez *et al.*, 2020), there was no significant difference between the experimental and control groups in terms of CT knowledge. Furthermore, it is found that there is no reviewed experimental study regarding CT integration into social science subject.

5.7. *Limitations of Research*

Conducting a systematic literature review is a rewarding endeavor as well as a huge undertaking. Conducting a SLR on CT is a race against time – the number of studies published keeps growing with increasing speed and the temptation to keep looking and adding new studies to the collection is always present. Therefore, we decided to draw a line at the end of 2019. (A number of papers have 2020 as the publication date due to being published online prior to appearing in print.) Another issue is the exercise in fine balance when deciding on search strategy. Authors could have used other terminology than what we expected and we could have missed some papers. To counter that threat, after several trials and much deliberation, we decided to cast a wide net and find a large number of papers, reasoning that it was better to invest time in sifting through a lot of papers than to risk missing the relevant ones. Finally, concerning the quality of the papers we found, we decided to include all of the papers that met our inclusion criteria. For a number of papers, we had to check whether the journals where they were published were really peer reviewed, and if that was the case, we included those papers. We decided to do so, because our priority lies with charting the full breadth of theory and practice of embedding CT in context, and by excluding the papers not meeting the highest quality standards, we could have jeopardized that intention and that way compromised the objectivity of the selection process. Related to coding procedure, there might be potential discrepancies in the interpretation of instructional strategies/or research methodologies among the authors of different papers. We minimize discrepancies and ensure a unified understanding regarding the instructional approach/research method employed by (1) including peer-reviewed articles (peer review process help authors of reviewed papers to strive for a more consistent and comprehensive representation of their research method), (2) discussing the coding scheme to reach consensus among researchers, (3) seeking scientific literature to interpret instructional strategies/research methods, and (4) providing references of chosen instructional strategies/methods.

6. Research Implications

Based on the findings of this study, considering the state of the field and identified gaps in the knowledge of how to embed CT in the school curriculum, we put forward the research implications for CT practitioners and some suggestions for future CT researchers.

6.1. *Research Implications for CT Practitioners*

We made the following remarks as instructional design suggestions for CT integrated lessons to K-12 teachers, curriculum developers, school administrators and teacher educators.

- To embed the learning of CT with the learning of subject matter successfully, more emphasis needs to be placed on the problem-solving aspect of CT. 40% of the reviewed studies focused on programming concepts as CT LO's. By using problem-solving as the focus, more teachers might be motivated to embed subcomponents of computational thinking in their regular academic subjects (Yadav *et al.*, 2016). While focusing on the problem-solving process it is important to (1) express the original problem in computational terms, (2) construct an executable computational solution and (3) interpret the computational solution in terms of the original subject matter (Kallia *et al.*, 2021).
- Higher transformation level (redefinition or modification) integrated CT lessons might allow students to create computational products/solutions to solve disciplinary related problems. The adapted CT Integration Model (Fig. 2) can provide different perspectives for teachers to see different CT integration levels through different integration perspectives (disciplinary approach, STEM integration, modeling etc.).
- Our results support the need to form the content of the lesson cohesively with a clear storyline for embedding computing ideas and principles into the subject ar-

eas. We observed that in some reviewed CT integration studies, clear instructional goals related to subject matter were absent, and instead, the subject matter was used as a motivation tool (Aaron *et al.*, 2017). On the other hand, some reviewed studies missed defining CT goals of the lesson explicitly (Förster *et al.*, 2018; Giordano and Maiorana, 2013). We believe that it is important to decide on/define how CT enhances the disciplinary learning, and how disciplinary learning uses/employs/provides context for CT.

- Employing various student-centered active learning strategies might help to enhance the integration of CT skills and subject skills. Problem-based learning, experiential learning and collaborative learning are most frequently used strategies for CT integration into context (Fig. 5). These learning strategies allow learners to actively engage in authentic problem-solving tasks, apply computational strategies to solve these problems, and collaborate with others to build and share knowledge.
- Model-based learning, use-modify-create, scaffolding approach and collaborative learning were found to be effective in building higher transformation level integration of CT into context. In transformation level studies, teachers engaged students in model/simulation using, adapting, building and verification activities to deepen in the subject related goals/problems.
- Open-ended nature of CT integrated learning environments might help to engage students to develop several CT dispositions such as dealing with complexity, persistence in working with open-ended problems, working with peers to achieve a common goal or solution. The scaffolding approach is an effective instructional approach by providing support to learners as they work on open-ended nature of CT tasks, which used frequently in higher transformation level CT integration studies. For example, in social science context, while students are learning about different forms of government, teachers can scaffold CT skills by providing a comparison chart or graphic organizer that prompts students to analyze and compare the different forms of government based on criteria such as power distribution.
- Focusing on unplugged activities and pedagogic aspects might help teachers to gain a deeper understanding of how computational thinking can be integrated into various subjects. However, based on the results, it is important to note that the focus of many reviewed experimental studies has been on the effectiveness of specific media or tools, such as block-based programming tools or physical devices, rather than on pedagogy itself. Specifically, it is important to shift the focus from viewing technology as a tool to viewing it as an integral part of the subject field, where its use is informed by a deeper understanding of the subject matter itself (Grover and Pea, 2018; Vallance and Towndrow, 2016). Additionally, the unplugged activities provide an easier and less intimidating entry point to CT because it leverages students' everyday language and makes explicit connections between intuitive thinking and algorithm concepts. Understanding al-

gorithmic logic before plugging in should help students and teachers feel more comfortable and confident (Peel *et al.*, 2022).

