Informatics in Education, 2024, Vol. 23, No. 1, 179–222

DOI: 10.15388/infedu.2024.01

Teaching Computing to Middle and High School Students from a Low Socio-Economic Status Background: A Systematic Literature Review

Ramon Mayor MARTINS, Christiane GRESSE VON WANGENHEIM
Graduate Program in Computer Science, Department of Informatics and Statistics,
Federal University of Santa Catarina, Florianópolis/SC, Brazil
e-mail: ramon.mayor@posgrad.ufsc.br, c.wangenheim@ufsc.br

Received: March 2023

Abstract. Information technology (IT) is transforming the world. Therefore, exposing students to computing at an early age is important. And, although computing is being introduced into schools, students from a low socio-economic status background still do not have such an opportunity. Furthermore, existing computing programs may need to be adjusted in accordance to the specific characteristics of these students in order to help them to achieve the learning goals. Aiming at bringing computing education to all middle and high-school students, we performed a systematic literature review, in order to analyze the content, pedagogy, technology, as well as the main findings of instructional units that teach computing in this context. First results show that these students are able to learn computing, including concepts ranging from algorithms and programming languages to artificial intelligence. Difficulties are mainly linked to the lack of infrastructure and the lack of pre-existing knowledge in using IT as well as creating computing artifacts. Solutions include centralized teaching in assistive centers as well as a stronger emphasis on unplugged strategies. However, there seems to be a lack of more research on teaching computing to students from a low socio-economic status background, unlocking their potential as well to foster their participation in an increasing IT market.

Keywords: computing education, low socio-economic status, Middle school, High school, K-12.

1. Introduction

Computing enables the development of new technology and innovations in every field and has far-reaching applications for the workforce of tomorrow with job numbers expected to grow through 2029 (Zilberman and Ice, 2021; Wyatt *et al.*, 2020). Therefore, to be well-educated citizens in a computing-intensive world and to be prepared for careers in the 21st century, students must have a clear understanding of the principles and practices of computing (CSTA, 2016), including computational thinking and programming (UNESCO, 2019), as well as an understanding of artificial intelligence (AI) (UNESCO, 2022). Thus, it is imperative that all students have the opportunity to learn computing already at school (Gretter *et al.*, 2019; Gresse von Wangenheim *et al.*, 2017). And, although, computing education has been introduced in K-12 in various countries during the last years (Falkner *et al.*, 2019), there still exist significant inequalities regarding computing education of students from a low socio-economic status (SES) background (The World Bank, 2022; Kim and Ryu, 2017; Parker and Guzdial, 2015). Yet, considering that education should be accessible to all students, regardless of their socioeconomic status, gender, ethnicity, or prior technology skills, computing education needs to be democratized by recognizing and circumventing the barriers that exist (Gretter *et al.*, 2019; Margolis *et al.*, 2015).

Recently, some initiatives have emerged to make computing more equal, teaching computing to an underprivileged and underrepresented public aiming at closing the gender gap (Girlswhocode, 2018), or to increase the representation of black girls (BlackgirlsCODE, 2022), black boys (Hidden Genius Project, 2023), black youth (Code2040, 2019), refugees and migrants (RefugeesCode, 2022) or youth from low-income families (ResilientCoders, 2022).

However, despite these initiatives, there still seems not to exist research on what and how to teach, considering the specific characteristics and limitations in the context of youth from a low SES background in middle and high school. Reviews are mostly conducted on computing education in general, such as Garneli *et al.* (2015) who performed a review on programming tools, educational context, and instructional methods used in teaching computing in K-12. Martins-Pacheco *et al.* (2019); Lye and Koh (2014) and Grove and Pea (2013), reviewed teaching computational thinking in K-12. Heintz *et al.* (2016) conducted a review of different models for introducing computing (programming and computational thinking) in K-12 education. However, none of these analyze the demographics or context of students from a low SES background in K-12 computing education. And, although Van der Meulen *et al.* (2021) study the demographics of computing education in K-12 in terms of age/grade, gender, race/ethnic background, location, prior computer science experience, socio-economic status, and disability, this study only provides a general panorama. Parker and Guzdial (2015) analyze existing research aiming at identifying privilege and inequality in STEM and computing education. Yet again, without addressing the characteristics of teaching computing in the context of low SES students. Holloman *et al.* (2021) and Ortiz-Lopez *et al.* (2020), present a systematic literature review on retention and barriers faced by ethnic groups in computing, but focus on higher education only.

Therefore, we performed a systematic literature review focusing on the following research question: Are there instructional units that teach computing to middle and high school students from a low SES background and what are their characteristics? With this review we intend to identify and analyze studies that aim to teach computing, directly or indirectly to middle and high school students in this context. The main contribution of this article is the review of characteristics of instructional units for teaching computing in this context, regarding their content, pedagogy, and technology, as well as the analysis of limitations, consequences, and mitigation actions adopted. In this regard the results of this review provide a first overview on this topic focusing specifically on computing education for young students from a low SES background, a topic not covered so far by other research at this level of detailed analysis. The results of this review offer valuable insights that can guide and facilitate the design and development of instructional units aimed at teaching/learning computing in this context. By providing a comprehensive overview of the main characteristics, limitations and findings in this specific context, this study can provide an initial contribution to the promotion of equality, equity, and inclusion in the field of computing education by assisting educators and researchers in creating more effective pedagogical approaches tailored to the specific needs of these students.

2. Background

2.1. *Computing Education in Middle and High School*

Computing education focuses on the study of computing and how it can be used to solve problems (ISTE, 2022), including several core concepts and practices according to the CSTA K-12 Computer Science Framework (2016) (Table 1).

Due to the recent insertion of Artificial Intelligence in our daily lives, students in K-12 should also learn about the big ideas as suggested by the AI4K12 guidelines (Touretzky *et al.*, 2019) (Table 2).

For teaching algorithms and programming, various types of languages are used such as block-based programming languages like Scratch, as well as text-based programming languages including Python and Java (Hsu *et al.*, 2018; Da Cruz-Pinheiro *et al.*, 2018). Artificial intelligence concepts are also taught by either using block-based languages, such as Snap!, visual workflow-based tools, such as Google Teachable Machine or text-based languages in Jupyter Notebooks (Martins and Gresse von Wangenheim, 2022; Gresse von Wangenheim *et al.*, 2021).

To teach these concepts in middle and high school, several pedagogical approaches are adopted (Hsu *et al.*, 2018) (Table 3). These approaches are mostly based on con-

Table 1

Relevant concepts and practices for computing education in K-12 (CSTA, 2016)

Concepts	Practices
• Computing Systems • Networks and the Internet • Data and Analysis • Algorithms and Programming • Impacts of Computing	• Fostering an Inclusive Computing Culture • Collaborating Around Computing • Recognizing and Defining Computational Problems • Developing and Using Abstractions • Creating Computational Artifacts • Testing and Refining Computational Artifacts • Communicating About Computing

Table 2

Student competencies in Artificial Intelligence based on Touretzky *et al.* (2019)

Big ideas	Student competencies
Perception	Understanding that machine perception of spoken language or visual imagery requires extensive domain knowledge.
Representation	Understanding the concept of representation and understanding that computers construct representations using data, and these representations can be manipulated by applying reasoning algorithms that derive new information from what is already known.
Learning	Understanding that machine learning is a kind of statistical inference that finds patterns in data.
Natural Interaction	Understanding that while computers can understand natural language to a limited extent, at present they lack the general reasoning and conversational capabilities of even a child.
Societal Impact	Ability to identify ways that artificial intelligence is contributing to their lives as well as that the ethical construction of artificial intelligence systems requires attention to the issues of transparency and fairness.

Table 3

Overview of pedagogical approaches for computing education in middle and high school

Pedagogical Approaches	Description
Active learning	An approach that involves students actively in learning, rather than just passively listening to lectures (Sanusi and Oyelere, 2020; Prince, 2004).
Collaborative learning	An approach in which students in groups become responsible for one another's learning as well as their own by completing tasks together (Teague and Roe, 2008; Hsu *et al.*, 2018).
Experiential learning	An approach that involves concrete experiences, reflection on these experiences and later a formation of abstraction of concepts and generalizations (Kolb, 1984).
Game-based learning	An approach in which students learn through games, which are designed to promote engagement, motivation, and active participation in the learning process.
Interactive learning	An approach in which students engage in an active and collaborative learning process through the use of interactive technology, tools, and activities.
Kinesthetic learning	An approach in which students learn through physical activities, such as movement, touch, and manipulation of objects (Begel *et al.*, 2004).
Problem-based learning	A student-centered approach in which students learn about a subject by working in groups to solve an open-ended problem.
Project-based learning	An approach that organizes learning around projects that involve students to work relatively autonomously over extended periods of time in design, problem-solving, decision-making, and investigative activities (Jones *et al.*, 1997; Hsu *et al.*, 2018).

structionism, emphasizing the idea that the student learns by building something meaningful (Papert, 1990).

Diverse instructional methods are used varying from direct instruction (e.g., lecture) to interactive studies (e.g., discussion) (Da Cruz-Pinheiro *et al.*, 2018; Saskatchewan Education, 1991) (Table 4).

Table 4
Instructional methods for computing education in middle and high school

Instructional method	Description
Discussion	The action or process of talking about something in order to reach a decision or to exchange ideas.
Group works	A collaborative learning environment in which students work through problems and assessments together.
Hands-on activity	Students participate or carry out activities relating to subject material rather than listening to a lecture (Ekwueme *et al.*, 2015).
Lecture	A teacher-centered method, in the form of an educational talk.
Think activity	A critical thinking exercise, aimed at helping students analyze and evaluate arguments through a systematic process of identifying claims, reasons, evidence, and counterarguments, and evaluating their strengths and weaknesses (Chaffee, 2019).
Unplugged activity	Teaching concepts by using constructivist, often kinesthetic, activities without computers or digital devices (Conde *et al.*, 2017).

In accordance with pedagogical approaches and content, diverse types of instructional materials are used, such as videos, exercises, tutorials, and code samples, among others (Cruz-Pinheiro *et al.*, 2018; Lye and Koh, 2014), including cards, robots, etc. (Conde *et al.*, 2017).

The students' learning is usually assessed by the instructor through observations, quizzes, interviews, etc. (Da Cruz-Pinheiro *et al.*, 2018). In some cases, performance-based assessments are performed based on artifacts created by students (e.g., software, machine learning models) as results of the learning process (Gresse von Wangenheim *et al.*, 2022).

2.2. *Defining Youth from a Low Socio-Economic Status Background*

The focus of this research is on middle and high school students from a low socio-economic status background that have less access to financial, educational, and infrastructure resources than those with a higher socioeconomic status (UNESCO, 2017). Students from such a background coming from families with low-income are often much less likely to have access to higher quality education and often frequent public schools (OECD PISA, 2019). These students are also generally less likely to have a computer at home or at their school than students coming from higher-income families (UNICEF, 2021).

However in literature often diverse terms are used to characterize this target audience. This includes the term ''underprivilege'' that is understood as an unintended and unsolicited disadvantage gained in the way society views an aspect of a student's identity, such as citizenship, race, ethnicity, gender, socioeconomic status, and language (Parker and Guzdial, 2015; UNRISDSP, 2015) (Table 5).

Table 5
Terms associated with the context of underprivileged students

Associated terms		References
By socioeconomic status	• Disadvantage groups • Vulnerability • Marginalized population • Poverty • Inequality • Low-income families	(Parker and Guzdial, 2015; UNRISDSP, 2015)
By underrepresentation and minority representation	• Ethnic groups • Gender groups • Deficient groups	(Kim and Ryu, 2017; Parker and Guzdial, 2015; UNRISDSP, 2015)
By social situation	• Refugee • Public school students	(ITWORX Education, 2023; UNRISDSP, 2015)
By residence area	• Criminal suburbs residents • Rural or Tribal area residents • Slum residents	(Byker *et al.*, 2014; UNRISDSP, 2015)

Kim and Ryu, (2017) reinforce the use of the term underprivileged for students who are defined as economically poor and having a low quality of life. Byker *et al.* (2014) use the term underprivileged to point to children, who are often victims of the inequalities and injustices that characterize life in India's slums. ITWORX Education (2023) mentions the economic obstacles of underprivileged refugee children in Lebanon due to limited school capabilities, transportation difficulties, curriculum and language differences, discrimination, and bullying. Kim and Ryu (2017) explain that most students with low socio-economic status come from multicultural families in rural areas. Often the term low socioeconomic status is also associated with underrepresentation. In the case of the U.S., the context of low socioeconomic status is often tied to the underprivileged students of racial or ethnic groups, such as African Americans, Hispanics, and Native Americans. In China, students from a low SES background are those considered underprivileged, with unstable social status due to political, economic, and social disability or deficiency (Kim and Ryu, 2017).

Thus, considering this variety of terms used in this context and sometimes considering several specific conditions at the same time, we also take into consideration other terms during the systematic literature review, although focusing specifically on students from a low SES background.

3. Definition and Execution of the Systematic Literature Review

Aiming at understanding how to teach computing effectively to youth from a low SES background, we conduct a systematic literature review, following the procedure defined by Kitchenham and Charters (2007). According to this procedure, we defined the research and analysis questions that reflect the study goals and delimit the research scope. We defined a review protocol defining the sources, search string, and selection criteria. Following the review protocol, we executed the searches and selected relevant results according to the defined inclusion, exclusion, and quality criteria. Once selected relevant articles, we extracted information related to the analysis questions, following the defined extraction strategy. Analyzing the extracted data, we examined and synthesized the results reported, and discussed the results.

