# How is Computational Thinking Assessed in European K-12 Education? A Systematic Review

**Masiar BABAZADEH**[1]
**Lucio NEGRINI**[1]

[1] Scuola Universitaria Professionale della Svizzera italiana, Dipartimento formazione e apprendimento, Laboratorio media e MINT, Locarno, Switzerland

DOI: 10.21585/ijcses.v5i4.138

**Abstract**

Computational thinking (CT) is seen as a key competence of the 21st century and different countries have started to integrate it into their compulsory school curricula. However, few indications exist on how to assess CT in compulsory school. This review analyses what tools are used to assess CT in European schools and which dimensions are assessed. We analysed 26 studies carried out in K-12 between 2016 and 2020 in Europe. The results indicate that 18 different tools have been used and they can be categorized into five groups: questionnaires, tests/tasks, observations, interviews and analysis of products. From the tools we analysed, more than 50 dimensions were assessed and the vast majority of those were closer to programming skills rather than CT per se. Based on these results it seems that a common operational definition of CT, a competence model that indicates which competences students should reach at which age, and a tool that allows all different facets of CT to be assessed are currently missing.

**Keywords:** computational thinking, assessment, k-12, computational thinking dimensions

## 1. Introduction

In recent years, in the wake of digitisation and automation of our society, computational thinking (CT) and more in general digital literacy have been seen as key competences of the 21st century (World Economic Forum, 2016). CT has been popularized by Wing (2006), and involves "solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science" (p.33). After Wing, different authors have proposed other definitions of CT or have tried to operationalise Wing's idea. For example, according to the International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA), CT includes formulating problems; logically organising and analysing data; representing data through abstractions, models and simulations; automating problem resolution through algorithmic thinking; testing and improving the possible solutions and transferring the problem solving process to a variety of problems (CSTA & ISTE, 2011).

Brennan and Resnick (2012) distinguish three dimensions of CT: computational concepts (the knowledge component) that include concepts that programmers use for example the variables; computational practices (the skills component) that include the problem solving practices that occurs in the process of programming; and computational perspectives (the attitude component) that include students' understandings of themselves, their relationships to others, and the technological world around them (Lye & Koh, 2014). Other authors define CT as "the ability to think with the computer-as-tool" (Berland & Wilensky, 2015, p.630) or as "students using computers to model their ideas and develop programs [...] and consider computer programming as one part of computational thinking" (Israel, Pearson, Tapia, Wherfel & Reese, 2015, p.264). Shute, Sun and Asbell-Clarke (2017) in a literature review, synthesize various definitions of CT and propose a framework for CT particularly for K-12. Their working definition of CT is: "The conceptual foundation required to solve problems effectively and efficiently (i.e. algorithmically, with or without the assistance of computers) with solutions that are reusable in

different contexts" (p.151) and formulated six main facets of CT: decomposition, abstraction, algorithm design, debugging, iteration, and generalization. According to Shute et al., (2017), "CT is primarily a way of thinking and acting, which can be exhibited through the use of particular skills, which then can become the basis for performance-based assessments of CT skills" (p.142). Consequently, computational thinking skills should be expressed in terms of competences, i.e., as a combination of knowledge, skills and attitudes that are expressed in a context (European Commission, 2016).

Due to the augmented importance of CT for our society, there have been increasing calls for CT and related concepts such as coding or programming to be integrated into European school curricula (Académie des Sciences, 2013; European Commission, 2016; Royal Society, 2012; Schweizerische Eidgenossenschaft, 2017). Several States therefore have, as part of curricula reforms, included CT and related concepts in compulsory schooling (e.g. England (UK), France, Finland, Italy, Germany, and Switzerland at a regional level) or are planning to introduce it in the next years (e.g. Norway, Denmark) (Bocconi et al., 2016; Bocconi, Chioccariello & Earp, 2018).

Different approaches on how to introduce CT in schools exist (Chioccariello & Olimpo, 2017). The most common approaches found in the literature are based on programming (e.g., Scratch), educational robotics, game design, or paper and pencil activities (Calmet, Hirtzig, & Wilgenbus, 2016). CT is often taught by asking students to create algorithms in order to solve exercises first, and open-ended problems after, or to create software artefacts like video games or animations for example. Those algorithms can be created without the use of technology (for example, by the means of unplugged activities where a student-programmer gives instructions to a student-robot to navigate through a maze), with visual programming languages or with textual programming languages (Da Cruz Alves, Gresse Von Wangenheim, & Hauck, 2019).

Despite the numerous suggestions from different didactic materials and tools to carry out CT in class, only a few indications exist on how to assess CT skills in K-12. A large body of literature published in recent years indicates different challenges in the development of widely accepted assessment methods and frameworks that encompass the complexity of CT (Brennan & Resnick, 2012; Denner, Werner, & Ortiz, 2012; Denning, 2017; Fronza, El Ioini, & Corral, 2017; Grover et al., 2017; Grover, Pea, & Cooper, 2015; Tikva & Tambouris, 2021; Zhong, Wang, Chen, & Li, 2016). These challenges are also due to the variety of CT definitions that make it difficult to develop a common and reliable assessment tool (Adams, Cutumisu, & Lu, 2019). The existing tools are often developed to measure single dimensions of CT highly linked to the tool adopted (e.g., programming concepts in Scratch, or debugging strategies with educational robots). This makes it even more difficult to measure CT in its entirety and therefore evaluate the effectiveness of the CT activities that are carried out in schools. This situation neither allows for an overview on the different approaches, nor to compare results across various studies (Shute et al., 2017). The assessment of CT is however essential in order to successfully implement CT in schools (Grover & Pea, 2013) since teachers need to collect evidence on what they propose to better understand their students' progress. Valid and reliable assessments tools also help to evaluate the effectiveness of different CT curricula (Basu, Rutstein, Xu, Wang & Shear, 2021). Tools to assess CT need to include all dimension of CT such as the assessment of understanding of programming or CT concepts alongside assessment of general problem-solving practices such as logical thinking, formulation of a problem as a set of computational steps, pattern recognition, abstraction and generalization, decomposition and modularization, data collection and organization; data-based decision making, and systematic incremental testing and debugging that are important in contexts beyond programming (Atmatzidou & Demetriadis, 2016; Barr, Harrison, & Conery, 2011; Basu, Rutstein, Xu, Wang & Shear, 2021; Csizmadia, Curzon, Dorling, Humphreys, Ng, Selby, & Woollard, 2015).