- Professional development programs can be structured to ensure that teachers effectively integrate computational thinking in their classrooms. It has been observed that teachers' pedagogical knowledge related to CT affects their teaching strategies since teachers who tried to teach programming/CT concepts have directed students to learn by self-discovery, or used instructional strategies such as explorative learning (Fig. 6). Additionally, in many reviewed studies, subject teachers taught unplugged activities related to the domain and for plugged activities, the research team aided the teacher for CT content (Cateté *et al.*, 2018), teachers were trained on the plugged materials (Lytle *et al.*, 2019). Other studies also suggested that many educators have inadequate knowledge about what CT skills are and a lack awareness of how these skills can be utilized in their classrooms (Yadav *et al.*, 2016).
- To evaluate multiple dimensions of CT, a skill might be required to use a combination of various assessment methods together, including self-assessment, performance-based assessments and tests (Table 5). This provides a more complete picture of students' CT skills and can help mitigate some of the limitations of individual assessment methods (Margulieux *et al.*, 2019). We believe that it is helpful to be aware of multiple forms of assessment to evaluate students' products and progress toward subject-related and CT-related learning objectives. When students engage in problem-solving activities to learn CT concepts and practices, it is important that the assessment aligns with the nature of this instructional approach. Rather than simply testing knowledge and the application of specific concepts, the assessment can focus on students' ability to utilize the CT concepts learned in a problem-solving context. One potential tool that can serve this purpose is short tasks (e.g. Bebras challenge) (tailored to the specific subject/domain), which scaffolding problem-solving activities that require the application of CT skills (Dagiene and Dolgopolovas, 2022)
- Automated assessment tools and rubrics might gain more attention to assessing CT skills. In the reviewed studies, automated assessment tools and rubrics are least preferable data collection tools for evaluating CT skills. We believed that it is crucial to increase the number of automated assessment tools to save time and provide more accurate and consistent assessments. On the other hand, assessing CT skills can be challenging because of the uniqueness of student solutions since there can be multiple valid solutions to a problem. To deal with this issue, it is recommended to use rubrics with clear criteria and examples to guide evaluators in assessing CT skills.
- The recommendations regarding assessment are based on the data collection procedures of the studies, which is thought to be insufficient in providing a practical holistic perspective for teachers regarding assessment. It can be assumed that teachers need more practical solutions/approaches that they can use during the real lesson, apart from the research setting.

6.2. *Suggestions for CT Researchers*

We make the following suggestions for future CT studies.

- Across the entire curriculum (particularly social science, language arts, art etc.), there is an obvious need to identify the subject-related LO's which can benefit from the embedding of CT. More than two-thirds of the reviewed studies include mathematics and science related LO's and there is no reviewed experimental study on CT integration into social science subjects. We can get inspired by state-of-the-art practices in industry and academia. It is important to provide more opportunities for embedding of CT into the social sciences, language arts and art subjects. Another intriguing and crucial subject that has to be addressed is how far CT integration can alter the subject related content of what is taught (curriculum), rather than how it is taught.
- Based on the results, the reviewed experimental studies mainly focused on the effectiveness of media/tool instead of instructional strategies. It is critical to conduct more experimental research on the comparison of different instructional strategies to understand the effectiveness of different teaching approaches according to the cognitive ability of students with different age. Also, the duration and who conducted CT activities (teacher, IT coordinator, researcher) can be analyzed and these aspects can contribute to a more comprehensive understanding of effective CT integration in educational settings.
- In the reviewed studies, assessment purpose of studies focused on the enhancement of positive attitudes and satisfaction toward CS or STEM disciplines and there are very few studies focusing on the assessment of CT skills in the integrated courses. Therefore, there is a lack of evidence related to valid and reliable assessment instruments for the assessment of learning gains regarding CT-related LO's. There is a need to conduct more studies to develop valid and reliable holistic assessment instruments in order to assess all three aspects of CT (concepts, practices, perspectives).
- One of the least frequently mentioned educational stages of participants for reviewed CT integration studies are the upper grades of secondary school. It is important to conduct research studies including students from the upper grades of secondary education since their curriculum is focusing on the in-depth study of a smaller number of disciplines. Their cognitive abilities also differ accordingly to other grades and it might be helpful to examine the adequate CT teaching strategies for these grades.
- Most of the research designs are still explorative (case studies) and concern small-scale studies. There is a need to conduct large-scale and experimental research to understand more in-depth ideas regarding the cause-effect relations in the field of CT integration into context and help to generalize the findings to larger populations. Also, it is recommended to work on the development of solid theoretical underpinnings for embedding CT in other disciplines, both for specific school subjects and generic which apply across various disciplines.

- Overall, in industry and academia, any form of meaningful work, let alone cutting-edge achievements, are unimaginable without the aid of specific software, for example, modeling, data analysis, machine learning, or simply for project management. Yet, while similar, simplified, and user-friendly software is increasingly available, its potential to help students improve their understanding of the subject matter is hardly ever reached. In light of these instructional design tips, we recommend to modernize all aspects of curricula and to conduct the research necessary to facilitate essential changes and improvements.

7. Acknowledgments

We would like to thank the authors of the reviewed studies included in this study. Their dedicated research and findings formed the foundation of our analysis, and we are indebted to their valuable contributions to the field.

8. Funding

This research received funding from the Netherlands Initiative for Education Research (NRO) which is a unit of the Netherlands Organization for Scientific Research (NWO), the Dutch Research Council (Grant number: 40.5.18540.153).

References

We have added an asterisk (*) to 108 reviewed papers in the reference list.

* Aaron, S., Blackwell, A., Burnard, P. (2017). The development of Sonic Pi and its use in educational partnerships: Co-creating pedagogies for learning computer programming. *Journal of Music, Technology and Education*, 9(1), 75–94. https://doi.org/10.17863/CAM.8369

* Akbar, M., Dura, L., Gates, A.Q., Ortega, A., Roy, M.K., Santiago, C., Tellez, J.G., Villa, E. (2018). *Sol y Agua: A Game-based Learning Platform to Engage Middle-school Students in STEM*. In 2018 *IEEE Frontiers in Education Conference (FIE)* (pp. 1–9)

* Almeida, R., Gomes, A., Bigotte, M.E., Pessoa, T. (2018). IProg: Getting started with Programming: Pilot experiment in two elementary schools. *2017 International Symposium on Computers in Education, SIIE 2017*, *2018-January*, 1–5. https://doi.org/10.1109/SIIE.2017.8259651

* Amador, J.M., Soule, T. (2015). Girls Build Excitement for Math from Scratch. *Mathematics Teaching in the Middle School*, 20(7), 408–415. https://doi.org/10.5951/mathteacmiddscho.20.7.0408

* An, J., Park, N. (2012). Computer Application in Elementary Education Bases on Fractal Geometry Theory Using LOGO Programming. *This Paper Is Extended from a Conference Paper Presented at the Journal of Korean Institute of Information Technology*, 9(8), 241–249. https://doi.org/10.1007/978-94-007-2598-0_26

Anderson, J.R., Reder, L.M., Simon, H.A. (1996). Situated learning and education. *Educational Researcher*, 25(4), 5–11.

Aydeniz, M. (2018). Integrating Computational Thinking in School Curriculum. In M. S. Khine (Ed.), *Computational Thinking in the STEM Disciplines: Foundations and Research Highlights* (pp. 253–277). Springer International Publishing. https://doi.org/10.1007/978-3-319-93566-9_13

* Baek, Y., Wang, S., Yang, D., Ching, Y.-H., Swanson, S., Chittoori, B. (2019). *Revisiting Second Graders' Robotics with an Understand/Use-Modify-Create (U[superscript 2]MC) Strategy*. *European Journal of STEM Education,* 4(1).