3.1. *Definition of the Review Protocol*

The research question is: Are there instructional units that teach computing to middle and high school students from a low socio-economic background and what are their characteristics?

This research question is decomposed in following analysis questions:

- **AQ1**. Which instructional units exist or were taught in a low SES background?
- **AQ2**. What are the context characteristics of the instructional units?
- **AQ3**. Which competences are taught in the instructional units?
- **AQ4.** What are the instructional characteristics of the units?
- **AQ5**. Which limitations were observed due to the low SES background, what were the consequences and how have they been approached?
- **AQ6**. What was the perceived quality of the instructional units?

Data sources. We examined articles or material published in English-language that are available on the Web via prominent digital libraries and databases in the field of computing, including ACM Digital Library, IEEE Xplore, arXiv, Scopus, SocArXiv, ERIC (U.S. Dept. of Education), ScienceDirect and SpringerLink. In addition, Google Scholar and Google searches were performed to complement the search, minimizing the risk of omission (Piasecki *et al.*, 2018).

Inclusion/exclusion criteria. We considered any artifact that presents findings related to an instructional unit (course, workshop, summer camp, curricula) that covers teaching computing, directly or indirectly to middle and high school students from a low SES background (Table 6).

Quality criteria. We considered only material which provides substantial information regarding the teaching of computing, directly or indirectly to middle and high school students from a low SES background, indicating, for example, content, pedagogical approaches, the technology used, as well as limitations and mitigation actions.

Definition of the search string. The search string was based on contextualized keywords and composed of concepts related to the research question, including synonyms, as indicated in Table 7. The definition of the keywords has been calibrated based on several informal searches to minimize the risk of omission.

We defined a generic search string, using wildcard characters to cover as many variations of the terms as possible, and adjusted the string in conformance with the specific syntax of each data source, as presented in Table 8:

```
("machine learning" OR "artificial intelligence" OR "computer science" OR "computational thinking" OR "computing" OR "coding" OR "programming") AND ("k-12" OR school*) AND (teach* OR education OR course OR learn*) AND ("low income" OR "poverty" OR "poor" OR "socioeconomic" OR "inequality" OR "vulnerability" OR "underprivileged" OR "disadvantaged students" OR "slum" OR "criminal suburbs" OR "digital divide" OR "marginalized communities").
```

Table 6
Inclusion/exclusion criteria

	Inclusion	Exclusion
Focus	Teaching computing (including computational thinking, programming, user interface design or Artificial Intelligence, etc.)	Teaching only the usage of digital devices and software programs (e.g., text editors, spreadsheets)
Context	Instructional units designed or applied to youth from a low SES background with low access to financial, educational or infrastructure resources	Inequality due to gender, race, age
Content	Presentation of any kind of instructional unit or curricula for teaching/ learning computing in a low SES context	No presentation of any kind of content, pedagogy, technology, limitations and findings related to teaching computing in a low SES context
Educational stage	Middle and high school	Other educational stages or teachers
Publication language	English	Other languages, e.g., Chinese, Portuguese, etc.
Type of publication	Scientific articles in journals, conferences, online repositories, internet, as well as academic works, such as dissertations, theses, etc.	Blogs, videos, or tools without further description of an instructional unit

Table 7
Search terms

Main concepts	Synonyms
Computing	machine learning, artificial intelligence, computer science, computational thinking, coding, programming
School	k-12
Education	teach, course, learn
Low socio-economic status	underprivileged, low income, poverty, poor, inequality, vulnerability, disadvantage students, slum, criminal suburbs, digital divide, marginalized communities

Table 8
Search string per data source

Source	Search string
ACM Digital Library	[[Abstract: "machine learning"] OR [Abstract: "artificial intelligence"] OR [Abstract: ""] OR [Abstract: "computer science"] OR [Abstract: "computing"] OR [Abstract: "coding"] OR [Abstract: "programming"]] AND [[Abstract: "k-12"] OR [Abstract: school*]] AND [[Abstract: teach*] OR [Abstract: education] OR [Abstract: course] OR [Abstract: learn*]] AND [[Abstract: "low income"] OR [Abstract: "poverty"] OR [Abstract: "poor"] OR [Abstract: "socioeconomic"] OR [Abstract: "inequality"] OR [Abstract: "vulnerability"] OR [Abstract: "underprivileged"] OR [Abstract: "disadvantaged students"] OR [Abstract: "slum"] OR [Abstract: "criminal suburbs"] OR [Abstract: "digital divide"] OR [Abstract: "marginalized communities"]]
arXiv	include_cross_list: True; terms: AND abstract=machine learning" OR "artificial intelligence" OR "computer science" OR "computational thinking" OR "computing" OR "coding" OR "programming"; AND abstract="k-12" OR school*; AND abstract=teach* OR education OR course OR learn*; AND abstract="low income" OR "poverty" OR "poor" OR "socioeconomic" OR "inequality" OR "vulnerability" OR "underprivileged" OR "disadvantaged students" OR "slum" OR "criminal suburbs" OR "digital divide" OR "marginalized communities"
ERIC (U.S. Dept. of Education)	abstract:((machine learning" OR "artificial intelligence" OR "computer science" OR "computational thinking" OR "computing" OR "coding" OR "programming") AND ("k-12" OR school*) AND (teach* OR education OR course OR learn*) AND ("low income" OR "poverty" OR "poor" OR "socioeconomic" OR "inequality" OR "vulnerability" OR "underprivileged" OR "disadvantaged students" OR "slum" OR "criminal suburbs" OR "digital divide" OR "marginalized communities"))
Google	*Due to limitations of the Google search engine a reduced search string has been used:* "artificial intelligence" "computer science" "k-12" "learn" "teach" "low income" "poverty" "socioeconomic"
Google Scholar	("machine learning" OR "artificial intelligence" OR "deep learning" OR "data science") AND ("high school" OR "k-12" OR teen* OR school*) AND (teach* OR education OR course OR MOOC OR learn*)
IEEE Xplore	(("Abstract":"machine learning") OR ("Abstract":"artificial intelligence") OR ("Abstract":"deep learning") OR ("Abstract":"data science")) AND (("Abstract":"high school") OR ("Abstract":"k-12") OR ("Abstract":teen*) OR ("Abstract":school*)) AND (("Abstract":teach*) OR ("Abstract":education) OR ("Abstract":course) OR ("Abstract":MOOC) OR ("Abstract":learn*)) Filters Applied: 2011–2021
ScienceDirect (Elsevier)	*Due to limitations of the ScienceDirect search engine a reduced search string has been used:* (("artificial intelligence" OR "computer science") AND ("k-12") AND (learn OR teach) AND ("low income" OR "poverty" OR "socioeconomic"))
Scopus	(TITLE-ABS-KEY (machine learning" OR "artificial intelligence" OR "computer science" OR "computational thinking" OR "computing" OR "coding" OR "programming") AND TITLE-ABS-KEY ("k-12" OR school*) AND TITLE-ABS-KEY (teach* OR education OR course OR learn*) AND TITLE-ABS-KEY ("low income" OR "poverty" OR "poor" OR "socioeconomic" OR "inequality" OR "vulnerability" OR "underprivileged" OR "disadvantaged students" OR "slum" OR "criminal suburbs" OR "digital divide" OR "marginalized communities")) AND (LIMIT-TO (SUBJAREA , "COMP"))

Continued on next page

Table 8 – continued from previous page

Source	Search string
SocArXiv	(machine learning" OR "artificial intelligence" OR "computer science" OR "computational thinking" OR "computing" OR "coding" OR "programming") AND ("k-12" OR school*) AND (teach* OR education OR course OR learn*) AND ("low income" OR "poverty" OR "poor" OR "socioeconomic" OR "inequality" OR "vulnerability" OR "underprivileged" OR "disadvantaged students" OR "slum" OR "criminal suburbs" OR "digital divide" OR "marginalized communities")
SpringerLink	(machine learning" OR "artificial intelligence" OR "computer science" OR "computational thinking" OR "computing" OR "coding" OR "programming") AND ("k-12" OR school*) AND (teach* OR education OR course OR learn*) AND ("low income" OR "poverty" OR "poor" OR "socioeconomic" OR "inequality" OR "vulnerability" OR "underprivileged" OR "disadvantaged students" OR "slum" OR "criminal suburbs" OR "digital divide" OR "marginalized communities")' Filter: within Computer Science
Wiley Online Library	machine learning" OR "artificial intelligence" OR "computer science" OR "computational thinking" OR "computing" OR "coding" OR "programming" " in Abstract and "k-12" OR school*" in Abstract and "teach* OR education OR course OR learn*" in Abstract and ""low income" OR "poverty" OR "poor" OR "socioeconomic" OR "inequality" OR "vulnerability" OR "underprivileged" OR "disadvantaged students" OR "slum" OR "criminal suburbs" OR "digital divide" OR "marginalized communities"" in Abstract

3.2. *Search Execution*

The search was realized in January 2023 by the first author and revised by the co-author. The initial search returned 31,565 artifacts. For analysis we focused on the 300 most relevant results from each search. In the first step, we reviewed titles, abstracts, and keywords to identify articles that adhere to the exclusion criteria, resulting in 52 potentially relevant artifacts (Table 9).

In the next step, we analyzed the full texts and excluded irrelevant ones following the inclusion/exclusion and quality criteria. We also excluded articles describing instructional units targeting undergraduate or graduate level (e.g., Camp *et al.*, 2021; Codding *et al.*, 2019), other K-12 levels such as elementary school (e.g., Chai *et al.*, 2021), or teachers' preparation programs/workshops (e.g., Simmonds *et al.*, 2019). We also excluded articles focused on teaching the usage of computer systems such as text editing (e.g., Li *et al.*, 2016). Applying the quality criteria we also excluded artifacts not providing substantial information with regard to our analysis questions (e.g., Tena-Meza *et al.*, 2021; Varma, 2009; Garg *et al.*, 2020; Judd, 2020; Vergara *et al.*, 2022). Furthermore, some potentially relevant articles that were inaccessible were not considered (e.g., Westin *et al.*, 2016).

In order to further reduce the risk of omission, we also conducted a snowballing procedure (backward and forward) (Wohlin, 2012). As a result, 2 additional relevant articles were encountered.

We then excluded duplicates and articles referring to the same instructional unit were unified. As a result, a total of 15 articles presenting findings and considerations with regard to 14 instructional units were identified (Table 10).

Table 9
Number of identified artifacts per repository and selection stage

Source	No. of search results	No. of analyzed results	No. of potentially relevant results	No. of relevant results (without duplicates)
ACM Digital Library	109	109	14	2
IEEE Xplore	70	70	10	4
arXiv	27	27	1	0
Scopus	226	226	9	2
SocArXiv	9	9	1	0
ERIC	128	128	3	1
ScienceDirect	175	175	0	0
SpringerLink	34	34	2	1
Wiley Online Library	87	87	2	0
Google	12,500	300	5	0
Google Scholar	18,200	300	6	3
Total	31,565	1,465	52	13
Snowballing				
Backward snowballing	2			
Forward snowballing	0			
Total number of relevant results without duplicates				**15**

4. Analysis of the Results

In this section we present the results for each of the analysis questions**.** Detailed information extracted from the encountered publications is presented in Appendix A–F.

4.1. *Which Instructional Units Exist or Were Taught in a Low SES Background?*

As a result of the research, only a small number (14) of instructional units that were either designed or applied for teaching computing to youth from a low SES background in middle and high school were identified (Table 10).

Table 10
Instructional Units

Reference	
(Anuar *et al.*, 2020)	Anuar, N. H., Mohamad, F. S., & Minoi, J.-L. (2020). Contextualising Computational Thinking: A Case Study in Remote Rural Sarawak Borneo. *Int. Journal of Learning, Teaching and Educational Research*, 19(8), 98–116.
(Araya *et al.*, 2021)	Araya, R., Isoda, M., & van der Molen Moris, J. (2021). Developing Computational Thinking Teaching Strategies to Model Pandemics and Containment Measures. *Int. Journal of Environmental Research and Public Health*, 18(23), 12520.

