A Systematic Literature Review on Teaching Programming to People with Cognitive Disabilities

Elaine Cristina Juvino de Araújo
Federal Institute of Paraíba
Federal University of Campina Grande
Campina Grande, PB, Brazil
elainec@ifpb.edu.br

Wilkerson L. Andrade
Federal University of Campina Grande
Campina Grande, PB, Brazil
wilkerson@computacao.ufcg.edu.br

***Abstract*—This Research Full Paper presents a Systematic Literature Review (SLR) that analyzes scientific studies based on teaching programming to people with cognitive disabilities. Teaching programming is naturally challenging, especially when faced with students with different learning levels, including cognitive impairment. Due to the lack of a panorama that contemplates the association between teaching programming and people with cognitive disabilities, we present this SLR to identify which initiatives are available to teach people with cognitive disabilities to programming. This SLR followed the protocol model of a systematic review adapted to cover specific requirements for planning, executing, and presenting results. There are some initiatives to include students with some cognitive impairment found in the literature. However, there are still gaps to explore. The main question answered by this research is to identify what are the efforts and contributions to teaching people with cognitive disabilities to program, looking for related papers in the Springerlink, ACM, and IEEE databases.**

***Index Terms*—teaching programming, cognitive disabilities, systematic review**

I. Introduction

Knowledge in Computer Science (CS) has become increasingly necessary, and most individuals who are trained in this area have access to a reasonable variety of job opportunities [5]. For example, in the United States, there are several efforts to include CS in both elementary and high school curricula to allow students to get in touch with CS concepts at an earlier age [10]. These initiatives are essential to promote the inclusion of students with disabilities, allowing equal access to academic and industry careers [10].

In practice, CS earlier education includes CS concepts, Computational Thinking (CT), programming, and robotics [10]. CT involves solving problems through fundamental concepts of CS [29]. There is evidence that students with cognitive disabilities can learn CT with proper support [17]. Programming crosses all areas of knowledge and can help develop several cognitive skills in the student, to mention: creativity, logical reasoning, critical problem-solving, reflection, interpretation, concentration, among others. Skills developed from the study of programming can cause a significant impact in shaping critical and conscious individuals who can contribute significantly to transforming society [23].

Since elementary school, many efforts have been made to establish programming earlier and earlier in school curricula. Although some initiatives might have been successful in popularizing programming with younger audiences, most forget to consider students who have learning disabilities or attention deficit disorders. Making learning accessible to all students is a significant challenge because teachers need to be appropriately trained to deal with students who learn differently.These students can obtain knowledge by meeting their specific learning needs. Schools seek to improve the educational process by searching for new methodologies, training teachers, and obtaining new technologies [6], [7]. In Brazil, for example, there is a growing increase in enrollments of people with disabilities [9], which leads us to question how to teach programming to students with different ways of learning.

There are several types of cognitive impairments cataloged, such as Autism Spectrum Disorder (ASD) and Attention Deficit Hyperactivity Disorder (ADHD), to name a few. ASD characterizes a deficit in communication and social interaction, behavior patterns, interest, and restricted and repetitive activities. ADHD characterizes a neurodevelopmental disorder that can present harmful levels of inattention, disorganization, and hyperactivity-impulsivity. Inattention and disorganization can involve inability to stay on task, difficulty in paying attention,, and missing or misplacing objects at levels that are inconsistent with age or level of development. Hyperactivity-impulsivity can come as excessive activity, restlessness, inability to remain seated, interference and activities of others, and inability to wait - characteristics that may come in excess for the age or the level of development [24].

Game development can be especially beneficial for people with ASD, which is one type of cognitive impairment. Assistive technologies can provide different ways of accessing information, thus guaranteeing a better environment for students with disabilities, including those with learning difficulties [1], [2].

This work aims to determine what and how researchers work in teaching programming for people with cognitive disabilities. To this end, we conducted a Systematic Literature Review (RSL) to answer the question: “what are the existing efforts and contributions to teach people with cognitive disabilities how to code”?

This paper is divided as follows: In Section II, we present

the related work. In Section III, we describe the methodology following the SLR process. We present the results answering the research questions in Section IV. We discuss the results observed during the SLR in Section V. We present the threats to validity in Section VI. Finally, we present the conclusions and pointers for further work in Section VII.

II. Related Work

There are reviews aimed at teaching programming, but no reviews focused on students with cognitive disabilities. Among the related works, we sought to provide an overview of introductory programming, organizing its results according to the approach: in students, teaching, the curriculum, and assessment. The student approach mentioned a subgroup of underrepresented students, mainly women and non-disabled students. [31].

Another review showed published studies on robotics in teaching programming but focused only on the visually impaired. This work signaled the teaching methodologies and difficulties faced in this scope, but which would not apply to students with cognitive disabilities [32].