In the last few years, some reviews on CT assessment tools have been published. Those reviews have been carried out in Canada (Adams et al., 2019; Cutumisu, Adams, & Lu, 2019), Brazil (Da Cruz Alves et al., 2019) and United States (Tang, Yin, Lin, Hadad & Zhai, 2020). The review by Da Cruz Alves et al., (2019), however focuses on the tools that assess programming activities based on code and not on CT in a broader sense as defined in this paper. Tang et al., (2020) include studies done in colleges, high schools or teacher education, while Cutumisu et al., (2019) analyse studies done between 2014 and 2018. The cited reviews show a breadth of methods employed to assess CT (Cutumisu et al., 2019) that include four main forms: traditional assessment composed of selected- and/or constructed-response questions, portfolio assessments, surveys, as well as interviews, and claim that most of the CT assessment tools analyse concepts directly related to algorithms and programming (Tang et al., 2020). Another study of Çoban and Korkmaz (2021), highlights in the literature, "it has been seen that computational thinking is evaluated with different measurement tools. Many more methods such as scales, portfolio studies, coding, multiple choice tests, task-based tests, observations, and rubrics have been applied with different methodologies" (p. 2). There are however no reviews focusing on contemporary studies conducted in European

compulsory schools with a broader understanding of CT. A review on how CT is currently assessed in European schools and which tools are used can therefore help to make an overview on the different existing practices and to formulate implications for the field. In this paper, we intend to explore and describe the CT assessment approaches used in K-12 education in Europe focusing on the last years. The questions addressed are the following: (1) Which tools are used to assess CT in Europe? and (2) Which dimensions of CT are assessed?

This review will help to create an overview on different approaches and tools used to assess CT in K-12 in Europe and on the CT dimensions assessed. This knowledge could potentially help design and develop a reliable and valid CT assessment tool. The paper is structured as follows: Section 2 describes the methodology used to obtain the reviewed papers. Section 3 presents the results, aggregating the studies and illustrating the tools used, and the dimensions of CT assessed. Section 4 discusses the results and Section 5 presents the conclusions.

## 2. Methodology

For this systematic review we adopted the method for implementing reviews in the social sciences by Petticrew and Roberts (2006). Specifically, we followed these steps: (1) research questions were formulated; (2) the search terms were defined, and appropriate databases were selected; (3) inclusion and exclusion criteria were formulated; (4) the obtained papers were screened and selected; (5) the data to answer the research questions were extracted.

### *2.1 Definition of Search Terms and Selection of Database*

For this review we have consulted seven databases: ERIC, PsycInfo, Scopus, Web of Science, Google Scholar, ACM Digital Library, and ProQuest Dissertations and Theses Global. The selection of these databases includes journals involving educational research as well as computer sciences research. We decided to also include Google Scholar and ProQuest Dissertations and Theses Global to also reach grey literature and dissertations. The search terms used are the following:

- Query 1: "computational thinking" AND ("K-12" OR "primary school" OR "secondary school") AND "assessment"
- Query 2: "computational thinking" AND ("assessment" OR "test" OR "evaluation" OR "exam" OR "measure") AND ("K-12" OR "primary school" OR "elementary school" OR "secondary school" OR "middle school")
- Query 3: "computational thinking" AND "assessment"

We started with Query 1, however we noticed that the term assessment could also be used with a synonym like "test", "evaluation", "exam", or "measure". We decided therefore to do a second query including also these terms. The first two queries however did not allow studies carried out in educational systems that use other names to refer to K-12 education other than "primary", "elementary", "secondary", or "middle school". In order to be sure to also include educational systems that use other names we carried out a third query with only the terms "computational thinking" and "assessment". This query is a superset of the previous two queries, with a much broader spectrum of results. Even though such a query ended up including many out-of-scope papers, we decided to keep the results and filter them ourselves in order not to exclude a priori papers that use another terminology. The research in fact yielded a total of 30432 papers. After removing duplicates, we obtained a corpus of 13872 papers. The consultation of the databases has been concluded during the first week of December 2020.

### *2.2 Inclusion and Exclusion Criteria*

To select the articles, we used the following inclusion criteria:

- We decided to include papers conducted from 2016 onward (10 years after Wing's seminal work). We are aware that this criterion could represent a limitation since we are excluding studies carried out before 2016. This decision was made in order to limit the number of papers and in order to include only actual and more recent assessment tools that are used in European schools.
- The study needs to be conducted in Europe since we are interested in the assessment tools used in European schools. This criterion was formulated in order to avoid including assessment tools that relate to instructional practices that are not present in European schools. We are aware that the school systems in Europe are different from each other, the instructional practices related to CT (robotics, coding, unplugged activities) are however very similar between European countries.
- The paper has to be written in English. This criterion could also represent a limitation since European literature could be published in different languages. The most common language used in this field of research however is English.

- The paper needs to have the full text available.
- The study has to be done in the context of formal K-12 education.
- The paper explains which dimensions of CT are assessed.
- The paper presents a tool/test to assess CT.
- The tool has been tested in class. The paper should present a tool that has been applied in class. Papers on theoretical reflections on how to assess CT without a tool tested in class have been excluded.

*2.3 Screening of Papers*

The screening process can be seen in Figure 1. All papers have been screened applying the inclusion criteria. A first screening round based mostly on the abstracts and metadata of the papers lead us to eliminate studies carried out before 2016, non-English papers and papers on studies conducted in non-European countries. After applying these first criteria, we obtained 175 papers. In a second round, the inclusion criteria were applied to the full-text versions of the 175 selected papers. The execution of this second round resulted in 26 papers that match the inclusion criteria.

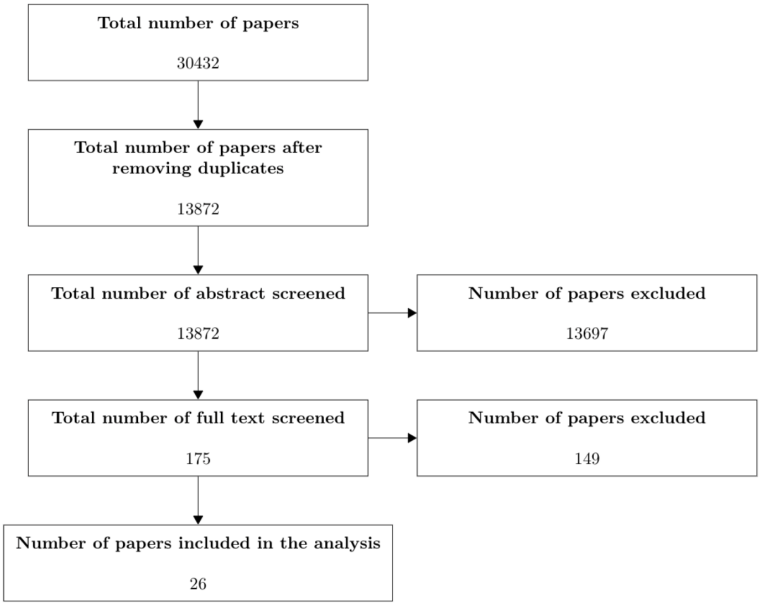


Figure 1. Screening process and stages.

*2.4 Data Extraction*

The 26 selected papers were read a second time more in depth and a series of information was selected in order to answer the research questions. For all 26 papers we extracted the following data:

- Authors and date
- Type of the paper (e.g., scientific article, conference paper, dissertation, ...)
- Nation where the study was conducted
- Name of the assessment tool implemented
- Form of the tool (e.g., paper based, computer based, test, questionnaire, ...)
- Length of the assessment (e.g., number of items)
- CT dimensions that were assessed
- Number of pupils in the study
- School grade in which the study was conducted

**3. Results**

The presentation of the findings was divided into three main topics: description of the selected studies, presentation of the assessment tools used in the different studies, and a report on the CT dimensions assessed by the different tools.