Barcelos, T.S., Munoz, R., Villarroel, R., Merino, E., Silveira, I.F. (2018). Mathematics learning through computational thinking activities: A systematic literature review. *Journal of Universal Computer Science*, 24(7), 815–845.

* Barr, V., Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, 2(1), 48–54. `https://doi.org/10.1145/1929887.1929905`

Barron, B.J., Schwartz, D.L., Vye, N.J., Moore, A., Petrosino, A., Zech, L., Bransford, J.D. (1998). Doing with understanding: Lessons from research on problem-and project-based learning. *Journal of the Learning Sciences*, 7(3–4), 271–311.

* Basawapatna, A.R., Repenning, A., Koh, K.H., Nickerson, H. (2013). The Zones of Proximal Flow: Guiding Students Through a Space of Computational Thinking Skills and Challenges. *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, 67–74. `https://doi.org/10.1145/2493394.2493404`

* Basu, S., Biswas, G., Kinnebrew, J., Rafi, T. (2015). Relations between modeling behavior and learning in a Computational Thinking based science learning environment. *Proceedings of the 23rd International Conference on Computers in Education, ICCE 2015*, 184–189. `https://www.scopus.com/inward/record.uri?eid=2-s2.0-85039921425&partnerID=40&md5=abfcc7ed678c49f8861bd94466bee9f4`

* Basu, S., Biswas, G., Kinnebrew, J.S. (2017). Learner modeling for adaptive scaffolding in a Computational Thinking-based science learning environment. *User Modeling and User-Adapted Interaction*, 27(1), 5–53.

* Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J.S., Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning*, 11(1). `https://doi.org/10.1186/s41039-016-0036-2`

* Basu, S., Kinnebrew, J.S., Biswas, G. (2014). Assessing student performance in a computational-thinking based science learning environment. *In Intelligent Tutoring Systems: 12th International Conference, ITS 2014, Honolulu, HI, USA, June 5–9, 2014. Proceedings 12 (pp. 476–481). Springer International Publishing.*

* Basu, S., McElhaney, K.W., Grover, S., Harris, C.J., Biswas, G. (2018). A principled approach to designing assessments that integrate science and computational thinking. *Proceedings of International Conference of the Learning Sciences, ICLS*, *1*(2018-June), 384–391.

Bell, J., Bell, T. (2018). Integrating Computational Thinking with a Music Education Context. *Informatics in Education*, 17(2), 151–166.

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

Biggs, J., Tang, C. (2011). *Teaching for quality learning at university*. McGraw-Hill International.

* Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., Punie, Y. (2016). Developing computational thinking in compulsory education. *European Commission, JRC Science for Policy Report*.

* Boulden, D.C., Wiebe, E., Akram, B., Aksit, O., Buffum, P.S., Mott, B., Boyer, K.E., Lester, J. (2018). Computational Thinking Integration into Middle Grades Science Classrooms: Strategies for Meeting the Challenges. *Middle Grades Review*, 4(3), n3.

Brennan, K., Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada*.

Brown, J.C. (2017). A Meta-synthesis of the Complementarity of Culturally Responsive and Inquiry-Based Science Education in K-12 Settings: Implications for Advancing Equitable Science Teaching and Learning. *Journal of Research in Science Teaching*, 54(9), 1143–1173.

* Burgett, T., Folk, R., Fulton, J., Peel, A., Pontelli, E., Szczepanski, V. (2015). DISSECT: Analysis of pedagogical techniques to integrate computational thinking into K-12 curricula. *Proceedings – Frontiers in Education Conference, FIE, 2014*. `https://doi.org/10.1109/FIE.2015.7344241`

* Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education*, 4(2), 121–135.

Burke, W.Q. (2013). *Coding & composition: Youth storytelling with scratch programming*.

Burleson, W.S., Harlow, D.B., Nilsen, K.J., Perlin, K., Freed, N., Jensen, C.N., Lahey, B., Lu, P., Muldner, K. (2018). Active Learning Environments with Robotic Tangibles: Children's Physical and Virtual Spatial

Programming Experiences. *IEEE Transactions on Learning Technologies*, 11(1), 96–106. https://doi.org/10.1109/TLT.2017.2724031

* Cakir, N.K., Guven, G. (2019). Arduino-Assisted robotic and coding applications in science teaching: Pulsimeter activity in compliance with the 5E learning model. *Science Activities*, 56(2), 42–51.

* Cateté, V., Lytle, N., Dong, Y., Boulden, D., Akram, B., Houchins, J., Barnes, T., Wiebe, E., Lester, J., Mott, B., Boyer, K. (2018). Infusing Computational Thinking into Middle Grade Science Classrooms: Lessons Learned. *Proceedings of the 13th Workshop in Primary and Secondary Computing Education*, 21:1-21:6. https://doi.org/10.1145/3265757.3265778

* Chang, C.-H. (2019). Does the learning of computational thinking concepts interact with the practice of digital curation in children? A preliminary case study. *Journal of Educational Media and Library Sciences*, 56(1), 45–68. https://doi.org/10.6120/JoEMLS.201903_56(1).0033.RS.CE

* Chen, Y., Chen, Z., Gumidyala, S., Koures, A., Lee, S., Msekela, J., Remash, H., Schoenle, N., Albright, S.D., Rebelsky, S.A. (2019). A middle-school code camp emphasizing digital humanities. *SIGCSE 2019 – Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 351–357. https://doi.org/10.1145/3287324.3287509

Clark, D.B., Sengupta, P. (2019). Reconceptualizing games for integrating computational thinking and science as practice: Collaborative agent-based disciplinarily-integrated games. *Interactive Learning Environments*. douglas.clark@ucalgary.ca

Clark, R.E. (1983). *Reconsidering Research on Learning from Media*. https://journals.sagepub.com/doi/abs/10.3102/00346543053004445?casa_token=ul-Kaj5_wBMAAAAA:wB2LGRyhy1byorIV9n6g8BP_3Q2uaAYphUDlfpgeOOPORKE-FGhzbxyXutfnqBqZJxMySkify44J

Cohen, L., Manion, L., Morrison, K.R.B. (2007). *Research methods in education, Sixth edition*. Routledge.

Council, N.R., Education, D. of B. and S. S. and, Education, B. on S., Standards, C. on a C. F. for N. K.-12 S. E. (2012). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. National Academies Press.

Creswell, J.W. (2012). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research*. Pearson.

Cunningham, T.D., Lansiquot, R.D. (2018). Modeling Interdisciplinary Place-Based Learning in Virtual Worlds: Lessons Learned and Suggestions for the Future. In R.D. Lansiquot & S.P. MacDonald (Eds.), *Interdisciplinary Place-Based Learning in Urban Education: Exploring Virtual Worlds* (pp. 133–145). Springer International Publishing. https://doi.org/10.1007/978-3-319-66014-1_7

Cuny, J., Snyder, L., Wing, J.M. (2010). Demystifying computational thinking for non-computer scientists. *Work in Progress*.