Some research was developed to include people with learning disabilities in programming or elementary computing. One paper proposes integrating social skills with the teaching of programming for students with ASD based on the idea that programming requires collaborative efforts. The focus of the course was programming, but there was an interest in taking advantage of the moment so that students could practice applied social skills, such as working in groups, receiving and giving feedback on activities, and knowing how and when to ask for help. The work suggests benefits in this practice but needs more research [25].

Studies believe that block programming can be more interesting for people with learning difficulties, applying in classes with children in elementary school [26], [30]. However, the papers suggest that more research is needed to generate greater student engagement and teacher support.

A study made the cross between the cognitive style of people with autism and the necessary cognitive skills in Software Engineering to generate guidelines for inclusive education. Unfortunately, the generated guidelines are purely speculative, requiring further studies to prove whether they are viable [28].

One paper reports an exploratory study that sought to expand the opportunities for K-12 students who learn differently in teaching computing principles. This research resulted in guidelines for adapting the materials used, being evaluated by feedback from students and teachers involved [6].

III. Methodology

An SLR [3] was carried out, seeking to investigate how the inclusion of programming for people with cognitive disabilities is being thought about and carried out. A few steps are required to perform the review: planning the review, conducting the review, and reporting the review.

In planning the review, we need to identify the need for it, define the research questions, and use the protocol to carry it out. The protocol begins with selecting primary studies, extraction, and synthesis of data in conducting the review. We define the mechanisms for generation and disclosing the information found in the review of the report.

A. Review Protocol

The RSL protocol defines that a critical question to be answered must be created, and the critical question of this work is to identify "what are the existing efforts and contributions to teach people with cognitive disabilities how to code"? From this question, we derived three research questions (RQ) for identifying the state of the art in this context:

- RQ1: How did teachers approach the content for students with cognitive disabilities?
  The purpose of RQ1 is to verify which approaches at the pedagogical level are being adopted in teaching programming to people with cognitive disabilities.
- RQ2: What technologies are used to support teaching?
  The objective of RQ2 is to identify which technologies support these approaches within this context.
- RQ3: How are students evaluated?
  The RQ3 aims to see what is being used to assess the learning of students with cognitive disabilities in teaching programming.

B. Conducted Search

Initially, we carry out tests to identify the best terms and bases to be used, which could be within the context of students with some cognitive impairment and programming teaching, arriving at the combination of terms presented in Table I.

TABLE I
Synonyms or Related Words

Terms	**Synonyms or Related Words for Keywords**
Programming-related	"computer science", "cs","programming"
Teaching-related	"teaching", "learning", "students"
Terms related to cognitive impairment	"autism", "cognitive", "intellectual", "learning", "neurodiversity"
Accessibility-related	"disabilities", "accessibility"

TABLE II
Summary of the studies returned in each digital library

Digital Library	**Search Results**
ACM	63
IEEE	280
Springer	369
Total	**712**

The search string chosen was:

("computer science" OR "cs" OR "programming") AND ("teaching" OR "learning" OR "students") AND ("autism" OR "cognitive" OR "intellectual" OR "learning" OR "neurodiversity") AND ("disabilities" OR "accessibility")

Table II shows the number of studies returned in each digital library. We found seven hundred and twelve screening papers using inclusion (CI) and exclusion (EC) criteria. The databases used were ACM, IEEE, and Springer, and we adapted the search string for each of these databases. The inclusion criteria defined for the selection were:

- (IC1) - Papers in English;
- (IC2) - Papers published in the last 10 years.
- (IC3) - Papers must contain the keywords used in the key; search, in the title and / or in the abstract and / or in the keywords of the selected paper;
- (IC4) - Complete papers that present methodologies, suggestions or practices for teaching programming to people with some type of cognitive impairment.

And the exclusion:

- (EC1) - Repeat papers;
- (EC2) - Review papers.

C. Screening of papers for inclusion and exclusion

The screening process is presented in Table III, where we applied the inclusion and exclusion criteria. We read the titles, keywords, and abstracts to select the papers, choosing only those that met the search scope.

The first step was to visualize the number of papers found in the search bases. Even applying a search string, databases often return papers that are not related to the search criteria. Therefore, after applying the inclusion criteria IC1 and IC2, some papers not written in English or not published in the last ten years were discarded.

The second step was to verify which papers had the terms defined in the search string in their title, abstract, or keywords, seeking to remove a good part of the returned papers that were not in the research context.

The third step was to read the titles of the papers, removing those that were not related to the review.

The fourth step was to read the abstracts of the papers, seeking to refine the list of selected ones further.

Finally, the fifth step of the screening process was the complete reading of the twelve papers resulting from the previous steps, of which only eight were considered relevant to the research.