*3.1 General Description of the Selected Studies*

Of the 26 studies selected, 7 were conducted in Spain 4 of which were by the same research team. In Turkey 6 studies were carried out. In the UK there were 3. In Germany, Greece and Italy, 2 studies were carried out in each. Chiazzese, Arrigo, Chifari, Lonati & Tosto (2019) is an extended paper of the study Chiazzese, Arrigo, Chifari, Lonati & Tosto (2018). The last 4 studies were conducted in Czech Republic, Finland, Ireland, and in Slovenia. Seventeen studies were published as journal papers, 8 were conference papers and 1 as a chapter in a book. The number of pupils assessed with the CT tools ranges between 16 (Gillott, Joyce-Gibbons, & Hidson, 2020) and 1251 (Roman-Gonzalez, Perez-Gonzalez, & Jimenez-Fernandez, 2017). Sixteen studies have been carried out in primary schools and 10 in secondary education. The attribution to a grade is however dependent on the country where the study was conducted. In this study, in which we are aware that we could not represent all European school systems, we divided the school grades as follow: Preschool/Kindergarten (pupils aged below 5 years); Primary School (pupils aged between 6 and 11; Grades 1st-6th); Secondary School (pupils aged between 12-15; Grades 7th-10th). In the cases where the studies covered more grades, we have counted them according to the grade where the majority of the pupils were enrolled considering only the grades in compulsory schools. Table 1 shows all the selected studies ordered by school grade. The selected studies are also marked with an * in the reference list.

Table 1. Selected studies

| **Study** | **Nr. of pupils** | **Grade** | **Nation** | **Publication Type** |
|---|---|---|---|---|
| Kalliopi and Michail, 2019 | 450 | 1st-2nd | Greece | Conference paper |
| del Olmo-Muñoz, Cózar-Gutiérrez, and González-Calero, 2020 | 84 | 2nd | Spain | Journal paper |
| Price and Price-Mohr, 2018 | 18 | 2nd-5th | UK | Journal paper |
| Leifheit, Jabs, Ninaus, Moeller and Ostermann, 2018 | 33 | 3rd- 4th | Germany | Conference paper |
| Chiazzese, Arrigo, Chifari, Lonati and Tosto, 2018 | 83 | 3rd-4th | Italy | Conference paper |
| Chiazzese, Arrigo, Chifari, Lonati and Tosto, 2019 | 51 | 3rd-4th | Italy | Journal paper |
| Bryndová and Mališů, 2020 | 90 | 3rd-8th | Czech Republic | Conference paper |
| Fagerlund, Häkkinen, Vesisenaho, and Viiri, 2020 | 57 | 4th | Finland | Journal paper |
| Kožuh, Krajnc, Hadjileontiadis, and Debevc, 2018 | 945 | 4th-6th | Slovenia | Journal paper |
| Pérez-Marín, Hijón-Neira and Bacelo, 2018 | 132 | 4th-6th | Spain | Journal paper |
| Yildiz Durak, 2018 | 110 | 5th | Turkey | Journal paper |
| Tonbuloğlu and Tonbuloğlu, 2019 | 114 | 5th | Turkey | Journal paper |
| Saez-Lopez, Roman-Gonzalez, and Vazquez-Cano, 2016 | 139 | 5th-6th | Spain | Journal paper |
| Allsop, 2019 | 30 | 5th-6th | UK | Journal paper |
| Kukul and Karatas, 2019 | 319 | 5th-7th | Turkey | Journal paper |
| Korucu, Gencturk, and Gundogdu, 2017 | 160 | 5th-8th | Turkey | Journal paper |
| Roman-Gonzalez, Perez-Gonzalez, and Jimenez-Fernandez, 2017 | 1251 | 5th-10th | Spain | Journal paper |
| Förster, Förster, and Löwe, 2018 | 22 | 6th | Germany | Conference paper |
| Segredo, Miranda, León and Santos, 2016 | 54 | 6th-14th | Spain | Conference paper |
| Román-González, Moreno-León, and Robles, 2017 | 148 | 7th-8th | Spain | Conference paper |
| Bati, Yetişir, Çalişkan, Güneş and Saçan, 2018 | 104 | 8th | Turkey | Journal paper |
| Román-González, Pérez-González, Moreno-León and Robles, 2018 | 314 | 8th-9th | Spain | Journal paper |
| Sarıtepeci and Durak, 2017 | 53 | 9th | Turkey | Chapter in a book |

| | | | | |
|---|---|---|---|---|
| Garneli and Chorianopoulos, 2018 | 34 | 10th | Greece | Journal paper |
| Gillott, Joyce-Gibbons and Hidson, 2020 | 16 | 10th-11th | UK | Journal paper |
| Lockwood and Mooney, 2018 | 292 | 10th-12th | Ireland | Conference paper |

*3.2 Which Tools are Used to Assess CT in Europe?*

In this section we present a more in-depth analysis of the tools used to assess CT. Table 2 shows the different types of tools used in the 26 studies grouped by tool nature. Across the 26 articles, 18 unique forms of assessment were identified and grouped in five categories: test/tasks (Visual Blocks Creative Computing Test, CT Test, PCNT test, Bebras tasks, Code.org tasks, Scratch tasks, Alice tasks, Java tasks, Educational robotics tasks, CT activities, PhysGramming), questionnaires (CT Self-efficacy Scales, CT Ability Scale, CONT questionnaire and self-developed online questionnaires), observations, interviews, and analysis of products (manually, or automated with Dr Scratch). The most used tools are the CT Test (5 times), Bebras tasks (5 times), the Computational Thinking Ability Scale (4 times) Dr. Scratch (3 times), and Scratch tasks (3 times).

Table 2. Tool used by the analyzed papers.

| **Study** | **Name of Tool** | **Tool nature** | **Length of assessment** |
|---|---|---|---|
| Fagerlund, Häkkinen, Vesisenaho, and Viiri, 2020 | Scratch tasks | Analysis of products | - |
| Garneli and Chorianopoulos, 2017 | Dr. Scratch | Analysis of products | - |
| Förster, Förster, and Löwe, 2018 | Dr. Scratch | Analysis of products | Automatic assessment over 24 program elements |
| Price and Price-Mohr, 2018 | Java tasks | Interview and analysis of products | - |
| Allsop, 2019 | Scratch and Alice tasks, observations and interviews | Observation, interview, analysis of products | - |
| Gillot, Gibbons, and Hidson, 2020 | Scratch tasks and observations | Observations, interview | - |
| Kalliopi and Michail, 2019 | PhysGramming | Tasks, observations and interview | - |
| Chiazzese, Arrigo, Chifari, Lonati and Tosto, 2018 | Bebras task | Tasks | - |
| Chiazzese, Arrigo, Chifari, Lonati and Tosto, 2019 | Bebras task | Tasks | 10 items |
| del Olmo-Munoz, Cózar-Gutiérrez and González-Calero, 2020 | Bebras task | Tasks | 10 items |
| Lockwood and Mooney, 2018 | Bebras task | Tasks | 13 items |
| Segredo, Miranda, León and Santos, 2016 | CT activities | Tasks | 5 activities |
| Leifheit, Jabs, Ninaus, Moeller and Ostermann, 2018 | Code.org tasks | Tasks | 9 tasks |
| Bryndová and Mališů, 2020 | Educational robotics tasks | Tasks | 16 items |
| Saez-Lopez, Roman-Gonzalez, and Vazquez-Cano, 2016 | Visual Blocks Creative Computing Test | Test | 40 items |
| Korucu, Gencturk, and Gundogdu, 2017 | CT Ability Scale | Questionnaire | 22 items |
| Sarıtepeci and Durak, 2017 | CT Ability Scale | Questionnaire | 22 items |
| Yildiz Durak, 2018 | CT Ability Scale | Questionnaire | 22 items |
| Tonbuloglu and Tonbuloglu, 2019 | CT Ability Scale and observations | Questionnaire and observations | 22 items |
| Román-González, Moreno-León, and Robles, 2017 | CT Test, Dr. Scratch and Bebras tasks | Questionnaire, analysis of product and tasks | 28 items (CT Test) |
| Román-González, Perez-Gonzalez, | CT Test | Questionnaire | 28 items |