Curriculum of the Netherland. (2019). *Digitale geletterdheid – curriculum.nu*. https://www.curriculum.nu/voorstellen/digitale-geletterdheid/

Dagiene, V., Dolgopolovas, V. (2022). Short Tasks for Scaffolding Computational Thinking by the Global Bebras Challenge. *Mathematics*, *10*(17), Article 17. https://doi.org/10.3390/math10173194

Daily, S.B., Leonard, A.E., Jörg, S., Babu, S., Gundersen, K. (2014). *Dancing Alice: Exploring embodied pedagogical strategies for learning computational thinking*.

Davies, S. (2008). The effects of emphasizing computational thinking in an introductory programming course. *2008 38th Annual Frontiers in Education Conference*, T2C-3-T2C-8.

de Araujo, A.L.S.O., Andrade, W.L., Serey Guerrero, D.D. (2016). A systematic mapping study on assessing computational thinking abilities. *2016 IEEE Frontiers in Education Conference (FIE)*, 1–9. https://doi.org/10.1109/FIE.2016.7757678

Dillenbourg, P. (1999). *What do you mean by collaborative learning?. In P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches.* Oxford: Elsevier.

* Djurdjevic-Pahl, A., Pahl, C., Fronza, I., El Ioini, N. (2016). A Pathway into Computational Thinking in Primary Schools. *International Symposium on Emerging Technologies for Education*, 165–175.

* Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., Andrews, A. (2019). PRADA: A Practical Model for Integrating Computational Thinking in K-12 Education. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 906–912. https://doi.org/10.1145/3287324.3287431

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

* Foerster, K.-T. (2016). Integrating Programming into the Mathematics Curriculum: Combining Scratch and Geometry in Grades 6 and 7. *Proceedings of the 17th Annual Conference on Information Technology Education*, 91–96. https://doi.org/10.1145/2978192.2978222

* Folgieri, R., Vanutelli, M.E., Galbiati, P.D.V., Lucchiari, C. (2019). Gamification and Coding to Engage Primary School Students in Learning Mathematics: A Case Study. *In Proceedings of the 11th International Conference on Computer Supported Education. 1 (pp. 506–513)* `https://doi.org/10.5220/0007800105060513`

* Folk, R., Lee, G., Michalenko, A., Peel, A., Pontelli, E. (2015). GK-12 DISSECT: Incorporating computational thinking with K-12 science without computer access. 2015 *IEEE Frontiers in Education Conference (FIE)*, 1–8. `https://doi.org/10.1109/FIE.2015.7344238`

* Fontenot, A.D., Burgess, R.A., Subburaj, V.H., Nash, D.J. (2013). Introducing 6-12 Grade Teachers and Students to Computational Thinking. *2013 ASEE Annual Conference & Exposition*, 23–812.

Force, C.C.T.T. (2011). *Operational Definition of Computational Thinking for K–12 Education* (Vol. 2013).

* Förster, E.-C., Förster, K.-T., Löwe, T. (2018). Teaching programming skills in primary school mathematics classes: An evaluation using game programming. *2018 IEEE Global Engineering Education Conference (EDUCON)*, 1504–1513.

Foster, I. (2006). A two-way street to science's future. *Nature*, 440(7083), 419–419.

* Freudenthal, E.A., Lim, K.H., Kranz, S., Tabor, C., Ramirez, J.L. (2013). Using programming to strengthen mathematics learning in 9th grade algebra classes. *2013 ASEE Annual Conference & Exposition*, 23–1332.

* Friend, M., Matthews, M., Winter, V., Love, B., Moisset, D., Goodwin, I. (2018). Bricklayer: Elementary Students Learn Math through Programming and Art. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 628–633.

* Fronza, I., Ioini, N.E., Corral, L. (2016). Computational Thinking Through Mobile Programming – A Case Study in a Liberal Education Context. *MobiWIS*, 9847, 67–80.

* Gadanidis, G. (2017). Five affordances of computational thinking to support elementary mathematics education. *Journal of Computers in Mathematics and Science Teaching*, 36(2), 143–151.

* 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.

* Garneli, V., Chorianopoulos, K. (2018). Programming video games and simulations in science education: Exploring computational thinking through code analysis. *Interactive Learning Environments*, 26(3), 386–401.

* Gendreau Chakarov, A., Recker, M., Jacobs, J., Van Horne, K., Sumner, T. (2019). Designing a Middle School Science Curriculum That Integrates Computational Thinking and Sensor Technology. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 818–824. `https://doi.org/10.1145/3287324.3287476`

* Gero, A., Levin, I. (2019). Computational thinking and constructionism: Creating difference equations in spreadsheets. *International Journal of Mathematical Education in Science and Technology*, 50(5), 779–787.

* Giordano, D., Maiorana, F. (2013). An interdisciplinary project in sustainable development based on modern visual programming environments and web 2.0 technologies. *3rd Interdisciplinary Engineering Design Education Conference*, 163–167.

* Goncharenko, T., Kushnir, N., Valko, N., Osipova, N. (2019). Activity plan template for supporting study science with robotics and programming. *Integration, Harmonization and Knowledge Transfer Proceedings of the 15th International Conference on ICT in Education, Research and Industrial Applications.*, 2, 132–143.

Gouws, L.A., Bradshaw, K., Wentworth, P. (2013). Computational Thinking in Educational Activities: An Evaluation of the Educational Game Light-bot. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 10–15. `https://doi.org/10.1145/2462476.2466518`

* Greenberg, R.I. (2017). Pythagorean approximations for LEGO: Merging educational robot construction with programming and data analysis. *International Conference on Robotics and Education* RiE 2017, 65–76.

* Grgurina, N., Barendsen, E., Suhre, C., Zwaneveld, B., Veen, K. van. (2018). Assessment of modeling and simulation in secondary computing science education. *In Proceedings of the 13th Workshop in Primary and Secondary Computing Education*, 7.

Grover, S. (2011). Robotics and Engineering for Middle and High School Students to Develop Computational Thinking. *Annual Meeting of the American Educational Research Association, New Orleans, LA*.

Grover, S. (2018, February 25). *The 5th 'C' of 21st Century Skills? Try Computational Thinking (Not Coding) – EdSurge News*. EdSurge. `https://www.edsurge.com/news/2018-02-25-the-5th-c-of-21-st-century-skills-try-computational-thinking-not-coding`

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/10.3102/0013189X12463051`

Grover, S., Pea, R. (2018). Computational Thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School*, 19–38.