Continued on next page

Table 10 – continued from previous page

Reference	
(Brackmann *et al.*, 2019)	Brackmann, C. P., Barone, D. A. C., Boucinha, R. M., & Reichert, J. (2019). Development of Computational Thinking in Brazilian Schools with Social and Economic Vulnerability: How to Teach Computer Science Without Machines. *Int. Journal of Innovation Education and Research*, 7(4), 79–96.
(Eguchi, 2021)	Eguchi, A. (2021). AI-Robotics and AI Literacy. In *Education in & with Robotics to Foster 21st-Century Skills: Proc. of Edurobotics, Studies in Computational Intelligence*, Online, 75–89.
(Everson *et al.*, 2022)	Everson, J., Kivuva, F. M., & Ko, A. J. (2022). "A Key to Reducing Inequities in Life, AI, is by Reducing Inequities Everywhere First": Emerging Critical Consciousness in a Co-Constructed Secondary CS Classroom. In *Proc. of the 53rd ACM Technical Symposium on Computer Science Education*, Providence, RI, USA, 209–215.
(Harari and Harari, 2017)	Harari, V., & Harari, I. (2017). Teaching programming to kids in situation of social vulnerability. In *Proc. of the 12th Latin American Conference on Learning Technologies*, La Plata, Argentina, 1–8.
(Levy, 2003)(Levy and Paz, 2005)	Levy, D. (2003). Introducing computer science to educationally disadvantaged high school students – the Israeli experience. In *Proc. of the IEEE Symposium on Human Centric Computing Languages and Environments*, Auckland, New Zealand, 269–270. Levy, D., & Paz, T. (2005). The principle of pattern-oriented curriculum and its implementation in a computer science module for high school students. In *Proc. of the IEEE Symposium on Visual Languages and Human-Centric Computing*, Dallas, TX, USA, 321–322.
(Miller *et al.*, 2018)	Miller, J., Raghavachary, S., & Goodney, A. (2018). Benefits of Exposing K-12 Students to Computer Science through Summer Camp Programs. In *Proc. of the IEEE Frontiers in Education Conference*, San Jose, CA, USA, 1–5.
(Nogueira *et al.*, 2021)	Nogueira, V. B., Teixeira, D. G., de Lima, I. A. C. N., Costa, D. A. S., & Oliveira, M. A. D. (2022). Towards an inclusive digital literacy: An experimental intervention study in a rural area of Brazil. *Education and Information Technologies*, 27(3), 2807–2834.
(Outlay, 2016)	Outlay, C. N. (2016). Targeting underrepresented minority and low-income girls for computing camps: early results and lessons learned. *Journal of Computing Sciences in Colleges*, 31(5), 85–94.
(Prasad *et al.*, 2016)	Prasad, R., Traynor, C., & Keeney, S. S. (2016). Hooking them young: Demystifying computer science and technology among underprivileged high school students. In *Proc. of the IEEE Integrated STEM Education Conference*, Princeton, NJ, USA, 149–155.
(Unnikrishnan *et al.*, 2016)	Unnikrishnan, R., Amrita, N., Muir, A., & Rao, B. (2016). Of Elephants and Nested Loops: How to Introduce Computing to Youth in Rural India. In *Proc. of the 15th International Conference on Interaction Design and Children*, Manchester, UK, 137–146.
(Yerousis *et al.*, 2015)	Yerousis, G., Aal, K., von Rekowski, T., Randall, D. W., Rohde, M., & Wulf, V. (2015). Computer-Enabled Project Spaces: Connecting with Palestinian Refugees across Camp Boundaries. In *Proc. of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, Seoul, Republic of Korea, 3749–3758
(Zhang *et al.*, 2022)	Zhang, H., Lee, I., Ali, S., Sargent, J., & Hsu, W. (2022). Integrating Ethics and Career Futures with Technical Learning to Promote AI Literacy for Middle School Students: An Exploratory Study. *International Journal of Artificial Intelligence in Education*.

Analyzing the year of publication we can observe that after a peak in publication in 2016, this topic has again increased in importance recently (Fig. 1).

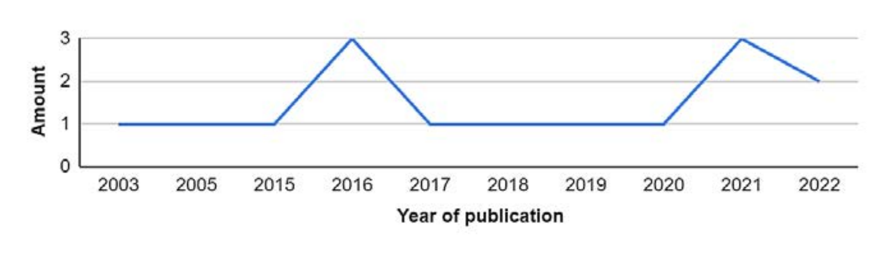

Fig. 1. Amount of publications on teaching computing in a low SES context.

4.2. *What are the Context Characteristics of the Instructional Units?*

The majority of these instructional units (86%) aim at novice students without any pre-existing computing knowledge or skills. Regarding the educational stage, six of the instructional units are exclusively aimed at middle school students (e.g., Zhang *et al.*, 2022; Harari and Harari, 2017) and three instructional units exclusively at high school students (Everson *et al.*, 2018; Levy and Paz, 2005; Prasad *et al.*, 2016) (Fig.2). On the other hand, some instructional units address different educational stages, e.g., elementary to high school, yet providing specific content and activities for each educational stage (Miller *et al.*, 2018; Araya *et al.*, 2021; Eguchi, 2021). However, as these articles do not report the results separately, we consequently classified these studies covering several educational stages in Fig. 2.

The instructional units encountered have been developed and applied mostly on the American continents (n = 10), mainly in the USA (n = 6) (e.g., Zhang *et al.*, 2022) and Brazil (n = 2) (e.g., Nogueira *et al.*, 2021). Further applications were reported in Asia (n = 3) (e.g., Anuar *et al.*, 2020) and the Middle Eastern (n = 2) (e.g., Yerousis *et al.*, 2015) (Fig. 3).

Although focusing on underprivileged students in terms of a low socio-economic status, we observed variations regarding the definition of this term. Instructional units reported from the U.S. focus primarily on minority, underrepresented low-income groups, including specifically girls (e.g., Miller *et al.*, 2018) and ethnically diverse

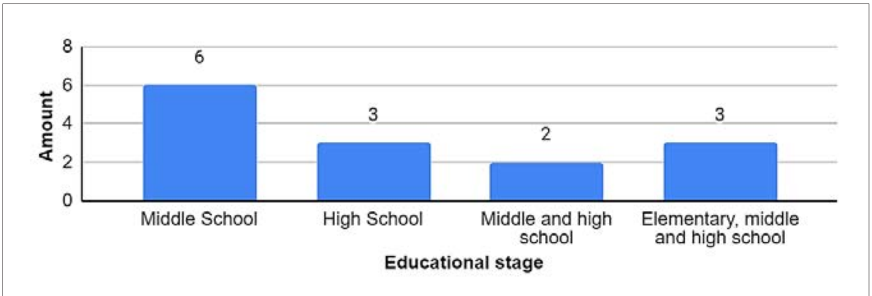

Fig. 2. Amount of instructional units per educational stage.

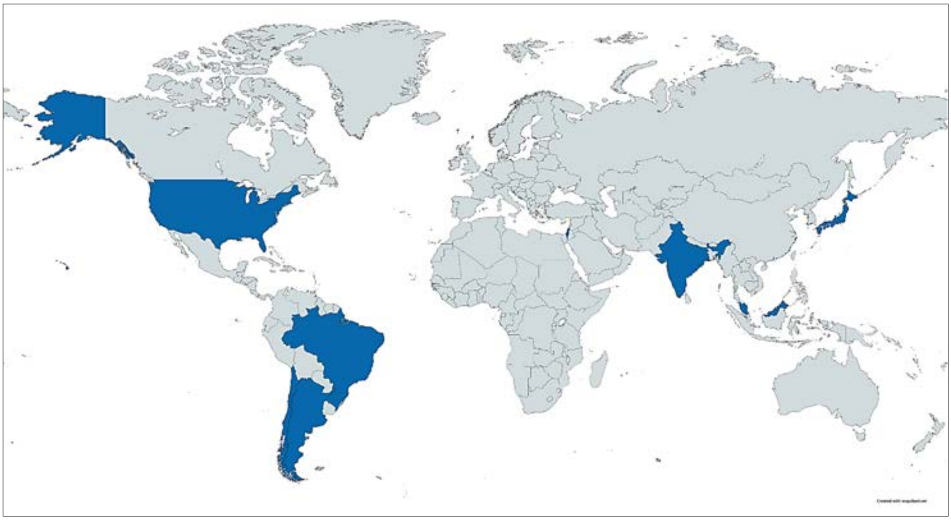

Fig. 3. Countries of reports of teaching computing to students from a low SES background.

audiences (Everson *et al.*, 2022) (Fig. 11). In Brazil, the context of low SES is also tied to the poorer conditions of public schools (Brackman *et al.*, 2019) or students studying in rural areas (Nogueira *et al.*, 2021). In addition, sensitive social contexts such as refugee students in conflict zones in Palestine are also placed in this context (Prasad *et al.*, 2016). Fig. 4 summarizes the terms associated with the context of the students from a low SES background demonstrating the variety of interpretations in literature.

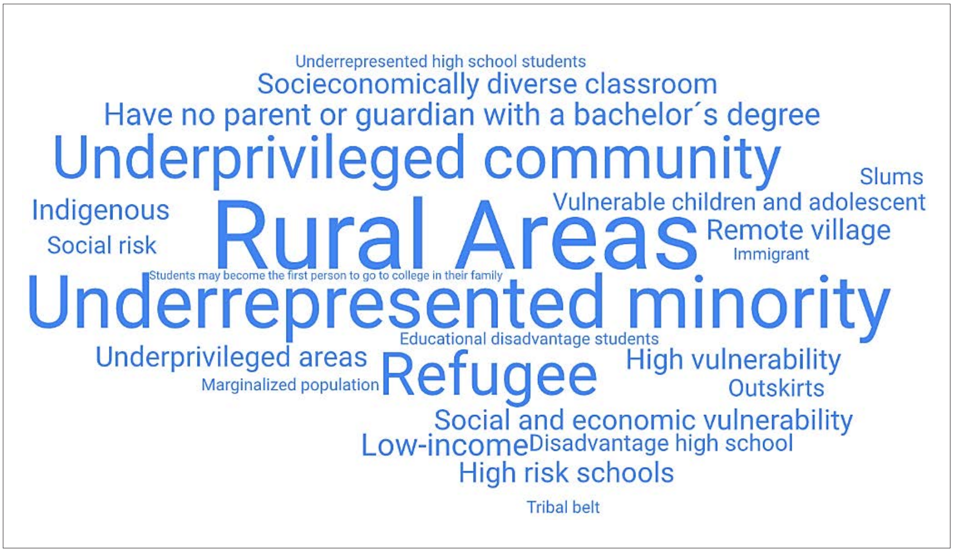

Fig. 4. Overview on the definition of the context of students from a low SES background.

4.3. *Which Competences are Taught in the Instructional Units?*

The large majority (92%) of the encountered instructional units address computational thinking and Algorithms and Programming concepts, (Fig. 5). Related specifically to computational thinking, Levy (2003) and Levy and Paz (2005) address problem decomposition. Concepts of algorithms are covered by Araya *et al.* (2021); programming logic (e.g., Nogueira *et al.*, 2021), Boolean logic (Prasad *et al.*, 2016), pseudocode (Unnikrishnan *et al.*, 2016), concepts of variables (e.g., Anuar *et al.*, 2020), control structures (e.g., Harari and Harari, 2017; Prasad *et al.*, 2016), repetition structures (e.g., Yerousis *et al.*, 2015), and programming concepts such as semantics (Harari and Harari, 2017).

A total of 4 instructional units address the Impacts of Computing (e.g., Nogueira *et al.*, 2021; Outlay, 2016). Recently, due to the popularization of Artificial Intelligence some of the instructional units (n = 5) also teach Data and Analysis concepts.

Four of the instructional units teach Artificial Intelligence with respect to the competencies of the AI4K12 guidelines (Araya *et al.*, 2021; Eguchi, 2021; Everson *et al.*, 2022; Zhang *et al.*, 2022) (Fig. 6). The contents covered varies from basic concepts such as ‘’how artificial intelligence works’’ (e.g., Everson *et al.*, 2022) to unsupervised learning concepts using advanced techniques such as Generative Adversarial Network (GANs) (Zhang *et al.*, 2022). Few instructional units (n = 2) address career aspects and future opportunities (Zhang *et al.*, 2022; Outlay, 2016).

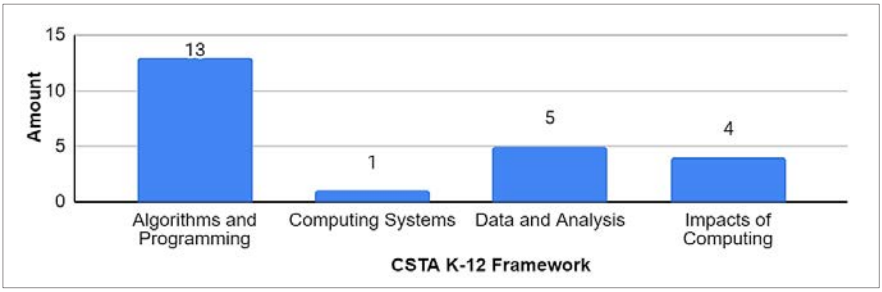

Fig. 5. Amount of instructional units that address CSTA (2016) competencies.

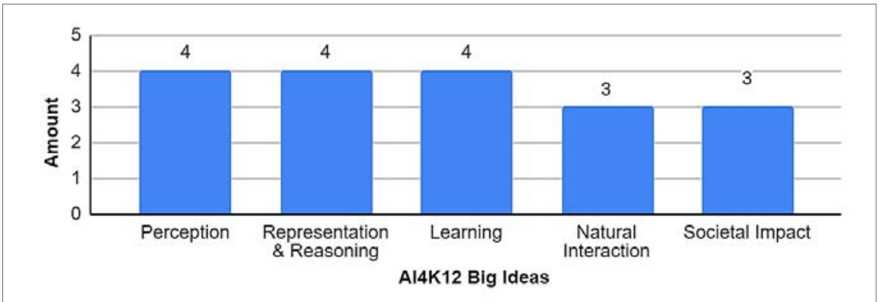

Fig. 6. Amount of instructional units that address AI4K12 Big Ideas (Touretzky *et al.*, 2019).

4.4. *What are the Instructional Characteristics of the Units?*

Type of instructional unit. Most instructional units are applied as extracurricular units, in a less organized and less structured way ranging from one-day camp (Outlay, 2016) to a full school year (Levy, 2003; Levy and Paz, 2005). Three of the encountered publications present only an activity (Anuar *et al.*, 2020; Brackmann *et al.*, 2019; Eguchi, 2021), while one presents a complete curricular unit (Levy and Paz, 2005) (Fig. 7).