| | | | |
|---|---|---|---|
| and Jimenez-Fernandez, 2017 | | | |
| Román-González, Pérez-González, Moreno-León and Robles, 2018 | CT Test | Questionnaire | 28 items |
| Bati, Yetişir, Çalişkan, Güneş and Saçan, 2018 | CT Test, observations and interviews | Questionnaire, observations and interviews | Depends on the module |
| Pérez-Marín, Hijón-Neira, and Bacelo, 2018 | CT Test, PCNT Test, CONT questionnaire | Questionnaire | 15 items (CONT), 28 items (CT Test), 14 Items (PCNT) |
| Kožuh, Krajnc, Hadjileontiadis, and Debevc, 2018 | Online survey questionnaire | Questionnaire | 13 items |
| Kukul and Karatas, 2019 | CT Self-efficacy Scale | Questionnaire | 18 items |

**Test/Tasks.** The majority of the selected studies uses a test or a task to assess CT. Under this category we can find for example the Visual Blocks Creative Computing Test. This test has 40 items with a structured and progressive sequence. Students answer items related to sequences, loops, conditional statements, parallel execution, coordination, event handling, and keyboard input. Another frequently used test is the CT Test (Roman-Gonzalez et al., 2017), in which pupils have to solve 28 tasks. For example, a sequence of instructions is given to them and they have to decide how many times the sequence has to be executed in order to move a character from point A to point B on a grid. The CT test targets secondary school pupils. A similar test for primary school pupils is the PCNT Test (Pérez-Marín et al., 2018). Other often used tasks to assess CT skills are the Bebras tasks. The Bebras tasks are a large set of tasks used for the worldwide annual International Challenge on Informatics and Computational Thinking. The aim of the challenge is to increase pupils' engagement in informatics and to promote the development of computational thinking through the resolution of real-life and attractive problems (Chiazzese et al., 2018). Also the *code.org* tasks can be used to assess CT: the platform offers in fact different online courses for pupils which contain assessment tasks to be solved. In other cases, the researchers used tasks created with different programming languages for example Scratch, Alice or Java. In some cases, educational robotics tasks also can be used. Bryndová and Mališů (2020) for example use the robots Ozobot EVO and BIT. A last system that has been used to assess CT is PhysGramming (Kalliopi & Michail, 2019). PhysGramming is a digital environment that allows pupils to create their own games.

**Questionnaires.** Questionnaires related to CT are also often used to assess CT skills. Kukul et al. (2019) have for example developed the CT Self-Efficacy Scale. The original scale contained 51 items arranged as 5-point Likert scale (1 = "Completely Disagree" - 5 = "Fully Agree"). The scale was applied as a pilot to secondary school pupils and the items were reduced to 18 items. Example items on the scale are: "If there are sub-problems in the problem, I can manage the solution processes of these sub-problems"; "I can make connections between the current problem and previously encountered problems". Another questionnaire used is the CT Ability Scale developed by Korkmaz, Çakır, and Özden (2016). The scale was originally developed for university students and then was adapted for secondary school pupils. Also in this case, a five-point Likert scale is used. Examples of items in the test are "I believe that I can easily catch the relation between the figures"; "It is fun to try to solve the complex problems". Other questionnaires allowed open questions an example is the CONT questionnaire that measure knowledge of programming concepts. An example of a question is "What do you think a program is? Can you give an example?" (Pérez-Marín et al., 2018).

**Observations.** In some studies, CT skills have been assessed by observing pupils solving different tasks. For example, Allsop (2019) collected data on CT skills by observing "the language children used for their 'self' explanations and group discussions, the gestures, the context of their relations with teacher, peers and technology in their classroom setting" (p.33).

**Interviews.** In order to assess CT, few studies also asked pupils to create a product (a coded story) and then interviewed them to let them explain their coded story (Price & Price-Mohr, 2018).

**Analysis of products.** Another method to assess CT skills is to analyse pupils' projects. An example of this are projects created in Scratch. The products can be analysed manually or automatically with Dr. Scratch for example. Dr. Scratch is an online analysis tool which can assess CT skills of a Scratch project based on the number of sprites, blocks, loops, and other concepts used in the project, and calculate a CT skills score. Dr Scratch however has some limitations as it cannot detect if the program is functioning as intended: a project with the appropriate blocks could get a high CT score, although it may lack functionality (Moreno-León & Robles, 2015).

*3.3 Which Dimensions are Assessed?*

The presented tools have been used to analyse different dimensions of CT. Table 3 shows an overview of the analysed dimensions according to used tools and study grouped by tool name. Analysed dimensions is composed of keywords taken from the papers where the tool was used. The analysed dimensions of a specific tool used in more than one paper could be different or have slightly different terminology among the papers, depending on the authors' research focus and on how they have used the instrument.

In terms of concepts and processes, we stayed with what the authors defined as a CT dimension in the reviewed papers. This could include programming constructs as well as non-programming terms and processes. We addressed this difference in Section 4 and Section 5 and we highlighted why this is still an issue to reach a common operational definition of CT.

Table 3. Tools used in the selected studies and dimensions analyzed by the tools.