D. Snowballing Search

Complementing the automatic review process in the databases, we use snowballing, a search process that involves looking for relevant studies in an paper's reference list or citations, seeking more relevant papers that the automatic search may have excluded. It is challenging to formulate a good search string, as there is often no standard in the terminology used, and the more comprehensive the search terms are, the greater the chances of finding irrelevant papers [12]. Combining search strategies can be an excellent alternative to obtain better results, using snowballing as a primary or secondary search strategy [22].

According to the strategy presented by [12], initially you must select the set of papers to be used as a basis. The basis here was the eight papers selected at the end of the screening done in the papers obtained in the automated search.

The snowballing methodology was applied backward from the initial set, identifying the studies that the papers in the set refer to within the investigated theme. At the end of the process, we selected six papers, which were later read and classified, totaling 14 papers for analysis in the review, as can be seen in Table IV.

TABLE III
SCREENING OF PAPERS

Steps	Papers
Step 1: Identify and organize the papers found in the search bases by the inclusion criteria IC1 and IC2;	712 papers
Step 2: Refine by inclusion criterion CI3;	241 papers
Step 3: Review of titles removing papers that do not adhere to the theme and / or fit the exclusion criteria;	40 papers
Step 4: Review of abstracts, removing papers that do not adhere to the theme;	12 papers
Step 5: Review of the complete papers removing the papers that do not adhere to the theme;	8 papers

TABLE IV
FINAL SELECTION OF PAPERS

Search	Papers
Digital Libraries;	8 papers
Snowballing Search	6 papers
Total:	14 papers

IV. RESULTS

This section presents the results of this SLR. The survey took place between August 2020 and January 2021. Here we present the results by year, by country, by types, and vehicles published. Finally, the following subsections detail the answer to each of the research questions.

A. General Results

While screening the papers included in this research, we found different approaches and technologies used to teaching programming to students with disabilities. Thus, we sought to answer the research questions and the answers are presented in this chapter. The results of the SLR shows that there are efforts to include people with cognitive disabilities in programming teaching, but there is still much to be explored.

1) Publication year: We distribute publications by year as shown in 1. We see that the peak of publications occurs in 2016 and 2018, corresponding to 21.42 % of publications. The years 2015 and 2020 account for 14.29 % each and 2011, 2013, 2017, and 2019 for 7.14 % each. Figure 1 shows that the scientific community is interested in the subject, and most years between 2010 and 2020, they published papers in this research context.

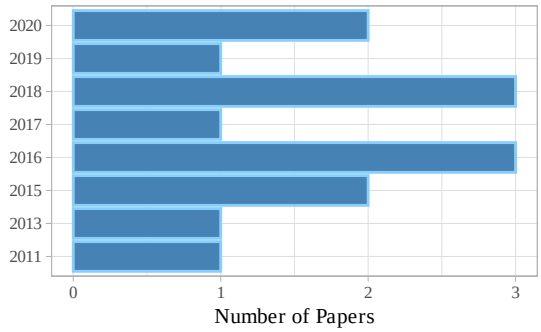

Fig. 1. Frequency by year

2) Countries distribution: Figure 2 shows that more than half of the papers published in the area come from the United States (US), totaling nine papers. Among the other papers there were two from the United Kingdom, one from Portugal / Brazil, one from Greece, one from Brazil, and one from Spain. The fact is that in the US there are more consistent initiatives to offer computer education and programming at an earlier age, which ends up reflecting on the most significant number of researches.

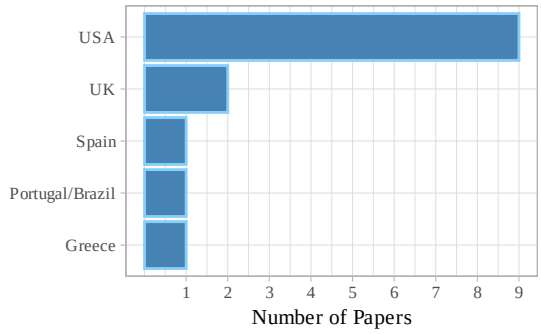

Fig. 2. Frequency by country

3) Publication vehicles: Most of the studies were published at conferences (six), as can be seen in Figure 3. The other papers appear in journals (four), symposiums (three), and workshops (one).

4) Types of studies: We separated the studies by type of research performed (Figure 4). Most were case studies (nine), followed by experience reports (four) and an exploratory study (one).