Instructional mode. Most of the instructional units (71.4%) are delivered face-to-face. For example, Anuar *et al.* (2020) report the application of the learning activities to students living in a remote rural village in Sarawak Borneo, and Yerousis *et al.* (2015) make use of a computer club to teach Information and Communication Technology to Palestinian refugee students in a marginalized camp. On the other hand, we also encountered reports of remote applications, e.g., Everson *et al.* (2022) due to the pandemia, with the university providing hardware and devices for the students to be able to participate in the course.

Pedagogical approach. Some instructional units (n = 5) use game-based learning as a way to spark interest and motivate students, especially younger ones (e.g., Anuar *et al.*, 2020; Prasad *et al.*, 2016) (Fig. 8). This approach is also combined with unplugged approaches to engage students (e.g., Anuar *et al.*, 2020; Unnikrishnan *et al.*, 2016). Few instructional units make use of experience-, and problem-based learning approaches.

Instructional methods. Faced with teaching young students from a low SES background, several instructional methods are used (Fig. 9). Half of the instructional units (n = 7) use hands-on activities, using, e.g., Google Teachable Machine to allow students to gain knowledge while building their machine learning models (e.g., Zhang *et al.*, 2022; Eguchi,2021). Four instructional units use group work, e.g., to work on sensitive topics such as privacy and ethical implications (Everson *et al.*, 2022). Yerousis *et al.* (2015) also observed that students working in groups feel more confident.

Two instructional units (Zhang *et al.*, 2022; Harari and Harari, 2017) report the adoption of interactive methodologies supported by kinesthetic activities in which students, for example, learn basic programming concepts by simulating being a robot.

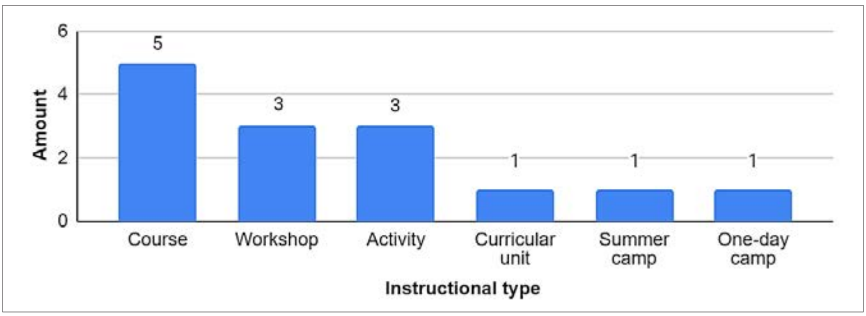

Fig. 7. Amount of instructional units per type.

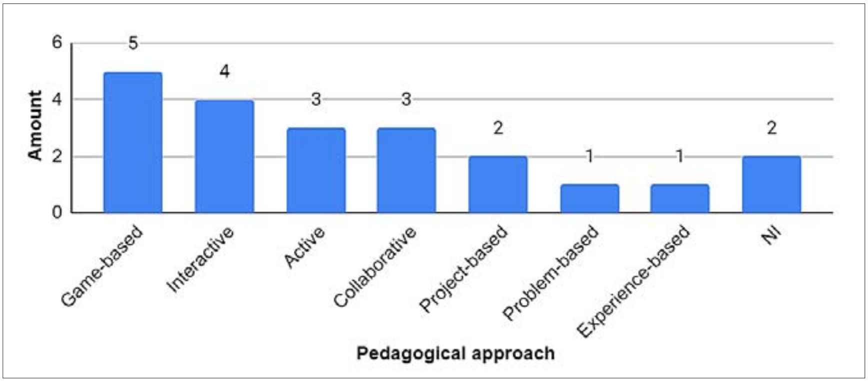

Fig. 8. Amount of instructional units per pedagogical approach.

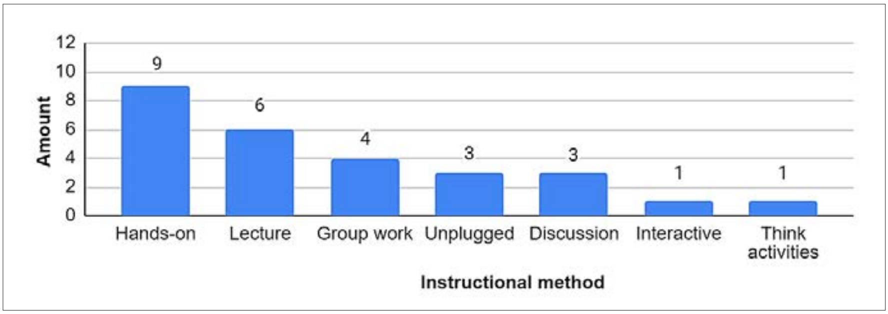

Fig. 9. Amount of instructional units per instructional method.

Instructional material. Diverse types of instructional materials are used to support learning. Three instructional units use ‘’paper and pencil’’ materials to support the unplugged activities (Araya *et al.*, 2020; Brackmann *et al.*, 2019; Unnikrishnan *et al.*, 2016). Videos are used also in three instructional units to support hands-on instructional methodologies (Araya *et al.*, 2020; Harari and Harari, 2017; Zhang *et al.*, 2022). Other materials include slides (n = 2) (Miller *et al.*, 2018; Outlay, 2016) or cards (n = 2) (Anuar *et al.*, 2020; Unnikrishnan *et al.*, 2016).

Technological tools. Almost half of the instructional units (n = 6) adopt the block-based programming language Scratch (e.g., Anuar *et al.*, 2020; Eguchi, 2021; Harari and Harari, 2017; Miller *et al.*, 2018; Yerousis *et al.*, 2015). Other programming languages used include Pascal and Logo in early applications work (e.g., Levy and Paz, 2005), and more recently Java (Miller *et al.*, 2018) or Python (Prasad *et al.*, 2016). Google Teachable Machine is used as a tool to teach artificial intelligence (e.g., Zhang *et al.*, 2022; Eguchi, 2021). In two cases, due to limitations regarding the location of the students, no technological tools are used (Anuar *et al.*, 2020; Unnikrishnan *et al.*, 2016).

Learning assessment is typically done through tests in seven instructional units (e.g., Nogueira *et al.*, 2021; Brackmann *et al.*, 2019) (Fig. 10). Nogueira *et al.* (2021) use tests

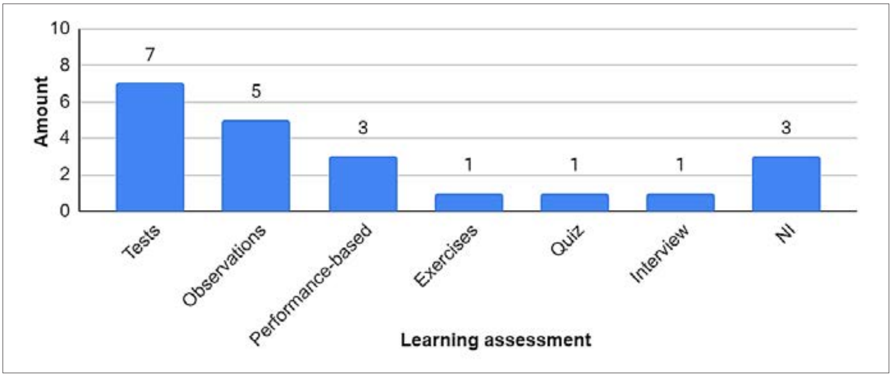

Fig. 10. Amount of publications per type of learning assessment.

to assess logic, and Brackmann *et al.* (2019) to assess computational thinking. Performance-based assessments are adopted in three instructional units based on artifacts created by students either creating a Scratch quiz (Anuar *et al.*, 2020), providing reflective feedback on the concepts taught (Everson *et al.*, 2022) or a student presentation on the project results (Yerousis *et al.*, 2015).

4.5. *Which Limitations were Observed Due to the Low SES Background, What were the Consequences and How Have they been Approached?*

Several limitations were predicted or encountered in the context of teaching computing to students from a low SES background (Table 11). Half of the studies identified the lack of infrastructure at application sites such as public schools (e.g., Brackmann *et al.*, 2010; Harari and Harari 2017), affecting student activities and learning (e.g. Yerousis *et al.*, 2015; Everson *et al.*, 2022), as well as leading to social division (Eguchi *et al.*, 2021), digital division (Yerousis *et al.*, 2015) and digital exclusion of students from a certain region (Nogueira *et al.*, 2021). Solutions adopted include using unplugged activities not requiring digital devices or creating partnerships with universities or other institutions providing devices.

Another limitation observed in six studies is the lack of competencies in basic computing or experience in using digital devices (e.g., Nogueira *et al.*, 2021; Prasad *et al.*, 2016; Unnikrishnan *et al.*, 2016), which can lead to learning difficulties and inequalities. Solutions include adjustments in the pedagogical approach to focus on subjects such as computational thinking (e.g., Nogueira *et al.*, 2021) and STEM (e.g., Prasad *et al.*, 2016).

Other limitations are linked to age heterogeneity and specific contexts of low SES, such as ethnicity, gender, and area of residence (e.g., Harari and Harari, 2017; Outlay, 2016; Unnikrishnan *et al.*, 2016). This can cause learning difficulties (Harari and Harari, 2017) to a mistaken perception of male dominance in IT (Outlay, 2016). Solutions include specific activities for these contexts, such as progressive activities for different age levels (Harari and Harari, 2017) or career discussions for girls (Outlay, 2016).

Table 11

Summary of limitations, consequences, and solutions

Limitations or identified needs	Consequences	Solutions/mitigation actions	References
Lack of infrastructure	This can lead to social division, digital division, and digital exclusion of a region. It can also affect students' activities and engagement	• Use of unplugged activities • Use of more accessible tools • Use of devices facilitated by partner universities or government projects.	(Araya *et al.*, 2021; Brackmann *et al.*, 2019; Eguchi, 2021; Everson *et al.*, 2022; Harari and Harari, 2017; Nogueira *et al.*, 2021; Yerousis *et al.*, 2015)
Lack of pre-existing basic computing knowledge or experience in using digital devices	This can lead to inequalities, social segregation, digital exclusion and division of students	• Co-constructing the course with the students • Using a slow and gentle learning pace • Introducing STEM focused computing with computational thinking	(Everson *et al.*, 2022; Levy, 2003; Levy and Paz, 2005; Nogueira *et al.*, 2021; Prasad *et al.*, 2016; Unnikrishnan *et al.*, 2016; Yerousis *et al.*, 2015)
Age heterogeneity	This affects the students' interest and teaching learning	• Use of different approaches for young and older students • Use of progressive activities from less complex to more complex	(Harari and Harari, 2017) (Miller *et al.*, 2018)
Specific contexts within a low SES background	This creates difficulty in understanding, communicating, and the perception that computing is a field for one gender only	• Use of group-specific approaches • Use of facilitators for communication (indigenous students) • Designing a specific curriculum for a particular group (low-income) to close a gap in access to content knowledge (artificial intelligence). • Teaching about career opportunities to a specific group (low-income girls). • Use of contextualized materials reflecting the students' reality	(Anuar *et al.*, 2020; Araya *et al.*, 2021; Outlay, 2016; Zhang *et al.*, 2022; Unnikrishnan *et al.*, 2016)

4.6. *What was the Perceived Quality of the Instructional Units?*

The majority of the publications report evaluations regarding the quality of the instructional units as well as the achieved learning by the students. The quality factor most analyzed is motivation (n = 4) (e.g., Prasad *et al.*, 2016; Brackmann *et al.*, 2020), followed by learning (n = 2) (Araya *et al.*, 2021; Anuar *et al.*, 2020), self-confidence (n = 2) (Anuar *et al.*, 2020; Nogueira *et al.*, 2021) and enthusiasm (Brackmann *et al.*, 2019; Unnikrishnan *et al.*, 2016). More than half of the studies (n = 8) evaluate the quality of the instructional units in an "ad-hoc" manner without a systematic definition. Four studies adopt a quasi-experimental research design (e.g., Brackmann *et al.*, 2019; Anuar *et al.*, 2020). Data collection is mostly collected through pre- and post-tests (n = 8) (e.g., Zhang *et al.*, 2022; Anuar *et al.*, 2020), observations (n = 3) (e.g., Anuar *et al.*, 2020) or performance-based assessment of artifacts created by students (n = 2)

(Everson *et al.*, 2022; Anuar *et al.*, 2020). Sample size ranges from small (11 students) as reported by Prasad *et al.* (2016) to 200 students (Miller *et al.*,2018).

As a result of the evaluations, several findings were reported. Regarding content, positive findings were mostly related to the learning of programming concepts as observed by Harari and Harari (2017), Unnikrishnan *et al.* (2016), and Anuar *et al.* (2020) (Table 12). Positive findings were also noted with regard to the use of the Scratch environment, enabling the majority of the students to solve the learning tasks (Harari and Harari *et al.*, 2017). On the other hand, Brackmann *et al.* (2019) observed a deficiency in basic mathematics and literacy concepts that can potentially affect the achievement of the learning goals. Teaching a complex subject like artificial intelligence, Zhang *et al.* (2022) reported that 25% of students who started the workshop with an incorrect understanding of artificial intelligence, continued to maintain it. Everson *et al.* (2022) pointed out that when approaching a potentially sensitive topic such as ethics in artificial intelligence, it is necessary to establish a classroom climate in which students feel comfortable to discuss topics related to underrepresented groups.