Guzdial, M. (2019). *Computing for Other Disciplines* (S.A. Fincher & A.V. Robins, Eds.; pp. 584–605). Cambridge University Press.

* Hadad, R., Thomas, K., Kachovska, M., Yin, Y. (2020). Practicing formative assessment for computational thinking in making environments. *Journal of Science Education and Technology*, 29(1), 162–173.

Hambrusch, S., Hoffmann, C., Korb, J.T., Haugan, M., Hosking, A.L. (2009). A Multidisciplinary Approach Towards Computational Thinking for Science Majors. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 183–187. https://doi.org/10.1145/1508865.1508931

* Hammond, T.C., Oltman, J., Manfra, M.M. (2019). Geo-Computational Thinking in the Third Grade: Making Computational Thinking Truly" For Everyone, Everywhere". *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geo-Computational Thinking in Education*, 1–5.

* Harrison, A., Hulse, T., Manzo, D., Micciolo, M., Ottmar, E., Arroyo, I. (2018). Computational Thinking Through Game Creation in STEM Classrooms. *International Conference on Artificial Intelligence in Education*, 10948, 134–138.

Hedy–Textual programming for the classroom. (2021). https://www.hedycode.com/

* Hestness, E., Ketelhut, D.J., McGinnis, J.R., Plane, J. (2018). Professional knowledge building within an elementary teacher professional development experience on computational thinking in science education. *Journal of Technology and Teacher Education*, 26(3), 411–435.

Hickmott, D., Prieto-Rodriguez, E., Holmes, K. (2018). A Scoping Review of Studies on Computational Thinking in K–12 Mathematics Classrooms. *Digital Experiences in Mathematics Education*, 4, 1–22. https://doi.org/10.1007/s40751-017-0038-8

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 & Education*, 126, 296–310.

* Hutchins, N., Biswas, G., Conlin, L., Emara, M., Grover, S., Basu, S., McELHANEY, K. (2018). Studying synergistic learning of physics and computational thinking in a learning by modeling environment. *26th International Conference on Computers in Education (ICCE), Manila, Philippines*.

* Israel, M., Lash, T. (2019). From classroom lessons to exploratory learning progressions: Mathematics + computational thinking. *Interactive Learning Environments*. misrael@coe.ufl.edu

* 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.

* Jacobson, M.J., Kim, B., Pathak, S., Zhang, B. (2015). To guide or not to guide: Issues in the sequencing of pedagogical structure in computational model-based learning. *Interactive Learning Environments*, 23(6), 715–730.

* Jaipal-Jamani, K., Angeli, C. (2017). Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology*, 26(2), 175–192.

* Jawad, H.M., Tout, S., Abualkibash, M., Xie, Y. (2018). Integrating Art and Animation in Teaching Computer Programming for High School Students Experimental Study. *2018 IEEE International Conference on Electro/Information Technology (EIT)*, 0311–0317. https://doi.org/10.1109/EIT.2018.8500178

* Jenkins, C. (2015). Poem generator: A comparative quantitative evaluation of a Microworlds-based learning approach for teaching English. *International Journal of Education and Development Using ICT*, 11(2).

Kafai, Y.B. (2016). From computational thinking to computational participation in K--12 education. *Communications of the ACM*, 59(8), 26–27. https://doi.org/10.1145/2955114

Kafai, Y.B., Burke, Q. (2017). Computational Participation: Teaching Kids to Create and Connect Through Code. In *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 393–405). Springer.

* Kafai, Y.B., Lee, E., Searle, K., Fields, D., Kaplan, E., Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1–20.

* Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., Grise, K. (2018). Computational What? Relating Computational Thinking to Teaching. *TECHTRENDS*, 62(6), 574–584. https://doi.org/10.1007/s11528-018-0290-9

Kallia, M., van Borkulo, S.P., Drijvers, P., Barendsen, E., Tolboom, J. (2021). Characterising computational thinking in mathematics education: A literature-informed Delphi study. *Research in Mathematics Education*, 23(2), 159–187. https://doi.org/10.1080/14794802.2020.1852104

Kampylis, P., Dagienė, V., Bocconi, S., Chioccariello, A., Engelhardt, K., Stupurienė, G., Masiulionytė-Dagienė, V., Jasutė, E., Malagoli, C., Horvath, M., Earp, J. (2023). Integrating Computational Thinking into Primary and Lower Secondary Education: A Systematic Review. *Educational Technology & Society*, 26(2), 99–117.

* Kanaki, K., Kalogiannakis, M. (2018). Introducing Fundamental Object-oriented Programming Concepts in Preschool Education Within the Context of Physical Science Courses. *Education and Information Technologies*, 23(6), 2673–2698. https://doi.org/10.1007/s10639-018-9736-0
* Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73, 26–39.
* Khan, M.J., Aji, C.A. (2018). Impact of programming robots and drones on STEM attitudes. *ASEE Annual Conference and Exposition, Conference Proceedings*, *2018-June*. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85051169795&partnerID=40&md5=6421c8fb088d4794bef5ccc8d44c430b
* Klopfer, E., Scheintaub, H., Huang, W., Wendel, D., Roque, R. (2009). The simulation cycle: Combining games, simulations, engineering and science using StarLogo TNG. *E-Learning and Digital Media*, 6(1), 71–96.
* Koh, K.H., Repenning, A., Nickerson, H., Endo, Y., Motter, P. (2013). Will It Stick?: Exploring the Sustainability of Computational Thinking Education Through Game Design. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 597–602. https://doi.org/10.1145/2445196.2445372

Kolb, D.A. (2014). *Experiential Learning: Experience as the Source of Learning and Development*. FT Press.
* Kong, S.C., Li, P. (2016). A case study illustrating coding for computational thinking development. *ICCE 2016 - 24th International Conference on Computers in Education: Think Global Act Local – Main Conference Proceedings*, 579–584.

Krippendorff, K. (2018). *Content Analysis: An Introduction to Its Methodology*. SAGE Publications.
* Krishnamoorthy, S.P., Kapila, V. (2016). Using a visual programming environment and custom robots to learn C programming and K-12 STEM concepts. *Proceedings of the 6th Annual Conference on Creativity and Fabrication in Education*, 41–48.