| **Study** | **Name of Tool (or type if N.A.)** | **Analysed dimensions** |
|---|---|---|
| Chiazzese, Arrigo, Chifari, Lonati and Tosto, 2018 | Bebras tasks | Algorithmic thinking, Implementing simple algorithmic procedures, Logically analyzing data, Logically organizing data, Representing data through formal encoding |
| Lockwood and Mooney, 2018 | Bebras tasks | Data ordering, Encoding, Gossip problem, If then else objects, Pattern matching attributes and variables, Stacks, Trees ciphering, Sorting |
| Del Olmo-Muñoz, Cózar-Gutiérrez, and González-Calero, 2020 | Bebras tasks | Algorithmic thinking, Decomposition, Evaluation, Generalisation |
| Leifheit, Jabs, Ninaus, Moeller and Ostermann, 2018 | Code.org tasks | Conditionals, Debugging, Events |
| Korucu, Gencturk, and Gundogdu, 2017 | CT Ability Scale | Algorithmic thinking, Analytical Thinking, Collaboration, Creativity, Problem solving |
| Sarıtepeci and Durak, 2017 | CT Ability Scale | Algorithmic thinking, Collaboration, Creativity, Critical thinking, Problem solving |
| Yildiz Durak, 2018 | CT Ability Scale | Algorithmic thinking, Collaboration, Creativity, Critical thinking, Problem solving |
| Tonbuloğlu and Tonbuloğlu, 2019 | CT Ability Scale | Algorithmic thinking, Collaboration, Creativity, Critical thinking, Problem solving |
| Segredo, Miranda, León and Santos, 2016 | CT activities | Abstraction, Algorithmic thinking, Cognitive planning, Logical thinking |
| Kukul and Karatas, 2019 | CT Self-efficacy Scale | Abstraction, Decomposition, Generalization, Reasoning |
| Román-González, Perez-Gonzalez, and Jimenez-Fernandez, 2017 | CT Test | Computational concepts (sequences, loops, conditionals, operators), Computational practices (testing and debugging, reusing and remixing, abstracting and modularizing) |
| Román-González, Moreno-León, and Robles, 2017 | CT Test, Dr Scratch, and Bebras tasks | Abstraction and problem decomposition, Data representation, Flow control, Logical thinking, Parallelism, Synchronization, User interactivity |
| Bati, Yetişir, Çalişkan, Güneş and Saçan, 2018 | CT Test, observations, and interviews | Assessing different approaches/solutions to a problem, Choosing effective computational tools, Creating abstractions, Developing modular computational solutions, Programming, Troubleshooting and debugging, Using problem solving strategies |
| Pérez-Marín, Hijón-Neira, and Bacelo, 2018 | CT Test, PCNT Test, CONT questionnaire | Abstract and encapsulate, Incremental and iterative development, Mix and reuse, Test and Debugging |
| Román-González, Pérez-González, Moreno-León and Robles, 2018 | CT Test | Computational concepts (sequences, loops, conditionals, operators), Computational practices (testing and debugging, reusing and remixing, abstracting and modularizing) |

| | | |
|---|---|---|
| Garneli and Chorianopoulos, 2017 | Dr Scratch | Data, Computational practices and perspectives, Conditionals, Events, Loops, Operators, Parallelism, Sequences |
| Förster, Förster, and Löwe, 2018 | Dr Scratch | Abstraction and problem decomposition, Algorithmic notions of flow control, Data representation, Logical thinking, Parallelism, Synchronization, User interactivity |
| Bryndová and Mališů, 2020 | Educational robotic tasks | Abstraction, Algorithmization, Decomposition, Evaluation, Generalization |
| Price and Price-Mohr, 2018 | Java tasks | Abstraction, Algorithmic thinking, Decomposition |
| Kožuh, Krajnc, Hadjileontiadis, and Debevc, 2018 | Online survey questionnaire | If-clause, Loops, Series of execute commands, Variables |
| Kalliopi and Michail, 2019 | PhysGramming | Abstraction, Algorithmic thinking, Data analysis (identifying misconceptions, reconsider choices), Data collection, Data organization |
| Allsop, 2019 | Scratch and Alice tasks, interviews, and observations | Abstractions, Conditionals, Events, Loops, Operators, Parallelism, Sequences, Variables |
| Fagerlund, Häkkinen, Vesisenaho, and Viiri, 2020 | Scratch | Abstraction, Algorithms, Automation, Coordination, Creativity, Data, Logic, Modeling and design, Patterns, Problem decomposition |
| Gillott, Joyce-Gibbons, and Hidson, 2020 | Scratch tasks, interviews, and observations | Abstraction, Algorithmic thinking, Computational concepts, Computational perspectives, Debugging/Testing, Decomposition, Evaluation, Formulate problems, Generalization/Reusing, Logical reasoning |
| Saez-Lopez, Roman-Gonzalez, and Vazquez-Cano, 2016 | Visual Blocks Creative Computing Test | Conditional statements, Event handling, Experimentation, Iteration, Keyboard input, Sequence, Threads, User Interface Design |

Table 4 shows the dimensions analysed in the selected papers ordered by number of appearances. The table doesn't take into account duplicates (that is, if a tool appeared in more than one study, the dimensions it analysed are counted only once). Whenever there are differences in the terminology used when referring to the same dimension, all the used terms are presented in the same row (e.g., abstract, abstraction). It is interesting to note that the tools presented in the selected studies allow 59 different dimensions to be analysed. Only 21 dimensions however appear more than once, while all the others (38) are representative of a single tool. This gives an initial indication which dimensions are most frequently taken into account and thus associated with CT. If the dimensions tested by the tools are analysed according to school grades it can be noticed that the type of dimensions used do not differ based on students' grades.

Table 4. Dimensions analysed by the reviewed papers.

| **Dimension** | **Appearances** | **Dimension** | **Appearances** |
|---|---|---|---|
| Abstract / Abstraction | 8 | Data collection | 1 |
| Algorithm / Algorithmic thinking / Algorithmization | 8 | Data ordering | 1 |
| Problem decomposition / Decomposition | 7 | Developing modular computational solutions | 1 |
| Conditionals / Conditional statements | 6 | Encapsulate | 1 |
| Generalization / Pattern recognition / Patterns | 6 | Encoding | 1 |
| Loops / Iterations | 5 | Execute commands | 1 |
| Events / Event handling | 4 | Experimentation | 1 |
| Logic / Logical thinking / Logical reasoning | 4 | Flow control | 1 |
| Sequences | 4 | Formulate problems | 1 |

| | | | |
|---|---|---|---|
| Creativity | 3 | Gossip problem | 1 |
| Debugging | 3 | Incremental development | 1 |
| Evaluation | 3 | Iterative development | 1 |
| Operators | 3 | Keyboard input | 1 |
| Parallelism | 3 | Mix / Remixing | 1 |
| Variables | 3 | Modeling and design | 1 |
| Analyze data / Data analysis | 2 | Programming | 1 |
| Computational practices and perspectives | 2 | Reasoning | 1 |
| Data | 2 | Reuse | 1 |
| Data representation | 2 | Simple functions | 1 |
| Organize data / Data organization | 2 | Sorting | 1 |
| Problem solving | 2 | Stacks | 1 |
| Analytical / critical thinking | 1 | Synchronization | 1 |
| Assessing different approaches/solutions to a problem | 1 | Test | 1 |
| Automation | 1 | Threads | 1 |
| Choosing Effective Computational Tools | 1 | Trees | 1 |
| Ciphering | 1 | Troubleshooting | 1 |
| Cognitive planning | 1 | User interactivity | 1 |
| Collaboration | 1 | User interface / User interface design | 1 |
| Computational Concepts | 1 | While conditional | 1 |
| Coordination | 1 | | |

The dimensions mentioned in Table 4 can be further divided into dimensions purely related to programming and informatics, and dimensions related in a broader sense to CT. To this respect, we decided to select the dimensions which can be associated with the definition of CT given Wing and Shute's seminal works and present them in Table 5. The table shows 19 dimensions in total related to Wing and Shute's works. Interestingly enough, more than half of the dimensions (11) are mentioned in more than one tool.