Findings related to pedagogical approaches indicate that students learned basic programming concepts in a fun and motivating way (Harari and Harari *et al.*, 2017; Prasad *et al.*, 2016) (Table 13). The game-based learning approach was also reported as very effective in some instructional units. Prasad *et al.* (2016) point out that the students were excited while working on the game. Unnikrishnan *et al.* (2016) also observe that childrens can create a game successfully, i.e., create commands to steer an elephant around the board using simple directions, loops, and branches. Brackmann *et al.* (2019) report positive findings using unplugged activities as an alternative for students living in underprivileged areas without access to computers. Anuar *et al.* (2020) note that using contextualized material presenting examples from the students'

Table 12

Overview on findings with respect to content

Positive findings	References
Students learned the basics of programming and programming languages	(Harari and Harari, 2017; Unnikrishnan *et al.*, 2016)
Students were able to accomplish the tasks and learn programming with Scratch	(Harari and Harari, 2017)
Students developed a general understanding of artificial intelligence concepts and processes	(Zhang *et al.*, 2022)
Students improved knowledge in computing and STEM concepts	(Nogueira *et al.*, 2021)
Difficulties	
Very large deficiencies in basic mathematical concepts and literacy were identified, which hindered the achievement of the learning goals	(Brackmann *et al.*, 2019)
About 25% of the students who started the workshop with an incorrect understanding of artificial intelligence, continued to stick to the incorrect understanding	(Zhang *et al.*, 2022)
In order to discuss ethics in artificial intelligence it is necessary to establish a classroom climate that allows to have tense conversations	(Everson *et al.*, 2022)

Table 13

Overview on findings with respect to pedagogical approaches

Positive findings	References
Techniques and materials are positive for learning computing when they are contextualized with examples from the students' reality and culturally adapted	(Anuar *et al.*, 2020)
Students were able to work with several tools, educational resources and programming languages	(Harari and Harari, 2017)
Students learn the basics of programming in a fun, entertaining and motivating way using a game-based learning approach	(Harari and Harari, 2017; Prasad *et al.*, 2016)
Unplugged activities are an effective approach and can be an alternative for students without access to digital devices	(Brackmann *et al.*, 2019)
Using an active learning approach increased student interest in computing and STEM, including girls from low-income families	(Miller *et al.*, 2018; Outlay, 2016; Prasad *et al.*, 2016;
Group work can leave students more confident of their ability to develop the game	(Yerousis *et al.*, 2015)
Difficulties	
Working with children can be tiring to teachers, especially when ideas become complex and feel less like play.	(Unnikrishnan *et al.*, 2016)
Online learning and too short interventions hindered students to achieve the learning goals with respect to in neural network concepts	(Zhang *et al.*, 2022)

reality and culturally adapted contributed to the students' learning. On the other hand, Unnikrishnan *et al.* (2016) point out that working with children can be tiring for teachers, especially when their ideas become more complex. Zhang *et al.* (2022) report that online classes can hinder or make the learning of complex concepts such as neural networks more difficult.

Few findings were reported regarding technological tools (Table 14). For example, Zhang *et al.*, (2022) point out that students engaged in learning machine learning concepts using Google Teachable Machine. On the other hand, Anuar *et al.* (2020) observed that the understanding of indigenous students from a low SES background (who speak Penan/Malay) may have been affected by the Scratch environment not being available in their language.

Table 14

Overview on findings with respect to technological tools

Positive findings	References
The Google Teachable Machine platform helped to engage students to learn concepts on machine learning and supervised learning	(Zhang *et al.*, 2022)
Difficulties	
Lack of availability of Scratch in a native indigenous language may have affected the way the participants understood computing concepts.	(Anuar *et al.*, 2020)

In addition, we also observed some general difficulties. For example, teaching students in a rural school in Brazil may be a challenge, as the students miss many classes (Nogueira *et al.*, 2021). Unnikrishnan *et al.* (2016) point out, there are many intrusions into the classroom (parents, siblings, goats, and even large quadrupeds) while teaching in rural areas in India. Yerousis *et al.* (2015) stated that one of the refugee camps (Camp Jalazone) where the course was given is adjacent to Beit El, an Israeli settlement and active military camp, and classes were frequently canceled as violent clashes erupted between in-camp youngsters and the Israel Defense Force.

5. Discussion

Considering the importance of computing education for middle and high school students, to empower them, and to allow them to break out of their low socio-economic status, very few instructional units were found covering a wide range of contexts, including students from public school, rural areas, specific ethnic groups, refugees, minorities, among others. Most reported instructional units are concentrated in America and Asia, also in accordance with data from the World Bank (2022) and U.S. Department of the Treasury (2022) in which countries like India, Brazil and the USA have many social inequalities.

Basically all instructional units are organized as extracurricular courses and workshops ranging from one-day camps to school year-long curricula aiming at novices from middle school. The units are mostly taught face-to-face, mainly due to the socioeconomic issue of the students and the lack of technological infrastructure, which limits the possibility of remote teaching/learning.

Most instructional units teach competencies that are indicated by the CSTA K-12 Framework, focusing mostly on algorithms & programming and computational thinking. Recently artificial intelligence literacy has also started to be addressed in this context teaching concepts related to the big ideas of AI4K12. The teaching of these contents is observed in a progressive way and varies in difficulty from problem decomposition and programming logic to complex concepts such as deep neural networks including GANs. In terms of technological support, visual tools predominate including the block-based programming environment Scratch as well as Google Teachable Machine for teaching neural networks.

As a pedagogical approach for young students, game-based learning seems to be motivating and engaging. In the context of a low SES background, often characterized by the lack of technological tools, unplugged or kinesthetic activities were adopted to support the teaching of computing concepts and to help to engage students in activities. In some instructional units, hands-on activities allowed students to associate with the real world, and working in groups left them more confident.

The main findings of evaluating the quality of the instructional units reveal that students from a low SES background were able to learn basic programming concepts and to solve tasks. The students also were able to gain a general understanding of artificial intelligence. The main difficulties faced by teaching computing to students from this background are related to a lack of infrastructure of public schools and the location where students reside. This lack of access to an appropriate IT infrastructure can cause digital exclusions and social divisions. Mitigation actions include for example the adoption of unplugged activities. Yet, in order to teach certain learning objectives minimal computational resources may be necessary. Therefore, some instructional units were taught at Education and Assistance Centers, computer clubs, or in partnerships with universities or government projects providing the necessary infrastructure. Other difficulties faced by the instructional units are related to the students' lack of experience and knowledge of basic computing, mathematics, literacy, or language competencies. This can cause inequalities and social segregation of students. As a solution, some instructional units sought to teach slowly, with a gentle pace covering also basic computational thinking.

These findings indicate that it is possible to successfully teach computing competencies to students from a low SES background, but that in order to enable these students to achieve the learning goals the instructional units need to be adjusted taking into consideration the specific needs in such a context.

However, considering the small amount of research focusing on computing education targeting students from a low SES background, the results of this review provide only a first overview on the characteristics, limitations and findings in such a context. Yet, pointing out the lack of research targeting these students as result of the review, also indicates the need for more studies in this context in order to provide equal opportunities to students.

Threats to validity. A common risk of systematic literature reviews is the omission of relevant studies. We carefully defined the search string, taking into account key concepts and synonyms. We included not only scientific articles but also scholarly papers, such as theses and dissertations, to reduce the risk of excluding existing instructional units. Furthermore, we searched multiple repositories related to the subject of our review and conducted snowballing in order to minimize the risk of omission. We addressed threats to the study selection and the data extraction by providing a comprehensive definition of the inclusion/exclusion criteria. We established and documented a strict protocol for study selection in which both authors participated, debating until we reached a consensus. Data has been extracted by the first author, revised by the second author and, if necessary, was discussed until consensus was reached.

6. Conclusion

In this article we present the state of the art in teaching computing to young students from a low SES background. Despite the small amount of studies found focusing on students from a low SES background in middle and high school, the results of our analysis provide a first indication that it is possible to teach computing in a motivating way to these students in different contexts, as they were able to learn computational thinking, programming, and artificial intelligence. The results of the review also point out some of the limitations, consequences, and mitigation actions in this context, including a common lack of infrastructure, pre-existing basic competencies, and age heterogeneity. Yet, emphasizing the use of unplugged activities, as well as slowing down the learning pace and adopting age appropriate tools may allow these students to achieve the learning goals. Observing the lack of more research focusing on students from a low SES background, we intend to conduct future case studies in this context to better understand the limitations and the impact of potential solution alternatives.

Acknowledgments

This work was supported by the CNPq (National Council for Scientific and Technological Development), a Brazilian government entity focused on scientific and technological development.

References

Begel, A., García, D.D., Wolfman, S.A. (2004). Kinesthetic learning in the classroom. In *Proc. of the 35th SIGCSE Technical Symposium on Computer Science Education*, Norfolk, VA, USA, 183–184.

Black Girls CODE (2022). `https://wearebgc.org/`

Brackmann, C.P. *et al.* (2019). Development of Computational Thinking in Brazilian Schools with Social and Economic Vulnerability: How to Teach Computer Science Without Machines. *Int. Journal of Innovation Education and Research*, 7(4), 79–96.

Byker, E. (2014). ICT Oriented Toward Nyaya: Community Computing in India's Slums. *Int. Journal of Education and Development using ICT*, 10(2), 19–28.

Conde, M.Á., *et al.* (2017). Promoting Computational Thinking in K-12 students by applying unplugged methods and robotics. In *Proc. of the 5th Int. Conference on Technological Ecosystems for Enhancing Multiculturality*, Cádiz, Spain, 1–6.

Chaffee, J. (2019). *Thinking Critically* (12th ed.). *Cengage Learning*.

Chai, C.S., Lin, P.-Y., Jong, M.S.-Y., Dai, Y., Chiu, T.K.F., Qin, J. (2021). Perceptions of and Behavioral Intentions towards Learning Artificial Intelligence in Primary School Students. *Educational Technology & Society*, 24(3), 89–101.

CSTA (2016). K-12 Computer Science Standards. `http://k12cs.org`

Code2040 (2019). `https://www.code2040.org/`

Codding, D., Mouza, C., Rolón-Dow, R., Pollock, L. (2019). Positionality and Belonging: Analyzing an Informally Situated and Culturally Responsive Computer Science Program. In *Proc. of FabLearn*, New York, NY, USA, 132–135.

Da Cruz-Pinheiro, F., Gresse von Wangenheim, C., Missfeldt Filho, R. (2018). Teaching Software Engineering in K-12 Education: A Systematic Mapping Study. *Informatics in Education*, 17(2), 167–206.

Ekwueme, C., Ekon, E.E. Ezenwa-Nebife, D. (2015). The Impact of Hands-On-Approach on Student Academic Performance in Basic Science and Mathematics. *Higher Education Studies*, 47(5).

Falkner, K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole, E., Liebe, C., Maiorana, F., McGill, M. M., Quille, K. (2019). An international comparison of K-12 computer science education intended and enacted curricula. In *Proc. of the 19th Koli Calling International Conference on Computing Education Research)*, Koli, Finland, 1–10.

Garg, N., De Guzman, R., Jung, E., Migler, T. (2020). The Joys and Challenges of Outreach in CS Education to Low-Income Populations. In *Proc. of the 51st ACM Technical Symposium on Computer Science Education*, Portland, OR, USA, 809–810.

Garneli, V., Giannakos, M.N., Chorianopoulos, K. (2015). Computing education in K-12 schools: A review of the literature. In *Proc. IEEE Global Engineering Education Conference*, Tallinn, Estonia, 543–555.

Girls Who Code (2018). `https://girlswhocode.com/`

Gresse von Wangenheim, C., Alves, N.C., Rauber, M.F., Hauck, J.C.R., Yeter, I.H. (2022). A Proposal for Performance-based Assessment of the Learning of Machine Learning Concepts and Practices in K-12, *Informatics in Education*, 21(3).

Gresse von Wangenheim, C., Hauck, J. C. R., Pacheco, F. S., Bertonceli Bueno, M. F. (2021). Visual Tools for Teaching Machine Learning in K-12: A Ten-Year Systematic Mapping, *Education and Information Technologies*, 26(5), 5733–5778.

Gresse Von Wangenheim, C., Alves, N.C., Rodrigues, P.E., Hauck, J.C. (2017). Teaching Computing in a Multidisciplinary Way in Social Studies Classes in School-A Case Study. *Int. Journal of Computer Science Education in Schools*, 1(2), 3–16.

Gretter, S., Yadav, A., Sands, P., Hambrusch, S. (2019). Equitable Learning Environments in K-12 Computing: Teachers' Views on Barriers to Diversity. *ACM Transactions on Computing Education*, 19(3), Article 24.

Heintz, F., Mannila, L., Färnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. In *Proc. of the IEEE Frontiers in Education Conference*, Erie, PA, USA, 1–9.

Hidden Genius Project (2023). `https://www.hiddengeniusproject.org/`

Holloman, T.K., *et al.* (2021). Underrepresented and Overlooked: Insights from a Systematic Literature Review about Black Graduate Students in Engineering and Computer Science. *Int. Journal of Engineering Education*, 37(2), 497–511.

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.

ISTE (2022). International Society for Technology in Education. `https://www.iste.org/`

ITWORX Education (2023). `https://itworx.education/`

Jones, B.F., Rasmussen, C.M., Moffitt, M.C. (1997). Real-life problem solving: A collaborative approach to interdisciplinary learning. *American Psychological Association*.

Judd, S. (2020). Activities for Building Understanding: How AI4ALL Teaches AI to Diverse High School Students. *Proc. of the 51st ACM Technical Symposium on Computer Science Education*, Portland, OR, USA, 633–634.

Kim, J., Ryu, J. (2017). Analysis of Educational Effects according to the Teaching Methods in Online-Education for Underprivileged Elementary Students. In *Proc. of the 2017 Int. Conference on Education and E-Learning*, Bangkok, Thailand, 47–49.