Kuo, W.-C., Hsu, T.-C. (2020). Learning Computational Thinking Without a Computer: How Computational Participation Happens in a Computational Thinking Board Game. *The Asia-Pacific Education Researcher*, 29(1), 67–83. https://doi.org/10.1007/s40299-019-00479-9
* Lai, C.-S., Lai, M.-H. (2012). Using Computer Programming to Enhance Science Learning for 5th Graders in Taipei. *2012 International Symposium on Computer, Consumer and Control*, 146–148. https://doi.org/10.1109/IS3C.2012.45
* Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L. (2011). Computational Thinking for Youth in Practice. *ACM Inroads*, 2(1), 32–37. https://doi.org/10.1145/1929887.1929902
* Lee, V.R., Vincent, H. (2019). An Expansively-framed Unplugged Weaving Sequence Intended to Bear Computational Fruit of the Loom. *Proceedings of FabLearn 2019*, 124–127. https://doi.org/10.1145/3311890.3311907
* Lehmkuhl-Dakhwe, K.V. (2019). An instructional framework, model lessons, and professional learning program for science standards-aligned computing in 4th–12th grade classrooms. *Proceedings – Frontiers in Education Conference, FIE*, *2018-October*. https://doi.org/10.1109/FIE.2018.8658580
* Leonard, A.E., Dsouza, N., Babu, S.V., Daily, S.B., Jörg, S., Waddell, C., Parmar, D., Gundersen, K., Gestring, J., Boggs, K. (2015). Embodying and programming a constellation of multimodal literacy practices: Computational thinking, creative movement, biology, & virtual environment interactions. *Journal of Language and Literacy Education*, 11(2), 64–93.
* Lévano, M., Córdova, V., Albornoz, A. (2016). Methodological framework for the development of computational thinking and programming through experiential learning: Case study from schools Juan Seguel and Allipen in Freire, Chile. *Proceedings of the 9th IADIS International Conference Information Systems 2016, IS 2016*, 107–114.
* Litts, B.K., Kafai, Y.B., Lui, D.A., Walker, J.T., Widman, S.A. (2017). Stitching Codeable Circuits: High School Students' Learning About Circuitry and Coding with Electronic Textiles. *JOURNAL OF SCIENCE EDUCATION AND TECHNOLOGY*, 26(5), 494–507. https://doi.org/10.1007/s10956-017-9694-0
* Litts, B.K., Lewis, W.E., Mortensen, C.K. (2019). Engaging youth in computational thinking practices through designing place-based mobile games about local issues. *Interactive Learning Environments*.
* Liu, J., Feenstra, W., Hunt, M., Siegel, A., McGrane, K., Saenz-Otero, A. (2014). Students touch space in Zero Robotics programming competition with free downloadable curriculum. *2014 IEEE Aerospace Conference*, 1–12. https://doi.org/10.1109/AERO.2014.6836498

Lockwood, J., Mooney, A. (2017). Computational Thinking in Education: Where does it Fit? A systematic literary review. *CoRR*, *abs/1703.07659*.

Lopez, V., Hernandez, M.I. (2015). Scratch as a computational modelling tool for teaching physics. *Physics Education*, 50(3), 310.

* Louca, L.T., Zacharia, Z.C., Michael, M., Constantinou, C.P. (2011). Objects, entities, behaviors, and interactions: A typology of student-constructed computer-based models of physical phenomena. *Journal of Educational Computing Research*, 44(2), 173–201.

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.

* Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J., Milliken, A., Isvik, A., Bounajim, D., Wiebe, E.N., Barnes, T. (2019). Use, Modify, Create: Comparing Computational Thinking Lesson Progressions for STEM Classes. *ITiCSE*, 395–401.

* Malyn-Smith, J., Lee, I.A., Martin, F., Grover, S., Evans, M.A., Pillai, S. (2018). Developing a framework for computational thinking from a disciplinary perspective. *Conference Proceedings of the International Conference on Computational Thinking Education*, 182–186.

Margulieux, L., Ketenci, T.A., Decker, A. (2019). Review of measurements used in computing education research and suggestions for increasing standardization. *Computer Science Education*, 29(1), 49–78. https://doi.org/10.1080/08993408.2018.1562145

Martins-Pacheco, L.H., Gresse von Wangenheim, C.A., Da Cruz Alves, N. (2019). Assessment of computational thinking in K-12 context: Educational practices, limits and possibilities – A systematic mapping study. *CSEDU 2019 – Proceedings of the 11th International Conference on Computer Supported Education*, 1, 292–303.

McCoy-Parker, K.S., Paull, L.N., Rule, A.C., Montgomery, S.E. (2017). Challenging Elementary Learners with Programmable Robots during Free Play and Direct Instruction. *Journal of STEM Arts, Crafts, and Constructions*, 2(2), 6.

Mendoza, B., Reyes-Alamo, J., Wu, H., Carranza, A., Zavala, L. (2016). iPractice: A Self-assessment Tool for Students Learning Computer Programming in an Urban Campus. *Journal of Computing Sciences in Colleges*, 31(3), 93–100.

Mensing, K., Mak, J., Bird, M., Billings, J. (2013). Computational, model thinking and computer coding for U.S. Common Core Standards with 6 to 12 year old students. *2013 IEEE 11th International Conference on Emerging ELearning Technologies and Applications (ICETA)*, 17–22. https://doi.org/10.1109/ICETA.2013.6674397

* Merkouris, A., Chorianopoulos, K. (2018). Programming Touch and Full-body Interaction with a Remotely Controlled Robot in a Secondary Education STEM Course. *Proceedings of the 22Nd Pan-Hellenic Conference on Informatics*, 225–229. https://doi.org/10.1145/3291533.3291537

* Messer, D., Thomas, L., Holliman, A., Kucirkova, N. (2018). Evaluating the effectiveness of an educational programming intervention on children's mathematics skills, spatial awareness and working memory. *Education and Information Technologies*, 23(6), 2879–2888.

Meulen, A. van der, Hermans, F., Aivaloglou, E., Aldewereld, M., Heemskerk, B., Smit, M., Swidan, A., Thepass, C., Wit, S. de. (2021). Who participates in computer science education studies? A literature review on K-12 subjects. *PeerJ Computer Science*, 7, e807. https://doi.org/10.7717/peerj-cs.807

* Miller, J. (2019). STEM education in the primary years to support mathematical thinking: Using coding to identify mathematical structures and patterns. *ZDM*, 51(6), 915–927.

Mingo, W.D. (2013). *The Effects of Applying Authentic Learning Strategies to Develop Computational Thinking Skills in Computer Literacy Students* [PhD Thesis]. Wayne State University.

Moore, M.G. (1973). Toward a Theory of Independent Learning and Teaching. *The Journal of Higher Education*, 44(9), 661–679. https://doi.org/10.1080/00221546.1973.11776906

* Moreno-León, J., Robles, G. (2016). Code to learn with Scratch? A systematic literature review. *2016 IEEE Global Engineering Education Conference (EDUCON)*, 150–156. https://doi.org/10.1109/EDUCON.2016.7474546

* Musaeus, L.H., Musaeus, P. (2019). Computational Thinking in the Danish High School: Learning Coding, Modeling, and Content Knowledge with NetLogo. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 913–919. https://doi.org/10.1145/3287324.3287452

* Nesiba, N., Pontelli, E., Staley, T. (2015). DISSECT: Exploring the relationship between computational thinking and English literature in K-12 curricula. *2015 IEEE Frontiers in Education Conference (FIE)*, 1–8. https://doi.org/10.1109/FIE.2015.7344063

* Niemelä, P., Partanen, T., Harsu, M., Leppänen, L., Ihantola, P. (2017). Computational thinking as an emergent learning trajectory of mathematics. *Koli Calling*, 70–79.