Table 5. Dimensions analysed that can be related to CT as seen in Wing and Shute's seminal work.

| **CT dimension (Wing, 2006 and Shute et al., 2017)** | **Appearances** |
|---|---|
| Abstract / Abstraction | 8 |
| Algorithm / Algorithmic thinking / Algorithmization | 8 |
| Problem decomposition / Decomposition | 7 |
| Generalization / Pattern recognition / Patterns | 6 |
| Logic / Logical thinking / Logical reasoning | 4 |
| Creativity | 3 |
| Debugging | 3 |
| Evaluation | 3 |
| Analyze data / Data analysis | 2 |
| Organize data / Data organization | 2 |
| Problem solving | 2 |
| Analytical / Critical thinking | 1 |
| Assessing different approaches/solutions to a problem | 1 |
| Cognitive planning | 1 |
| Collaboration | 1 |
| Data collection | 1 |
| Formulate problems | 1 |
| Reasoning | 1 |
| Test | 1 |

The remaining dimensions fall closer to programming skills rather than to the definition of CT we decided to focus on. For example, among these dimensions (Table 6) we find conditional statements, iterations, events, sequences, variables, execution of commands, data representation, and operators. While it may be argued which dimensions

are purely related to programming rather than being adaptable in a broader sense to CT, it is clear that dimensions such as conditionals, loops, or programming are concepts that are mostly related to programming and informatics, rather than CT. To this end, debugging can be seen as the act of fixing an error in a mental algorithm/procedure (Shute et al., 2017), and thus can be seen as part of CT given a much inclusive definition.

Table 6. Dimensions analyzed that can be seen as purely related to programming.

| Programming dimension | Appearances | Programming dimension | Appearances |
|---|---|---|---|
| Conditionals / Conditional statements | 6 | Experimentation | 1 |
| Loops / Iterations | 5 | Flow control | 1 |
| Events / Event handling | 4 | Gossip problem | 1 |
| Sequences | 4 | Incremental development | 1 |
| Operators | 3 | Iterative development | 1 |
| Parallelism | 3 | Keyboard input | 1 |
| Variables | 3 | Mix / Remixing | 1 |
| Computational practices and perspectives | 2 | Modeling and design | 1 |
| Data | 2 | Programming | 1 |
| Data representation | 2 | Reuse | 1 |
| Automation | 1 | Simple functions | 1 |
| Choosing effective computational tools | 1 | Sorting | 1 |
| Ciphering | 1 | Stacks | 1 |
| Computational concepts | 1 | Synchronization | 1 |
| Coordination | 1 | Threads | 1 |
| Data ordering | 1 | Trees | 1 |
| Developing modular computational solutions | 1 | Troubleshooting | 1 |
| Encapsulate | 1 | User interactivity | 1 |
| Encoding | 1 | User interface / User interface design | 1 |
| Execute commands | 1 | While conditional | 1 |

## 4. Discussion

This paper reports the assessment tools and the assessed dimensions of 26 European studies conducted between 2016 and 2020. The results conform to the existing literature that shows a variety of CT assessments methodologies (e.g. Çoban, & Korkmaz, 2021) and indicates different CT assessment tools exist that can be categorized in five groups: questionnaires, tests/tasks, observations, interviews and analysis of products. The first two categories (questionnaires and test/tasks) were the two most common. Most studies use a single form of assessment (either questionnaire, test, interview, etc...) and often limit themselves to assess if students can recognize and recall knowledge out of context. These forms therefore do not allow assessment of the competences and in particular CT skills that have a multifaceted nature. This can also be noticed analysing the dimensions of CT that the reviewed tools allow to assess. Wing (2006) defined CT as a fundamental skill, a definition further expanded by Shute et al. (2017), clearly decoupling it from basic computer science. Nonetheless most of the dimensions that the review tools assess are related to programming skills, rather than effectively measuring the ability to solve problems through CT which goes beyond computer science. Our review focused on European K-12 education confirms some of the results found in other reviews carried out by researchers in non-European countries: a breadth of methods employed to assess CT (Cutumisu et al., 2019) and the majority of them analysing concepts directly related to algorithms and programming (Tang et al. 2020). The need for tools that allow to assess all facets of CT has been discussed already in other studies such as highlighted in Basu, Rutstein, Xu, Wang and Shear, 2021. This is also related to the different operational definitions of CT making it difficult to agree on the dimensions of CT and to develop a common and reliable assessment tool (Adams et al., 2019). In fact, in the 26 studies selected, as many as 59 different dimensions are associated with CT, however only 21 dimensions appear more than once, while the remaining 38 are representative of a single tool. The analysis of the tools also shows that they do not refer to a shared competence model of CT differentiated by age. The different studies assess CT in pupils in different grades, however it is not clear what competences pupils should reach at which age since the same

dimensions and tools are used for pupils of different ages. The challenges in the development of assessment methods and frameworks that include all facets of CT is already mentioned by different authors (e.g. Brennan & Resnick, 2012; Denner, Werner, & Ortiz, 2012; Denning, 2017; Fronza, El Ioini, & Corral, 2017; Grover et al., 2017; Grover, Pea, & Cooper, 2015; Tikva & Tambouris, 2021; Zhong, Wang, Chen, & Li, 2016) are still present and should drive the need for future research in the field.

## 5. Conclusion

In this paper we focused on answering two main questions: “Which tools are used to assess CT in Europe?” and “Which dimensions of CT are assessed?”. In her 2006 paper, Wing describes CT as "[...] solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science". The dimensions shown in Table 6 can be associated with the fundamental concepts of computer science, yet on an abstract level they can also be used to describe human behavior when solving a problem (Voskoglou & Buckley, 2012). We argue that the presented tools rely too much on computer science concepts rather than focusing on problem-solving skills in educational contexts (Rahman, 2019).

Based on these reflections we can formulate following issues that are present in the assessment of CT in European K-12 education and should drive future research in the field:

- A common operational definition of CT is still absent;
- A competence model that indicates which competences students should reach in CT at which age is absent;
- Still missing is a tool that takes into consideration all the different dimensions of CT and does not focus only on a few of them or just on programming skills.

In order to advance in this research field, we believe it would be important to define a competence model of CT according to the pupil’s age, i.e., a model of the pupil’s skills, knowledge and possible behaviours in a given context. Based on this model, assessment rubrics (Popham, 1997) could be defined.

An assessment rubric consists, in general, in a qualitative description of possible observable behaviours that can be observed during the accomplishment of a task, corresponding to different performance levels with respect to the components of the competence being assessed. The different performance levels could be expressed in different ways (Dawson, 2017), for example, they could be defined through the amount of assistance needed during the resolution of the task. In this case, during the activity students can have access to different aid; the more aid they need, in form of hints, suggestions, or supplementary tools or instruments to produce an acceptable solution to the given problem, the less competent they are. With the help of these rubrics, students could potentially be assessed by being given tasks to be solved that include all dimensions of a shared operational definition of CT.

## Statements on Ethics & Conflicts-References

This research did not receive any specific grant from funding agencies. Authors report no conflict of interest. All authors have equally participated in this article.

## References

Académie des Sciences (2013). *L’enseignement de l’informatique En France - Il Est Urgent de Ne Plus Attendre*. Paris: Académie des Sciences.

Adams, C., Cutumisu, M. & Chang, L. (2019). Measuring K-12 Computational Thinking Concepts, Practices and Perspectives: An Examination of Current CT Assessments. In *Proceedings of Society for Information Technology & Teacher Education International Conference 2019*, (pp. 275–85). Las Vegas, NV, United States: Association for the Advancement of Computing in Education (AACE). https://www.learntechlib.org/p/207654.