Kitchenham, B.A., Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Keele University and Durham University Joint Report.

Kolb, D.A. (1984). *Experiential learning: Experience as the source of learning and development*. Prentice-Hall.

Li, C., Said, H., Michael, R., Johnson, M., Meyer, H. (2016). Competency based IT experiences. In *IEEE Frontiers in Education Conference*, Eire, PA, USA, 1–4.

Liebe, C., Camp, T. (2019). An Examination of Abstraction in K-12 Computer Science Education. In *Proc. of the 19th Koli Calling Int. Conference on Computing Education Research*, Koli, Finland, Article 9, 1–9.

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.

Margolis, J., Goode, J., & Chapman, G. (2015). An equity lens for scaling: a critical juncture for exploring computer science. *ACM Inroads*, 6(3), 58–66.

Martins, R.M., Gresse von Wangenheim, C. (2022). Findings on Teaching Machine Learning in High School: A Ten – Year Systematic Literature Review. *Informatics in Education*.

Martins-Pacheco L.H., *et al.* (2020). *Educational Practices in Computational Thinking: Assessment, Pedagogical Aspects, Limits, and Possibilities: A Systematic Mapping Study*. Computer Supported Education. Communications in Computer and Information Science, 1220. Springer, Cham.

OECD PISA. (2019). *PISA 2018 Results (Volume II): Where All Students Can Succeed*, OECD Publishing, Paris.

Ortiz-Lopez, K.D., Holanda, M., Furuta, R., Da Silva, D. (2020). Educational Initiatives to Retain Hispanic/Latinx Students in Computing: A Systematic Literature Mapping. In *Proc. of Research on Equity and Sustained Participation in Engineering, Computing, and Technology*, Portland, OR, USA, 1–2.

Parker, M.C., Guzdial, M. (2015). A critical research synthesis of privilege in computing education. In *Proc. of Research in Equity and Sustained Participation in Engineering, Computing, and Technology*, Charlotte, NC, USA, 1–5.

Papert, S. (1990). *A critique of technocentrism in thinking about the school of the future*. MIT Media Lab Epistemology and Learning Memo, MIT Media Lab, No.2.

Petersen, K., *et al.* (2008). Systematic Mapping Studies in Software Engineering. In *Proc. of the 12th Int. Conference on Evaluation and Assessment in Software Engineering*, Bari, Italy, 68–77.

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education*, 93(3): 223–31.

Refugees Code (2022). https://refugeescode.org/

Resilient Coders (2022). http://www.resilientcoders.org/

Sanusi, I.T. and Oyelere, S.S. (2020). Pedagogies of Machine Learning in K-12 Context. In *Proc. of the IEEE Frontiers in Education Conference*, Uppsala, Sweden, 1–8.

Saskatchewan Education. (1991). *Instructional Approaches: A Framework for Professional Practice*. Saskatchewan Education, Canada.

Simmonds, J., Gutierrez, F. J., Casanova, C., Sotomayor, C., & Hitschfeld, N. (2019). A Teacher Workshop for Introducing Computational Thinking in Rural and Vulnerable Environments. In *Proc. of the 50th ACM Technical Symposium on Computer Science Education*, Minneapolis, MN, USA, 1143–1149.

Teague, D., Roe, P. (2008). Collaborative learning: Towards a solution for novice programmers. In: *Proc. of the 10th Conference on Australasian computing education,* Wollongong, Australia, 147–153.

Tena-Meza, S., Suzara, M., Alvero, A.J. (2021). *Coding with Purpose: Learning AI in Rural California*. arXiv preprint arXiv:2108.13363.

Touretzky, D.S., Gardner-McCune, C., Martin, F., Seehorn, D. (2019). Envisioning AI for K-12: What should every child know about AI? In: *Proc. of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA, 9725–9726.

UNESCO. (2022). *K-12 AI curricula – A Mapping of Government-endorsed AI curricula*. Paris.

UNESCO. (2019). *Exploring STEM Competences for the 21st Century*. Paris.

UNESCO. (2017). *Reducing Global Poverty Through Universal Primary and Secondary Education. Policy Paper 32.* Paris.

UNICEF. (2021). *The State of the Global Education Crisis: A Path to Recovery*. A joint UNESCO, UNICEF and World Bank report. Geneva.

UNRISDSP. (2015). United Nations Research Institute for Social Development. According to Social Protection and Human Rights.

U.S. Department of the Treasury. (2022). Racial inequality in the United States.

Van der Meulen, A., Hermans, F., Aivaloglou, E., Aldewereld, M., Heemskerk, B., Smit, M., Swidan, A., Thepass, C., de Wit, S. (2021). Who participates in computer science education studies? A literature review on K-12 subjects. *PeerJ Computer Science*, 7.

Varma, R. (2009). Attracting Native Americans to computing. *Communications of the ACM*, 52(8), 137–140.

Vergara, K., Herskovic, V., Guerrero, P. (2022). Understanding Gender Bias: Differences in Tech Stereotypes According to the Socio-economic Background of Girls. In: *Proc. of the 2022 ACM Conference on Int. Computing Education Research*, Virtual event and Lugano, Switzerland, 55–56.

Wyatt, J., Feng, J., Ewing, M. (2020). *AP Computer Science Principles and the STEM and Computer Science Pipelines.CollegeBoard*. https://apcentral.collegeboard.org/media/pdf/ap-csp-and-stem-cs-pipelines.pdf?course=ap-computer-science-principles

Westin, T., Manniko-Barbutiu, S., Perera, H., Anuradha, U. (2016). Game-based learning of programming in underprivileged communities of Sri Lanka. In: *Proc. of the European Conference on Games-based Learning*, Paisley, Scotland, 773–780.

Wohlin, C., *et al.* (2012). *Experimentation in Software Engineering*. Springer-Verlag, Berlin.

World Bank. (2022). https://data.worldbank.org/

Zilberman, A., Ice, L. (2021). *Why computer occupations are behind strong STEM employment growth in the 2019–29 decade. Beyond the Numbers: Employment and Unemployment*, 10(1). https://www.bls.gov/opub/btn/volume-10/why-computer-occupations-are-behind-strong-stem-employment-growth.htm

R.M. Martins is a professor for Telecommunications at the Instituto Federal de Santa Catarina (IFSC), São José, Brazil. Currently, he is a Ph.D. student of the Graduate Program in Computer Science (PPGCC) at the Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, and, a research student at the initiative Computing at Schools/INCoD/INE/UFSC. He received an MSc. (2014) in Telecommunications from the National Institute of Telecommunications (INATEL), a specialization (2015) in Telecommunications systems, (2018) in Systems Engineering from ESAB, and B.Eng. in Telecommunications from UNISUL. His main research interests are computing education and Machine Learning.

C. Gresse von Wangenheim is a professor at the Department of Informatics and Statistics (INE) of the Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, where she coordinates the Software Quality Group (GQS) focusing on scientific research, development and transfer of software engineering models, methods and tools and software engineering education. She also coordinates the initiative Computing at Schools, which aims at bringing computing education to schools in Brazil. She received the Dipl.- Inform. and Dr. rer. nat. degrees in Computer Science from the Technical University of Kaiserslautern (Germany), and the Dr. Eng. degree in Production Engineering from the Federal University of Santa Catarina

Appendix A

AQ1. Which instructional units exist or were taught in a low SES context?

Reference	Bibliographic reference	Name of the instructional unit	Brief description
(Anuar *et al.*, 2020)	Anuar, N. H., Mohamad, F. S., & Minoi, J.-L. (2020). Contextualising Computational Thinking: A Case Study in Remote Rural Sarawak Borneo. Int. Journal of Learning, Teaching and Educational Research, 19(8), 98-116.	Computational Thinking in Remote Rural Sarawak Borneo	Activities to introduce computational thinking (CT) to young novice underprivileged indigenous students.
(Araya *et al.*, 2021)	Araya, R., Isoda, M., & van der Molen Moris, J. (2021). Developing Computational Thinking Teaching Strategies to Model Pandemics and Containment Measures. Int. Journal of Environmental Research and Public Health, 18(23), 12520.	Computational Thinking Teaching Strategies to Model Pandemics and Containment Measures	A framework for computational thinking from the Inclusive Mathematics for Sustainability in a Digital Economy Project by Asia-Pacific Economic Cooperation: algorithmic thinking, computational modeling, and machine learning.
(Brackmann *et al.*, 2019)	Brackmann, C. P., Barone, D. A. C., Boucinha, R. M., & Reichert, J. (2019). Development of Computational Thinking in Brazilian Schools with Social and Economic Vulnerability: How to Teach Computer Science Without Machines. Int. Journal of Innovation Education and Research, 7(4), 79-96.	Unplugged Computational Thinking	Unplugged computational thinking classes in Brazilian schools with social and economic vulnerabilities.
(Eguchi, 2021)	Eguchi, A. (2021). AI-Robotics and AI Literacy. In Education in & with Robotics to Foster 21st-Century Skills: Proc. of Edurobotics, Studies in Computational Intelligence, Online, 75-89.	AI-Robotics and AI Literacy	A project focused on developing an affordable open source tool to address the need to promote artificial intelligence (AI) literacy worldwide and especially support the urgent needs of developing countries and underprivileged communities.
(Everson *et al.*, 2022)	Everson, J., Kivuva, F. M., & Ko, A. J. (2022). "A Key to Reducing Inequities in Life, AI, is by Reducing Inequities Everywhere First": Emerging Critical Consciousness in a Co-Constructed Secondary CS Classroom. In *Proc. of the 53rd ACM Technical Symposium on Computer Science Education*, Providence, RI, USA, 209-215.	Creatively Coding a Better Future	A co-constructed high school course to racially, ethnically, socioeconomically, and gender diverse classroom, framing the course as both a creative and critical introduction to computing.
(Harari and Harari, 2017)	Harari, V., Harari, I. (2017). Teaching programming to kids in situation of social vulnerability. In 2017 Twelfth Latin American Conference on Learning Technologies (LACLO), La Plata, Argentina, 1-8.	Teaching Programming to Kids in Situation of Social Vulnerability	Teaching programming for vulnerable children and teens at social risk.

Continued on next page

Table – continued from previous page

Reference	Bibliographic reference	Name of the instructional unit	Brief description
(Levy, 2003) (Levy and Paz, 2005)	Levy, D. (2003). Introducing computer science to educationally disadvantaged high school students - the Israeli experience. In Proc. IEEE Symposium on Human Centric Computing Languages and Environments, Auckland, New Zealand, 269-270.	Introducing Computer Science to Educationally Disadvantaged High School Students	Principles for a course intended to introduce basic computing concepts and ideas to educationally disadvantaged high school students.
	Levy, D., & Paz, T. (2005). The principle of pattern-oriented curriculum and its implementation in a computer science module for high school students. In IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05), Dallas, TX, USA, 321-322.	The Principle of Pattern-Oriented Curriculum and its Implementation in a Computer Science Module for High School Students	An approach for teaching computer science in high school to and by educationally disadvantaged students.
(Miller *et al.*, 2018)	Miller, J., Raghavachary, S., & Goodney, A. (2018). Benefits of Exposing K-12 Students to Computer Science through Summer Camp Programs. In IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 1-5.	CS@SC Summer Camps	A summer camp program to teach K-12 students about computer science, serving as a motivator for underrepresented computer science groups (girls, minorities, low-income families).
(Nogueira *et al.*, 2021)	Nogueira, V. B., Teixeira, D. G., de Lima, I. A. C. N., Costa, D. A. S., & Oliveira, M. A. D. (2022). Towards an inclusive digital literacy: An experimental intervention study in a rural area of Brazil. Education and Information Technologies, 27(3), 2807-2834.	Towards an inclusive digital literacy	A course focused on digital technology, digital culture, and computational thinking, applied as an experimental intervention study in a rural area of Brazil.
(Outlay, 2016)	Outlay, C. N. (2016). Targeting underrepresented minority and low-income girls for computing camps: early results and lessons learned. Journal of Computing Sciences in Colleges, 31(5), 85-94.	Targeting Underrepresented Minority and Low-Income Girls For Computing Camps	A one-day website development camp for minority and low-income middle school and high school girls.
(Prasad *et al.*, 2016)	Prasad, R., Traynor, C., & Keeney, S. S. (2016). Hooking them young: Demystifying computer science and technology among underprivileged high school students. In IEEE Integrated STEM Education Conference (ISEC), Princeton, NJ, USA, 149-155.	Hooking them young: demystifying Computer Science and Technology Among Underprivileged High School Students	Programs to engage refugee, immigrant and underrepresented high school students for a semester-long college campus experience in the field of science, technology, and computing.
(Unnikrishnan *et al.*, 2016)	Unnikrishnan, R., Amrita, N., Muir, A., & Rao, B. (2016). Of Elephants and Nested Loops: How to Introduce Computing to Youth in Rural India. In Proc. of the 15th International Conference on Interaction Design and Children (IDC '16), Manchester, UK, 137-146.	My Elephant Friend	A game concept which serves as a tool for teaching programming and computational thinking to underprivileged children in rural India.

Continued on next page

Table – continued from previous page

Reference	Bibliographic reference	Name of the instructional unit	Brief description
(Yerousis *et al.*, 2015)	Yerousis, G., Aal, K., von Rekowski, T., Randall, D. W., Rohde, M., & Wulf, V. (2015). Computer-Enabled Project Spaces: Connecting with Palestinian Refugees across Camp Boundaries. In Proc. of the 33rd Annual ACM Conference on Human Factors in Computing Systems, Seoul, Republic of Korea, 3749-3758.	Computer-Enabled Project Spaces	Computer club projects applied in marginalized refugee camps.
(Zhang *et al.*, 2022)	Zhang, H., Lee, I., Ali, S., Sargent, J., & Hsu, W. (2022). Integrating Ethics and Career Futures with Technical Learning to Promote AI Literacy for Middle School Students: An Exploratory Study. International Journal of Artificial Intelligence in Education.	Developing AI Literacy (DAILy)	A workshop that integrates ethics and career futures with technical learning to promote AI literacy for middle school students.