* Noa, R. (2018). Computational Thinking: Constructing the Perceptions of Pre-service Teachers from Various Disciplines. *ISSEP*, 11169, 167–179.

* Noushad, N.F., Shim, J., Yoon, S.A. (2017). Integrating social problem solving with programming to enhance science agency through creation of mobile apps in middle school. In *Smith, B. K., Borge, M., Mercier, E., and Lim, K. Y. (Eds.). (2017). Making a Difference: Prioritizing Equity and Access in CSCL, 12th International Conference on Computer Supported Collaborative Learning (CSCL) 2017* (Vol. 2). Philadelphia, PA: International Society of the Learning Sciences.

* Nugent, G., Barker, B., Grandgenett, N., Adamchuk, V. (2009). The use of digital manipulatives in k-12: Robotics, GPS/GIS and programming. *2009 39th IEEE Frontiers in Education Conference*, 1–6.

* Nugent, G., Barker, B., Lester, H., Grandgenett, N., Valentine, D. (2019). *Wearable Textiles to Support Student STEM Learning and Attitudes*.

* Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., Wilensky, U. (2016). Bringing computational thinking into high school mathematics and science classrooms. *Proceedings of International Conference of the Learning Sciences, ICLS*, 2, 705–712.

Page, M.J., Moher, D., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., … McKenzie, J.E. (2021). PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews. *BMJ*, 372, n160. https://doi.org/10.1136/bmj.n160

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

* Pardo, A.M.S. (2018). Computational Thinking Between Philosophy and STEM – Programming Decision Making Applied to the Behavior of "Moral Machines" in Ethical Values Classroom. *IEEE-RITA*, 13(1), 20–29.

* Park, Y.-S., Park, M. (2018). Exploring Students Competencies to be Creative Problem Solvers With Computational Thinking Practices. *JOURNAL OF THE KOREAN EARTH SCIENCE SOCIETY*, 39(4), 388–400. https://doi.org/10.5467/JKESS.2018.39.4.388

* Parker, T. (2012). ALICE in the real world. *Mathematics Teaching in the Middle School*, *17*(7), 410–416.

* Peel, A., Friedrichsen, P. (2018). Algorithms, abstractions, and iterations: Teaching computational thinking using protein synthesis translation. *The American Biology Teacher*, 80(1), 21–28.

* Peel, A., Fulton, J., Pontelli, E. (2015). DISSECT: An experiment in infusing computational thinking in a sixth grade classroom. *2015 IEEE Frontiers in Education Conference (FIE)*, 1–8. https://doi.org/10.1109/FIE.2015.7344240

Peel, A., Sadler, T.D., Friedrichsen, P. (2022). Algorithmic Explanations: An Unplugged Instructional Approach to Integrate Science and Computational Thinking. *Journal of Science Education and Technology*, 31(4), 428–441. https://doi.org/10.1007/s10956-022-09965-0

Perkins, D.N., Salomon, G. (1989). Are cognitive skills context-bound? *Educational Researcher*, 18(1), 16–25.

* Pinto-Llorente, A.M., Casillas-Martín, S., Cabezas-González, M., García-Peñalvo, F. J. (2018). Building, coding and programming 3D models via a visual programming environment. *Quality and Quantity*, 52(6), 2455–2468. https://doi.org/10.1007/s11135-017-0509-4

* Pires, F., Maquine Lima, F.M., Melo, R., Serique Bernardo, J.R., de Freitas, R. (2019). Gamification and Engagement: Development of Computational Thinking and the Implications in Mathematical Learning. *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, 362–366. https://doi.org/10.1109/ICALT.2019.00112

Pöllänen, S., Pöllänen, K. (2019). Beyond Programming and Crafts: Towards Computational Thinking in Basic Education. *Design and Technology Education: An International Journal*, 24(1), 13–32.

Poulakis, E., Politis, P. (2021). Computational Thinking Assessment: Literature Review. In T. Tsiatsos, S. Demetriadis, A. Mikropoulos, V. Dagdilelis (Eds.), *Research on E-Learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives* (pp. 111–128). Springer International Publishing. https://doi.org/10.1007/978-3-030-64363-8_7

Puentedura, R. (2010). *SAMR and TPCK: Intro to advanced practice.* http://hippasus.com/resources/sweden2010/SAMR_TPCK_IntroToAdvancedPractice.pdf

Qin, H. (2009). Teaching computational thinking through bioinformatics to biology students. *SIGCSE'09 – Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 188–191. https://doi.org/10.1145/1508865.1508932

Ragonis, N. (2018). Computational Thinking: Constructing the Perceptions of Pre-service Teachers from Various Disciplines. *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, 167–179.

Repenning, A., Basawapatna, A. (2021). Explicative programming. *Communications of the ACM*, 64(11), 30–33. https://doi.org/10.1145/3486642

* Rich, K.M., Yadav, A., Schwarz, C.V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education*, 27(2), 165–205.

* Rodríguez-Martínez, J.A., González-Calero, J.A., Sáez-López, J.M. (2020). Computational thinking and mathematics using Scratch: An experiment with sixth-grade students. *Interactive Learning Environments*, 28(3), 316–327. https://doi.org/10.1080/10494820.2019.1612448

Ruthmann, A., Heines, J.M., Greher, G.R., Laidler, P., Saulters, I.I. (2010). Teaching computational thinking through musical live coding in scratch. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 351–355.

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

Selby, C. (2013, January). Computational thinking: The developing definition. *The 18th Annual Conference on Innovation and Technology in Computer Science Education*. http://eprints.soton.ac.uk/346937/

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(2), 351–380. https://doi.org/10.1007/s10639-012-9240-x

* So, S. (2018). Developing the STEM Experience for In-Service Primary Teachers through Micro-controlling Hardware and Coding. *2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET)*, 1–5.

Sullivan, F.R., Heffernan, J. (2016). Robotic construction kits as computational manipulatives for learning in the STEM disciplines. *Journal of Research on Technology in Education*, 48(2), 105–128.

* Swanson, H., Irgens, G.A., Bain, C., Hall, K., Woods, P., Rogge, C., Horn, M., Wilensky, U. (2018). *Characterizing Computational Thinking in High School Science*. International Society of the Learning Sciences, Inc.[ISLS].