*Allsop, Y. (2019). Assessing Computational Thinking Process Using a Multiple Evaluation Approach. *International Journal of Child-Computer Interaction 19*, 30–55. https://doi.org/10.1016/j.ijcci.2018.10.004.

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

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning &*

*Leading with Technology, 38*(6), 20–23. https://files.eric.ed.gov/fulltext/EJ918910.pdf

Basu, S., Rutstein, D.W., Xu, Y., Wang, H. & Shear, L. (2021). A principled approach to designing computational thinking concepts and practices assessments for upper elementary grades. *Computer Science Education, 31*(2), 169-198. DOI:10.1080/08993408.2020.1866939

*Bati, K., Yetişir, M. I., Çalişkan, I. & Güneş, G. (2018). Teaching the Concept of Time: A Steam-Based Program on Computational Thinking in Science Education. *Cogent Education, 5*(1), 1–16. https://www.tandfonline.com/doi/abs/10.1080/2331186X.2018.1507306.

Berland, M., & Wilensky, U. (2015). Comparing Virtual and Physical Robotics Environments for Supporting Complex Systems and Computational Thinking. *Journal of Science Education and Technology, 24*, 628–47. https://doi.org/10.1007/s10956-015-9552-x.

Bocconi, S., Chioccariello, A., & Earp, J. (2018).The nordic approach to introducing computational thinking and programming in compulsory education. *Report prepared for the Nordic@BETT2018Steering Group*. Retrieved from http://edudoc.ch/record/131448doi: 10.17471/54007

Bocconi, S., Chioccariello, A., Dettori, G. Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). *Developing Computational Thinking in Compulsory Education – Implications for Policy and Practice*. JRC Science for Policy Report. Luxembourg: Publications Office of the European Union. https://doi.org/10.2791/792158.

Brennan, K., & Resnick, M. (2012). *New Frameworks for Studying and Assessing the Development of Computational Thinking*. Vancouver, Canada. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf.

*Bryndová, L., & Mališů, P. (2020). Assessing the Current Level of the Computational Thinking Within the Primary and Lower Secondary School Students Using Educational Robotics Tasks. In *2020 the 4th International Conference on Education and Multimedia Technology*, (pp. 239–43). ICEMT 2020. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3416797.3416819.

Calmet, C., Hirtzig, M., & Wilgenbus, D. (2016). *1, 2, 3... Codez !: Enseigner l'informatique à l'école Et Au Collège (Cycles 1, 2 Et 3)*. Éducation. Paris: Editions Le Pommier.

*Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V. & Tosto, C. (2018). Exploring the Effect of a Robotics Laboratory on Computational Thinking Skills in Primary School Children Using the Bebras Tasks. In *Proceedings of the 6th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2018)*, (pp. 27–30). https://doi.org/10.1145/3284179.3284186.

*Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V. & Tosto, C. (2019). Educational Robotics in Primary School: Measuring the Development of Computational Thinking Skills with the Bebras Tasks. *Informatics, 6*, 43. https://doi.org/10.3390/informatics6040043.

Chioccariello, A., & Olimpo, G. (2017). Editorial. *Italian Journal of Educational Technology, 25*, 3–6. https://doi.org/10.17471/2499-4324/986.

Çoban, E., & Korkmaz, Ö. (2021). An alternative approach for measuring computational thinking: Performance-based platform. *Thinking Skills and Creativity, 42*, 100929. https://doi.org/10.1016/J.TSC.2021.100929

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking. A guide for teachers*. Computing At School. Retrieved from https://eprints.soton.ac.uk/424545/1/150818_Computational_Thinking_1_.pdf

CSTA and ISTE. (2011). *Operational Definition of Computational Thinking for k–12 Education*. 2011. https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf.

Cutumisu, M., Adams, C., & Chang L. (2019). A Scoping Review of Empirical Research on Recent Computational Thinking Assessments. *Journal of Science Education and Technology, 28*, 651–76. https://doi.org/10.1007/s10956-019-09799-3.

Da Cruz Alves, N., Gresse Von Wangenheim, C. & Hauck, J.C.R. (2019). Approaches to Assess Computational Thinking Competences Based on Code Analysis in K-12 Education: A Systematic Mapping Study. *Informatics in Education, 18*(1), 17–39. https://doi.org/10.15388/infedu.2019.02.

Dawson, P. (2017). Assessment Rubrics: Towards Clearer and More Replicable Design, Research and Practice. *Assessment & Evaluation in Higher Education, 42*(3), 347–60. https://doi.org/10.1080/02602938.2015.1111294.

*del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J.A. (2020). Computational Thinking Through Unplugged Activities in Early Years of Primary Education. *Computers & Education, 150*, 103832. https://doi.org/https://doi.org/10.1016/j.compedu.2020.103832.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education, 58*(1), 240–249. https://doi.org/10.1016/j.compedu.2011.08.006

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM, 60*(6), 33–39. https://doi.org/10.1145/2998438

European Commission. (2016). *A New Skills Agenda for Europe. Working Together to Strengthen Human Capital, Employability and Competitiveness*. Brussels: European Commission. https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX\%3A52016DC0381.

*Fagerlund, J., Häkkinen, P. Vesisenaho, M., & Viiri, J. (2020). Assessing 4th Grade Students Computational Thinking Through Scratch Programming Projects. *Informatics in Education*, 611–40. https://doi.org/10.15388/infedu.2020.27.

*Förster, E.-C., Förster, K.-T. & Löwe, T. (2018). Teaching Programming Skills in Primary School Mathematics Classes: An Evaluation Using Game Programming. In *2018 IEEE Global Engineering Education Conference (EDUCON)*, (pp. 1504–13). https://doi.org/10.1109/EDUCON.2018.8363411.

Fronza, I., El Ioini, N., & Corral, L. (2017). Teaching computational thinking using agile software engineering methods: A framework for middle schools. *ACM Transactions on Computing Education, 17*. https://doi.org/10.1145/3055258

*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. https://doi.org/10.1080/10494820.2017.1337036.

*Gillott, L., Joyce-Gibbons, A., & Hidson, E. (2020). Exploring and Comparing Computational Thinking Skills in Students Who Take GCSE Computer Science and Those Who Do Not. *International Journal of Computer Science Education in Schools, 3*, 3–22. https://doi.org/10.21585/ijcses.v3i4.77.

Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education, 17*. https://doi.org/10.1145/3105910

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., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education, 25*(2), 199–237. https://doi.org/10.1080/08993408.2015.1033142

Israel, M., Pearson, J.N., Tapia, T., Wherfel, Q.M., & Reese, G. (2015). Supporting All Learners in Schoolwide Computational Thinking: A Crosscase Qualitative Analysis. *Computers & Education, 82*, 263–79. https://doi.org/10.1016/j.compedu.2014.11.022.

*Kalliopi, K., & Michail, K. (2019). Assessing Computational Thinking Skills at First Stages of Schooling. In *Proceedings of the 2019 3rd International Conference on Education and e-Learning*, (pp. 135–39). ICEEL 2019. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3371647.3371651.