Appendix B
AQ2. What are the context characteristics of the instructional units?

Reference	Context of the target audience	Educational stage	Formalism of education	Proficiency level	Competency prerequisites	Device prerequisites	Country
(Anuar *et al.*, 2020)	Underprivileged community living in a remote village in Sarawak Borneo. Indigenous students living in rural areas	Middle School	Informal	Novice	None	None	Malaysia
(Araya *et al.*, 2021)	Students from high-risk schools.	Elementary to High School	Informal	Novice to Intermediate	None (Progressive sequence)	None (instead of using a computer, students can work with a classmate playing its role, to emulate training with a data set).	Chile/ Japan (APEC Project)[1]
(Brackmann *et al.*, 2019)	Students who live in unprivileged areas with schools with social and economic vulnerability	Middle School	Informal	Novice	None	None	Brazil
(Eguchi, 2021)	Underprivileged communities	Elementary to High School	Informal	NI	NI	Robotic kit and computer for Google Teachable Machine.	U.S.

Continued on next page

Table – continued from previous page

Reference	Context of the target audience	Educational stage	Formalism of education	Proficiency level	Competency prerequisites	Device prerequisites	Country
(Everson *et al.*, 2022)	Socioeconomically (and racially, ethnically, gender) diverse classroom. Students who are low-income and/or have no parent or guardian with a bachelor's degree.	High School	Non-formal	NI	None	Computer (all students had school-provided hardware and all coursework was device agnostic)	U.S.
(Harari and Harari, 2017)	Vulnerable children and adolescents at social risk, in neighborhoods on the outskirts (slums) of the city of La Plata.	Middle School	Formal	Novice	None	Computer (in assistance Centers maintained by the Faculty of Informatics)	Argentina
(Levy, 2003) (Levy and Paz, 2005)	Disadvantaged high school students (due to their families socio-economic status, their individual learning and schooling history, behavioral factors, or some diagnosed learning disabilities)	High School	Formal	Novice	None	Computer*	Israel
(Miller *et al.*, 2018)	Underrepresented groups in computing (girls, minorities and from low-income families)	Elementary to High School	Non-formal	Novice	None	Computer*	U.S.
(Nogueira *et al.*, 2021)	Rural area students	Middle School	Non-formal	Novice	None	Computer (provided by the cour-se, donated by a government pro-ject and repaired by the students themselves)	Brazil
(Outlay, 2016)	Underrepresented minority and low-income girls	Middle and High School	Informal	Novice	None	Computer*	U.S.
(Prasad *et al.*, 2016)	Refugee, immigrant and underrepresented high school students	High School	Formal	Novice	None	Computer*	U.S.
(Unnikrishnan *et al.*, 2016)	Tribal and rural area in India	Middle and High School	Non-formal	Novice	None	None	India
(Yerousis *et al.*, 2015)	Palestinian refugee camp´s marginalized population.	Middle School	Informal	Novice	None	Computer (low-cost hardware and free Internet access provided by the local ISP[§] Some mini-laptops were provided by the University Birzeit	Palestine
(Zhang *et al.*, 2022)	Students from low-income families	Middle School	Non-formal	Novice	None	Computer	U.S.

*inferred from the context / †extracted from the material provided / NI not informed or not identified

[§]ISP Internet service provider

References:

[1]APEC (2020). Asia-Pacific Economic Cooperation (APEC) InMside Project. `https://www.criced.tsukuba.ac.jp/math/apec/apec2020/`

Appendix C
AQ3. Which competences are taught in the instructional units?

Reference	Competences according to K-12 Framework CSTA (CSTA, 2016)	Competences according to AI4K12 (Touretzky *et al.*, 2019)	Content taught
(Anuar *et al.*, 2020)	Algorithms and Programming	-	• *Computational Thinking and programming skills*† (Programming Scratch; Tinkering; Variable; Create a quiz using Scratch; Create a project)
(Araya *et al.*, 2021)	Algorithms and Programming; Data and Analysis	Perception; Representation & Reasoning; Learning	• *STEM*† (Mathematical Concepts: Probability) • *Computational Thinking and programming skills*† (Algorithms) • *AI skills*† (Computational modeling: Prediction; Basic algorithms of machine learning: Classification domain; Decision tree; Measures e.g. accuracy, correct and incorrect classification, graphic representations)
(Brackmann *et al.*, 2019)	Algorithms and Programming	-	• *Computational Thinking and programming skills*† (Process of computing concepts)
(Eguchi, 2021)	Algorithms and Programming; Data and Analysis; Societal Impacts	Perception; Representation & Reasoning; Learning; Natural Interaction; Societal Impact	• *AI skills*† (Image classification) • *Computational Thinking and programming skills*† (Programming Scratch) • *Ethics and societal impacts*†(Societal impacts)
(Everson *et al.*, 2022)	Data and Analysis; Impacts of Computing	Perception; Representation & Reasoning; Learning; Natural Interaction; Societal Impact	• *AI skills*† (Explore data; How artificial intelligence works and its sources of bias creating a simple chatbot) • *Ethics and societal impacts*† (Explore and research data privacy and its ethical implication; Develop a website that provides privacy and transparency to the user)
(Harari and Harari, 2017)	Algorithms and Programming; Impacts of Computing	-	• *Computational Thinking and programming skills*† (Concepts of programming: semantic, control structures Programming Scratch, LightBot, Baymax)
(Levy, 2003) (Levy and Paz, 2005)	Algorithms and Programming	-	• *Computational Thinking and programming skills*† (Problem decomposition; Programming Pascal and Hebrew version of Logo[1]; constructing programs using modular procedures; structured programming; global variables and interactive programming using variables; Basic algorithm pattern, conditional)
(Miller *et al.*, 2018)	Algorithms and Programming	-	• *Computational Thinking and programming skills*† (Programming Scratch; Java; Python)

Continued on next page

Table – continued from previous page

Reference	Competences according to K-12 Framework CSTA (CSTA, 2016)	Competences according to AI4K12 (Touretzky *et al.*, 2019)	Content taught
(Nogueira *et al.*, 2021)	Algorithms and Programming; Data and Analysis; Impacts of Computing	NI	• *STEM*§ (Digital technology (math themes) • *Computational Thinking and programming skills*† (Data visualization and introduction to programming logic) • *Digital Culture*† (Digital culture)
(Outlay, 2016)	Algorithms and Programming; Impacts of Computing	-	• *Computational Thinking and programming skills*† (Introductory web development: HTML and CSS coding; Web development processes) • Career opportunities†(potential salaries and jobs)
(Prasad *et al.*, 2016)	Computing Systems; Algorithms and Programming	-	• *In 3th program: Computational Thinking and programming skills*† (Fundamental computing concepts and an introduction to simple programming concepts using Python: concepts such as variables, loops, Boolean logic, and conditionals statements)
(Unnikrishnan *et al.*, 2016)	Algorithms and Programming	-	• *Computational Thinking and programming skills*† (Basic computing concepts; pseudocode)
(Yerousis *et al.*, 2015)	Algorithms and Programming; Societal Impacts	-	• *Computational Thinking and programming skills*† (Basic programming concepts: conditional, looping blocks; Programming Scratch)
(Zhang *et al.*, 2022)	Algorithms and Programming; Data and Analysis	Perception;Representation & Reasoning; Learning; Natural Interaction; Societal Impact	• *AI skills*† (Introduction to AI; Logic Systems: Decision Trees; Machine Learning: Supervised learning, Neural Networks, Unsupervised learning, Generative Adversarial Network, GANs) • *Ethics and societal impacts*† (Ethics) • *Career opportunities*† (Career)

*inferred from the context / †content organized into major themes / NI not informed or not identified / - not applicable

[1]LCSI. (2023). www.microworlds.com

Appendix D
AQ4. What are the instructional characteristics of the units?

Reference	Type of instructional unit	Duration	Instructional mode	Instructional method	Instructional material	Pedagogical approach	Technological tool	Learning assessment
(Anuar *et al.*, 2020)	Activity	Three visits which took place over 12 months	Face-to-face	Lecture; Unplugged	Cards	Game-based learning; Problem-basedlearning; Collaborative learning; Project-based learning	Scratch tool for programming	Observations; Test (comprehension checks); Performance-based assessment based on participants' learning artifacts)
(Araya *et al.*, 2021)	Course	1 month	Face-to-face	Think activity; Hands-on activities (that integrate computational thinking with real-world problems); Discussion	Videos; Paper and pencil†	Collaborative learning	Online platform (ConectaIdeas)	Test
(Brackmann *et al.*, 2019)	Activity	10h class	Face-to-face	Lecture; Hands-on activities*; Unplugged	Paper and pencil†	Active learning*; (Constructionism†)	None	Test (Computational Thinking Test)
(Eguchi, 2021)	Activity	NI	NI	Hands-on activities	Robotic kit	NI	Robotic Kit: CogBots (open source robotics kits developed by CogLabs with Google andUNESCO);Google Teachable Machine; Arduino; Scratch with ThinkBot extension (for Android)	NI

Continued on next page

Table – continued from previous page

Reference	Type of instructional unit	Duration	Instructional mode	Instructional method	Instructional material	Pedagogical approach	Technological tool	Learning assessment
(Everson *et al.*, 2022)	Course	6-week summer term (4 remote courses meeting for 40 min 4 days/week)	Remote	Group work	Dataset (Code.org)	Collaborative learning (Sustaining pedagogy[2])	NI	Observations; Performance-based assessment based on reflective artifacts
(Harari and Harari, 2017)	Course	1h per week for the entire school term	Hybrid (some students online)	Group work; Hands-on activities*	Videos; Exercises (Code.org)	Game-based learning; Interactive-based learning*	Online platform (Code.org, Program.ar[3]); LightBot; Byamax (Walt Disney project); Scratch	Exercises
(Levy, 2003) (Levy and Paz, 2005)	Curricular unit	One school year long, 90h class. 3h week	Face-to-face*	Lecture; Discussion	Software tools	Experience-oriented learning; Project-based learning	Microworlds Project Builder[4]	NI
(Miller *et al.*, 2018)	Summer camp	7h, for one week	Face-to-face	Lecture (by Teacher assistant); Hands-on activities*	Slides	Active learning*	Scratch; Java; Python	NI
(Nogueira *et al.*, 2021)	Course	16h course with 8 (2h classes)	Face-to-face	Lecture	NI	NI	NI	Observations; Test (logic/math assessment test)
(Outlay, 2016)	One-day camp	One day	Face-to-face	Hands-on*; Lecture	Slides	Interactive learning; Active learning*	Online platform (Weeb-ly: a drag-and-drop web publishing tool)	Test (skill assessment); quiz.
(Prasad *et al.*, 2016)	Course	One day a week (2h) during an entire semester	Face-to-face	Hands-on activities; Group work	NI	Game-based learning*	NI	Test

Continued on next page

Table – continued from previous page

Reference	Type of instructional unit	Duration	Instructional mode	Instructional method	Instructional material	Pedagogical approach	Technological tool	Learning assessment
(Unnikrishnan *et al.*, 2016)	Workshop	1 session (16h)	Face-to-face	Hands-on activities*; Unplugged	Cards; Paper and Pencil*	Game-based learning*	None	Observation on completion of activities
(Yerousis *et al.*, 2015)	Workshop	6-8 weeks	Face-to-face	Group work	Training guide	Interactive learning*	Scratch*	Performance-based assessment based on students projects
(Zhang *et al.*, 2022)	Workshop	10h every week for three weeks	Remote (synchronous Zoom meetings and curricular materials on Google Classroom)	Hands-on activities; Interactive activities (kinesthetic); Discussion	Video	Interactive learning*; Game-based learning*	Google Teachable Machine	Tests; Interviews; Observations

[§]content organized into major themes/ *inferred from the context / †extracted from the material provided / NI not informed or not identified / - not applicable

References:

[1]Román-González, M. (2015). Computational Thinking Test: Design Guidelines and Content Validation. In *Proc. of the 7th Annual Int. Conference on Education and New Learning Technologies*, Barcelona, Spain, 2436-2444.

[2]Geneva Gay. (2010). Culturally responsive teaching: Theory, research, and practice. Teachers College Press.

[3]Program.ar (2023). `http://program.ar`

Appendix E

AQ5. Which limitations were observed due to the low SES background context, what were the consequences and how have they been approached?