Tang, X., Yin, Y., Lin, Q., Hadad, R., Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers and Education*, *148*(103798). https://doi.org/10.1016/j.compedu.2019.103798

Tedre, M., Simon, Malmi, L. (2018). Changing aims of computing education: A historical survey. *Computer Science Education*, *28*(2), 158–186. https://doi.org/10.1080/08993408.2018.1486624

* Terwilliger, M.G., Jackson, J.L., Stenger, C.L., Jerkins, J.A. (2019). Using computer programming activities and robots to teach generalization of a geometry concept. *Journal of Computing Sciences in Colleges*, 34(3), 82–90.

* Towhidnejad, M., Kestler, C., Jafer, S., Nicholas, V. (2014). Introducing computational thinking through stealth teaching. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 1–7.

Tress, G., Tress, B., Fry, G. (2005). Clarifying Integrative Research Concepts in Landscape Ecology. *Landscape Ecology*, 20(4), 479–493. https://doi.org/10.1007/s10980-004-3290-4

* Urban, M.J. (2015). *Improving K-12 STEM education outcomes through technological integration*. IGI Global.

* Valentine, K.D. (2018). Tinkering with Logo in an elementary mathematics methods course. *Interdisciplinary Journal of Problem-Based Learning*, 12(2).

Vallance, M., Towndrow, P.A. (2016). Pedagogic transformation, student-directed design and computational thinking. *Pedagogies: An International Journal*, 11(3), 218–234. https://doi.org/10.1080/1554480X.2016.1182437

Voogt, J., Fisser, P., Good, J., Mishra, P., Yadav, A. (2015). Computational Thinking in Compulsory Education: Towards an Agenda for Research and Practice. *Education and Information Technologies*, 20(4), 715–728. https://doi.org/10.1007/s10639-015-9412-6

* Wagh, A., Cook-Whitt, K., Wilensky, U. (2017). *Bridging Inquiry-Based Science and Constructionism: Exploring the Alignment between Students Tinkering with Code of Computational Models and Goals of Inquiry*. https://doi.org/10.1002/tea.21379

Walden, J., Doyle, M., Garns, R., Hart, Z. (2013). An Informatics Perspective on Computational Thinking. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 4–9. https://doi.org/10.1145/2462476.2483797

Waterman, K.P., Goldsmith, L., Pasquale, M. (2020). Integrating Computational Thinking into Elementary Science Curriculum: An Examination of Activities that Support Students' Computational Thinking in the Service of Disciplinary Learning. *Journal of Science Education and Technology*, *29*(1), 53–64. https://doi.org/10.1007/s10956-019-09801-y

Wei, X., Lin, L., Meng, N., Tan, W., Kong, S.-C., Kinshuk. (2021). The effectiveness of partial pair programming on elementary school students' Computational Thinking skills and self-efficacy. *Computers & Education*, 160, 104023. https://doi.org/10.1016/j.compedu.2020.104023

* Weintrop, D., Beheshti, E., 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(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Weintrop, D., Wilensky, U. (2013). Robobuilder: A Computational Thinking Game (Abstract Only). *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 736–736. https://doi.org/10.1145/2445196.2445430

* Weitze, C.L. (2017). Reflective, creative and computational thinking strategies used when students learn through making games. *Proceedings of the 11th European Conference on Games Based Learning, ECGBL 2017*, 744–753.

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

Wing, J.M. (2011). Computational thinking. *2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 3–3. https://doi.org/10.1109/VLHCC.2011.6070404

* Wolz, U., Stone, M., Pearson, K., Pulimood, S.M., Switzer, M. (2011). Computational Thinking and Expository Writing in the Middle School. *Trans. Comput. Educ.*, 11(2), 9:1–9:22. https://doi.org/10.1145/1993069.1993073

Wood, D.F. (2003). Problem based learning. *BMJ*, 326(7384), 328–330. https://doi.org/10.1136/bmj.326.7384.328

* Xiang, L., Passmore, C. (2015). A framework for model-based inquiry through agent-based programming. *Journal of Science Education and Technology*, 24(2–3), 311–329.

Yadav, A., Hong, H., Stephenson, C. (2016). Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TECHTRENDS*, 60(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

* Yadav, A., Krist, C., Good, J., Caeli, E.N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, 28(4), 371–400.

* Yu, X., Guo, X. (2018). Case study on "STEM+computational thinking" education model in Chinese K-12 schools. *Journal of Science Education*, 19(1), 163–177.

S. Yeni is a postdoctoral researcher at Leiden University in the Leiden Institute of Advanced Computer Science (the Netherlands). She received her B.Sc., M.Sc., and Ph.D. degrees in Computer Education and Instructional Technology Department. Her research interests focus on computer science education in K-12, specifically on computational thinking in context, contextualized computing, and technology-enhanced learning.

N. Grgurina works as a computing curriculum developer for primary and secondary education at SLO, the Netherlands Institute for Curriculum Development, and as computing teacher educator at the University of Groningen. In the past, she had worked as mathematics and computing teacher in secondary education. Her research interests include employing computational modelling and simulation in secondary education, and more broadly, the integration of computational thinking and digital literacy into various school subjects.

M. Saeli is an assistant professor of Science Education at Radboud University (Nijmegen, the Netherlands). Her scientific research focuses on computer science education, specifically teacher knowledge (e.g., pedagogical content knowledge and pedagogical design capacity) and the integration of computational thinking in the curriculum (CT in context). Also, her interests include sustainability in computer science education.

F. Hermans is a professor of Computer Science Education at the Vrije Universiteit Amsterdam. She also works as a high-school CS teacher one day a week at Lyceum Kralingen in the Codasium program. She is the creator of the Hedy programming language, a gradual and multi-lingual programming languages designed for teaching. She is the author of "The Programmer's Brain", a book that helps programmers understand how their brain works and how to use it more effectively. In 2021, she was awarded the Dutch Prize for ICT research. She also has a biweekly tech column on BNR, a Dutch radio station.

J. Tolboom works as a mathematics and computing curriculum developer for upper secondary education at the Netherlands Institute for Curriculum Development. His research interests focus on curricular and didactical aspects of STEM education, with an emphasis on the utilization of technology for mathematics and computing education.

E. Barendsen is a professor of Science Education at Radboud University (Nijmegen, the Netherlands) and a professor of Computing Education at Open University of the Netherlands. His scientific interests include design-based and context-based teaching and learning in computer science and STEM subjects, scientific literacy, computational thinking and its integration into the school curriculum, digital literacy, and teachers' practical knowledge, in particular Pedagogical Content Knowledge (PCK).