Korkmaz, Ö., Çakır, R. & Özden, Y.M. (2016). Computational Thinking Levels Scale (CTLS) Adaptation for Secondary School Level. *Gazi Journal of Educational Science, 1*(2), 143–62.

*Korucu, A. T., Gencturk, A. T. & Gundogdu, M. M. (2017). Examination of the Computational Thinking Skills

of Students. *Journal of Learning and Teaching in Digital Age, 2*(1), 11–19. https://eric.ed.gov/?id=ED572684.

*Kožuh, I., Krajnc, R., Hadjileontiadis, L. J. & Debevc, M. (2018). Assessment of Problem Solving Ability in Novice Programmers. *PLoS ONE, 13*(9). https://doi.org/10.1371/journal.pone.0201919.

*Kukul, V., & Karatas, S. (2019). Computational Thinking Self-Efficacy Scale: Development, Validity and Reliability. *Informatics in Education, 18*(1), 151–64. https://doi.org/10.15388/infedu.2019.07.

*Leifheit, L., Jabs, J., Ninaus, M., Moeller, K. & Ostermann, K. (2018). Programming Unplugged: An Evaluation of Game-Based Methods for Teaching Computational Thinking in Primary School. In *European Conference on Game-Based Learning (ECGBL)* (pp. 344–53). https://www.scopus.com/inward/record.uri?eid=2-s2.0-85058927915\&partnerID=40\&md5=3c9851a83ff1c00f11bc838a9ffa8587.

*Lockwood, J., & Mooney, A. (2018). Developing a Computational Thinking Test Using Bebras Problems. In *Joint Proceedings of the CC-TEL 2018 and TACKLE 2018 Workshops, co-located with 13th European Conference on Technology Enhanced Learning* (EC-TEL 2018). https://www.scopus.com/inward/record.uri?eid=2-s2.0-85053103913\&partnerID=40\&md5=8e374245636670e245ebdafc427904dc.

Lye, S.Y., & Koh, J.H.L. (2014). Review on Teaching and Learning of Computational Thinking Through Programming: What Is Next for K-12? *Computers in Human Behavior, 41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012.

Moreno-León, J., & Robles, G. (2015). Analyze Your Scratch Projects with Dr. Scratch and Assess Your Computational Thinking Skills. In *Proceedings of the 7th international Scratch conference (Scratch2015AMS)*, (pp. 12–15). Amsterdam, Netherlands.

*Pérez-Marín, D., Hijón-Neira, R. & Bacelo, A. (2018). Can Computational Thinking Be Improved by Using a Methodology Based on Metaphors and Scratch to Teach Computer Programming to Children? *Computers in Human Behavior, 105*. https://www.sciencedirect.com/science/article/pii/S0747563218306137.

Petticrew, M., & Roberts, H. (2006). *Systematic Reviews in the Social Sciences: A Practical Guide*. First. London: Blackwell Pub. https://doi.org/10.5860/choice.43-5664.

Popham, W. J. (1997). What's Wrong—and What's Right—with Rubrics. *Educational Leadership, 2nd series, 55*(2), 72–75.

*Price, C. B., & Price-Mohr, R. M. (2018). An Evaluation of Primary School Children Coding Using a Text-Based Language (Java). *Computers in the Schools. 35*(4), 284–301. https://www.tandfonline.com/doi/abs/10.1080/07380569.2018.1531613.

Rahman, M. (2019). 21st Century Skill "Problem Solving": Defining the Concept. *Asian Journal of Interdisciplinary Research, 2*(1), 64–74. https://doi.org/10.34256/ajir1917.

*Roman-Gonzalez, M., Perez-Gonzalez, J.C. & Jimenez-Fernandez, C. (2017). Which Cognitive Abilities Underlie Computational Thinking? Criterion Validity of the Computational Thinking Test. *Computers in Human Behavior, 72*, 678–91. https://doi.org/10.1016/j.chb.2016.08.047.

*Román-González, M., Moreno-León, J. & Robles, G. (2017). Complementary Tools for Computational Thinking Assessment. In *Proceedings of the International Conference on Computational Thinking Education*, (pp. 154–59). http://www.eduhk.hk/cte2017/doc/CTE2017Proceedings.pdf.

*Román-González, M., Pérez-González, J.-C. Moreno-León, J. & Robles G. (2018). Can Computational Talent Be Detected? Predictive Validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction, 18*, 47–58. https://doi.org/10.1016/j.ijcci.2018.06.004.

Royal Society (2012). *Shut Down or Restart? The way forward for Computing in UK Schools*. London: The Royal Society.

*Saez-Lopez, J.M., Roman-Gonzalez, M. & Vazquez-Cano, E. (2016). Visual Programming Languages Integrated Across the Curriculum in Elementary School: A Two Year Case Study Using 'Scratch' in Five Schools. *Computers & Education, 97*, 129–41. https://doi.org/10.1016/j.compedu.2016.03.003.

*Sarıtepeci, M., & Durak, H. (2017). Analyzing the Effect of Block and Robotic Coding Activities on Computational Thinking in Programming Education. In I. Koleva & G. Duman (Eds.), *Educational research and practice*, (pp. 464–73). Sofia: St. Kliment Ohridski University Press.

Schweizerische Eidgenossenschaft. (2017). *Bericht Über Die Zentralen Rahmenbedingungen Für Die Digitale Wirtschaft*. Schweizerische Eidgenossenschaft.

*Segredo, E., Miranda, G. León, C. & Santos, A. (2016). Developing Computational Thinking Abilities Instead of Digital Literacy in Primary and Secondary School Students. In *Smart Education and e-Learning*, (pp. 235–45). https://link.springer.com/chapter/10.1007/978-3-319-39690-3_21.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying Computational Thinking. *Educational Research Review, 22*, 142–58. https://doi.org/10.1016/j.edurev.2017.09.003.

Tang, X., Yin, Y., Lin, Q. Hadad, R. & Zhai, X. (2020). Assessing Computational Thinking: A Systematic Review of Empirical Studies. *Computers & Education, 148*. https://doi.org/https://doi.org/10.1016/j.compedu.2019.103798.

*Tonbuloğlu, B., & Tonbuloğlu, I. (2019). The Effect of Unplugged Coding Activities on Computational Thinking Skills of Middle School Students. *Informatics in Education, 18*, 403–26. https://doi.org/10.15388/infedu.2019.19.

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 edu-cation: A conceptual model based on a systematic literature Review. *Computers & Education, 162*, 104083. https://doi.org/10.1016/j.compedu.2020.104083

Voskoglou, M., & Buckley, S. (2012). Problem Solving and Computational Thinking in a Learning Environment. *Egyptian Computer Science Journal, 36*(4), 28–46.

Wing, J. (2006). Computational Thinking. *Communications of the ACM, 49*, 33–35. https://doi.org/10.1145/1118178.1118215.

World Economic Forum. (2016). *New Vision for Education: Fostering Social and Emotional Learning Through Technology*. World Economic Forum.

*Yildiz Durak, H. (2018). The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving. *Tech Know Learn, 25*, 179–95. https://doi.org/10.1007/s10758-018-9391-y.

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research, 53*. https://doi.org/10.1177/0735633115608444

**Copyrights**