Reference	Limitation and/or identified needs	Consequences	Solutions/mitigation action
(Anuar *et al.*, 2020)	• Language barrier	• Students had problems using Scratch as it is not available in their mother tongue, the Penan language.	• The Malay language is used as a medium to teach using Scratch.. The code switching between Penan and Malay as they interacted with the facilitators and among themselves seemed to have supported their understanding of the assigned tasks
	• Need for making connections to indigenous knowledge	• Computational thinking skills were not an obvious schema in their living environment	• Mixing game-based learning, collaborative learning, problem-based learning, and project-based learning to present a new "foreign" concept to young novice indigenous children in engaging and understanding computational thinking skills
(Araya *et al.*, 2021)	• Lack of integration of computational thinking and school curriculum	• Students had difficulties to understand real problems	• Teaching computational thinking in integration with mathematics and science curricula
	• Lack of infrastructure in vulnerable school	• Difficulty to teach CT and AI in vulnerable schools	• Instead of using a computer, students can work with a classmate playing its role, emulating training using paper and pencil
(Brackmann *et al.*, 2019)	• Lack of infrastructure in schools with social and economic vulnerability (no digital devices, Internet or even electrical power)	• Difficulty for teachers to teach CT using instructional strategies that require computers	Applying unplugged activities in order to teach computing without the need of a computer or other equipment
(Eguchi, 2021)	• Lack of infrastructure	• Social division (haves and have-nots) in underprivileged communities	• Using an AI educational tool that is accessible in any school in the world
(Everson *et al.*, 2022)	• Computing education classrooms are not diverse, characterized by inequity	• Inequalities in classroom	• Sustaining pedagogy, co-constructing courses with the students, giving students a voice and gathering feedback from the students at the end of activities
	• Teaching online to low-income high school students in pandemic conditions while the students may not have access to computers	• Difficulty for low-income students to learn because they do not have digital devices	• University providing agnostic devices for students

Continued on next page

Table – continued from previous page

Reference	Limitation and/or identified needs	Consequences	Solutions/mitigation action
(Harari and Harari, 2017)	• Lack of computer equipment in public schools in slums	• Students with little or no access to digital devices	• Strengthen links between the university and the slums in order to teach computing. Intervention by the computing faculty allowed students in an Educational and Assistance Center to work with computers and take programming classes
	• Heterogeneous groups with respect to age, ability, interest, and behavior familiarization with programming	• Students with little or no pre-existing basic computing competencies	• Organizing activities from lower to higher complexity
	• Students with no prior programming know-ledge	• Difficulty in learning	• Students simulating to be a robot
(Levy, 2003) (Levy and Paz, 2005)	• Students without access to computers • Students who experience major difficulties in 'regular' school curriculum competencies, such as writing, reading, math	• Difficulties in learning*	• Using MicroWorlds Project Builder, for the first unit especially because of its advantages in the area of simple project building. Students can design a simple programming project using few procedures upon the middle of the learning unit, and then program a 'real' project with graphics and multimedia
(Miller *et al.*, 2018)	• Age heterogeneity of students	• Impact on student interest*	• Using Scratch Jr. and Scratch for younger students, which naturally leads to a focus on developing games • Using Java and Python with older students to develop various projects (including ones other than games) because they have the patience required to learn a text-based language
(Nogueira *et al.*, 2021)	• Some students did not have prior contact with computers • Lack of computing knowledge by public school students in rural areas	• Digital exclusion in multiple forms of social segregation of these students	• Introduction of digital technology in the classroom, improving STEM competencies such as CT and performing logical activities
	• Lack of infrastructure in public schools	• Digital exclusion of these students	• Computers donated by the project "one computer per student" from the Brazil Government
(Outlay, 2016)	• A perception of IT as a male-dominated field, adopted by young girls in the early stages of their educational career and/or ne-gative impressions of the nature of IT work	• Difficulty to recruit and retain girls from and low income families to computing camps	• One-day camp activity, to teach girls about web development jobs and potential salaries

Continued on next page

Table – continued from previous page

Reference	Limitation and/or identified needs	Consequences	Solutions/mitigation action
(Prasad *et al.*, 2016)	• Lack of STEM and computing knowledge of refugee students	• Digital exclusion and inequalities	Programs at Access Academy to create opportunity and motivate students to study STEM and computing skills provided engaged experiential learning
(Unnikrishnan *et al.*, 2016)	• Lack of computing knowledge by rural children	• Children in rural India know less about what a device can do and how to operate it	• Teach technology, using a slow and gentle learning pace using contextualized game-based learning material
(Yerousis *et al.*, 2015)	• Lack of knowledge about Information and Communication Technology by the marginalized population of the Palestinian refugee camp	• Digital divide among students	• Teaching in computer clubs ◦ Teaching Scratch and 3D printing ◦ Forming university partnerships to obtain mini-laptops
(Zhang *et al.*, 2022)	• Gap in access to computing and AI education between students from minority groups and/or low-income families and their white, more affluent peers	• Difficulty in understanding IA concepts • Inequalities	• Design of a curriculum to teach AI including technical concepts and processes • Using everyday context and interactive activities (e.g., hands-on, kinesthetic ones) to explicate AI processes and implications and emphasize the relevance of AI to the students' live • Have students explore AI tools that generate text, images, and videos, and discuss the ethical and societal implications of generative models

*inferred from the context / †extracted from the material provided

Appendix F
AQ6. What was the perceived quality of the instructional units?

Reference	Research Design	Factor(s) evaluated	Data collection method(s)	Sample size	Findings	
					Positive impact	Difficulties
(Anuar *et al.*, 2020)	quasi-experimental	learning; understanding; motivation; playfulness; curiosity; creativity; self-confidence; self-critic;	Observation; Performance-based assessment (participant learning artifacts); Pre-test and post-test	22 (11 girls, 11 boys)	• Indigenous children's learning characteristics were primarily learning-by-making', collaborative, highly motivated, playful, curious, and imaginative while they attempted to learn CT • The experimental group performed marginally better than the control group in the pre-test and demonstrated a substantially better performance in the post-test. • Findings illustrate a direction in which novice indigenous children could learn and be informed about CT through a mix of game-based, collaborative, problem-based, and project-based learning. • Findings also revealed that participants appeared to have gained self-confidence, illustrated creativity on task, and were self-critical throughout their participation in the study. • Engagement levels have visibly increased during the construction of their programming projects. • Techniques, when used with localized examples in the participants' home environment, were deemed culturally adaptable to the learning of CT.	• CT post-test results showed that understanding of computational concepts was poor • Need for supportive peers, guided scaffolding to learn the contents, and a positive learning environment • Switching code from Penan to Malay in verbal interactions, and to English, as the students read prompts using Scratch, might have affected the way the participants understood CT. • An unguided instructional approach is less effective than guided instructional strategies
(Araya *et al.*, 2021)	ad-hoc	learning, understanding	Post-test	90 (48 girls. 42 boys)	• The computational modeling example can be implemented by teachers and correctly understood by students • Students managed to connect it with the real problem, understand some of the limitations of the model, and were able to reflect effectively on how to extend it to overcome some of its limitations	• Short duration of classes (only 90 min) may have hindered the learning

(Brackmann *et al.*, 2019)	quasi-expe-rimental	enthusiasm; motivation; understanding	Pre-test and post-test	63 (28 girls, 35 boys)	• Using unplugged activities is an effective approach as an alternative for students who live in unprivileged areas • CT has been confirmed as a cognitive variable which mainly consists in problem-solving and is not necessarily connected only to computer programming	• Acceptance of the activity by students depends on the educational stage (e.g., "The Elephants" activity uses children's music and did not appeal to older students) • The "Repetition Drawing" activity had to be explained several times and in different ways until the students were able to understand it.* • Very large deficiencies in basic mathematical concepts and literacy hindered the learning process
(Eguchi, 2021)	-	-	-	-	• -	-
(Everson *et al.*, 2022)	ad-hoc	tensions	Performance-based assessment (Reflective artifacts); Pre- and post-survey	14	• Students engaged with tensions discussing topics of equity, justice and marginalization in the AI context.	• Need to establish a classroom climate that allows to have tense conversation in the AI context • Need for a conscientious and poised instruc-tor who motivates the discussion of issues related to AI
(Harari and Harari, 2017)	ad-hoc	motivation; performance	Exercises	30	• The results obtained in literacy of the program-ming field were very satisfactory. • Students were able to work with several tools, educational resources and programming languages • Students were able to accomplish the tasks of programming ''a human-robot'' (100%), the tasks on a cheat sheet (85%), using LightBot, Bymax, and Code.org (100%) • Students were able to accomplish the tasks using Scratch (83%) • A total of 35% of the students progressed through more complex tasks on their own, discovering the concepts before they were approached	• Heterogeneous groups in terms of age, with different problems, level of education, comp-licates learning
(Levy, 2003) (Levy and Paz, 2005)	NI	NI	NI	NI	• Although some of the disadvantaged students do not possess the ability to develop the required algorithms, they were still able to use given patterns and implement them in the programming environment.	NI

Continued on next page

Table – continued from previous page

Reference	Research Design	Factor(s) evaluated	Data collection method(s)	Sample size	Findings: Positive impact	Findings: Difficulties
(Nogueira *et al.*, 2021)	quasi-experimental	self-confidence; performance	Observational testing; Self-assessment	40 (20 intervention group, 20 control group)	• The students had a strong interest in the classes • Although some students did not have prior contact with computers, their development was outstanding • Digital literacy competencies and technology-use behavior increased throughout the semester independent of family income and use of digital devices at home • Students' scores on the logic/math assessment showed significant improvement	• Problems with the school's infrastructure (lack of installations, electrical networks and intermittent electricity; inappropriate internet connection; inadequacies in both classroom environment and storage of equipment) • Completely digitally excluded students • Teachers had little familiarity with the digital environment • School attendance was challenging, since children often missed class
(Outlay, 2016)	ad-hoc	learning	Quiz; Self-assessment	14 (14 girls)	• The quiz results showed that the majority of the girls had retained most of the knowledge taught that day • The IT faculty believed the camp had been successful in teaching web development knowledge to the girls • In general, camp faculty realized that the girls who attended the day camp were interested in computers and computing careers	• Challenges were related to communication, transportation and parental involvement
(Prasad *et al.*, 2016)	ad-hoc	motivation	Observational testing; Self-assessment	11	• Preliminary surveys indicate that most students seemed to have increased their interest in the STEM field as well as their interest in pursuing a college degree • Students were excited while working on the game in the project "Computational Approach to Problem Solving" s	• One of the biggest challenges was attendance. The majority of the students relied on the school bus to come to the course site.

(Unnikrishnan *et al.*, 2016)	ad-hoc	enthusiasm;	Activities	38 (18 workshop 1, 10 workshop 2, 10 workshop 3)	• Childrens can play the game successfully, i.e., create commands to steer the elephant around the board using simple directions, loops, and branches • The fact that it is possible to move from little or no IT experience to actual programming within an afternoon indicates that this approach can be effective for smoothing the early curve of the learning journey to code • It underlined the easily ignored fact that technology per se cannot bridge the digital divide and increase inclusion amongst the rural poor: it requires human relationships, warmth, and discernment to wield the tools	• Some children are more enthusiastic than others, and these were invited to make playful efforts at writing their own application in processing • Working with children can be tiring, especially when ideas become complex and feel less like play. • Classroom orchestration is tough on the facilitator, especially when considering the high level of distractions in a rural school • There are frequent intrusions into the classroom from outside, such as siblings, parents, and animals • Children may be called away on chores • Electricity is likely to come and go as it pleases. • Teachers used to a more sanitized and orderly classroom, may become rattled by the intrusion of large quadrupeds
(Yerousis *et al.*, 2015)	ad-hoc	-	Observation	-	• In the project ”Scratch”, childrens agree to work in groups of two, as they felt more confident of their ability to develop the game as a group • The computer club enables the emergence of social ties among residents of the camp and university students acting as tutors	• In the Camp Jalazone, adjacent to Beit El, an Israeli settlement and active military, the sessions frequently canceled as violent clashes erupted between in-camp youngsters who threw stones and burnt tires, and the Israel Defense Force that responded with the firing of tear gas, rubber-coated bullets. Fortunately, Al-Almari camp does not directly border any Israel settlements, and would be less interrupted.
(Zhang *et al.*, 2022)	quasi-experimental	engaging; enjoying;	Pre and post-test; survey); Interviews; Observation notes; presentation.	25	• Most students developed a general understanding of AI concepts and processes (e.g., supervised learning and logic systems). • Students were able to identify bias, describe ways to mitigate bias in machine learning and start to consider how AI may impact their future lives and careers • Of Students who started the workshop with an incorrect understanding of AI, 37.5% students developed a complex understanding of AI after the workshop.	• Of The students who started the workshop with an incorrect understanding of AI, 25% continued to stick to the incorrect understanding • Complex AI processes such as neural networks may require longer exposures and more time for students to reflect and internalize • Students tended to reason about prediction based on their everyday experience of human intelligence but may have had different interpretations of prediction. They encountered challenges when discerning technologies that make predictions

Continued on next page

Table – continued from previous page

Reference	Research Design	Factor(s) evaluated	Data collection method(s)	Sample size	Findings	
					Positive impact	Difficulties
					• *Students'Technical Knowledge:* Students made the biggest learning gains on four subscales: (1) general concepts of AI, (2) logic systems, (3) general concepts of machine learning, and (4) supervised learning • Students made statistically significant gains on recognizing AI. • Students achieved significant gains on items that assessed understanding of logic systems • Students were able to discern supervised and unsupervised learning by examining whether the technology uses labeled data • Students improved their understanding of supervised learning after the workshop. The Teachable Machine activity engaged students in experiencing how supervised learning takes place by training and testing their models. These activities helped demystify supervised learning processes and also reinforced the idea that AI technology can be biased due to the dataset it was trained on • Students on average improved their understanding of how the GANs work, however, the learning gains were not statistically significant • Students also continued to find the relevance of AI to their lives after the workshop • The career training sessions of the workshop engaged students in exploring their matched future job choices and creating a roadmap to enter those fields. • Overall, the interview results demonstrate that most students incorporated ethical and societal implications into their views of AI technology after the workshop	Students made no gains on instruments that examine student understanding of the processes of neural networks. Possible reasons: intervention too short; online learning format limited student interactions and interpretations; Neural network activities did not include real-world applications of NN to foster student understanding of this concept.

*summarized/ NI - not informed or not identified / - not applicable