B. RQ1: How did teachers approach the content for students with cognitive disabilities?

It is understood that traditional teaching methods end up excluding students with cognitive disabilities or who learn differently. Therefore, the studies found reinforce this idea, experimenting with different approaches to help these students

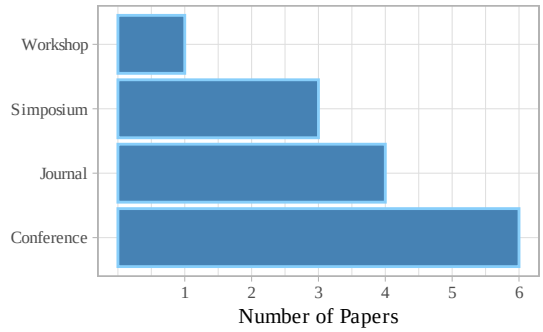

Fig. 3. Frequency by vehicle

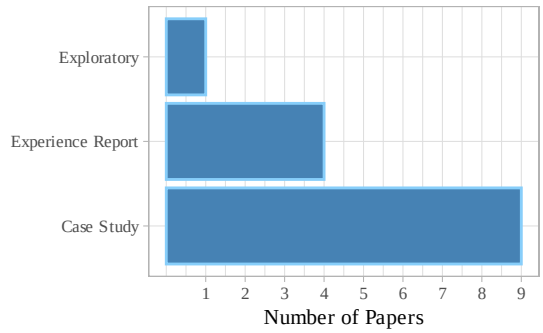

Fig. 4. Frequency by type

learn programming, as shown in Table V. Block programming and robotics were the most cited approaches in the studies.

Teachers who teach computing and CT should pursue integrated and straightforward approaches for instruction and tailor them to the content and learning objectives. There are numerous resources for teachers related to teaching programming and computing based on block programming, which makes it a good strategy for the initial concepts [5], [17], [19]–[21].

Robotics is very popular, mainly in kindergarten, being a strategy that uses fun to represent the most varied programming concepts. Robotics can be a factor of inclusion and equality, in addition to providing students with new knowledge through its attractive challenges [11], [13], [14], [16], [19]. On the other hand, one of the papers points out that students have more engagement when they use pre-built robots, but when they have to build the robots, they become less involved and more frustrated with more difficulty focusing on activities [13].

Other approaches have also proved popular, such as Universal Design for Learning (UDL) and Unplugged Programming (UP). Most initiatives are mainly focused on involving students, making them want to learn, and flexibly providing content, so that each student can choose the best way to receive them. UDL seeks to adapt traditional content and activities to reach different students with their different levels

of learning. There was no standardization among teachers for using this approach, and the main emphasis was on applying the principles of engagement and multiple forms of content presentation. We also realized that using this approach in computing was not significantly different from applying it in other areas. The key to using the UDL would be the teacher's view of their learning goals, noting the students' strengths, and seeking to reduce barriers to learning. [7], [10], [11], [20].

Teaching programming with non-electronic alternatives has become an exciting alternative. Physical toolkits can make abstract computing concepts easier to understand. Papers suggest that this approach is beneficial for teaching people with disabilities, but few studies still point in this direction. The use of physical tools associated with collaborative learning proved promising, with good interaction and enthusiasm among students, helping to keep them focused on activities for longer periods. On the other hand, it is not yet possible to guarantee that the activities with these devices are, in fact, inclusive for all students [11], [18]–[21].

The use of games was mentioned in fewer papers, focusing on the relationship between play and learning. They validate the idea that games leverage skills which could help individuals in the future labor market, such as teamwork and problem solving. This idea would also be interesting for students with ASD, as it stimulates some social skills. Robotics concepts are often used in association with games, but students with a higher functioning spectrum were more interested in game design and real-world storytelling with the characters they created [13], [15].

A single work selected in this review presented accessibility guidelines focused on students with dyslexia, seeking to help these students to overcome their difficulties and stand out in programming classes. A set of recommendations associated with the skills needed for programming was proposed, showing what difficulties the dyslexic programmer may face when trying to gain these skills and the reasonable adjustments that could also be applied to students who learn differently [27].

TABLE V
APPROACH THE CONTENT

Approach	Papers
Block Programming	[5], [7], [13], [17], [19]–[21]
Robotics	[11], [13], [14], [16], [19]
Unplugged Programming	[11], [18]–[21]
UDL	[7], [10], [11], [20]
Gamming	[13], [15]
Own Guidelines	[27]

C. RQ2: What technologies are used to support teaching?

The use of specific technology aimed at teaching how to code is a popular approach in many studies, since there are countless options to be explored to find the best way to serve students. However, there is no standard for teaching people with disabilities, as we can see in Table VI. Two studies used most technologies, except Scratch, Code.org, and Dash Robot, covered by 6, 4, and 3 studies each, respectively.

Another interesting fact is that almost all technologies referenced by papers use blocking or robotics, with Magic Cubes Toolkit being the only technology used for UP.

In addition to serving as an accessible entry path for learning how to code, more playful technologies can promote coding as a fun and stimulating activity.

TABLE VI
TECHNOLOGIES

Technologies	Papers
Lego Mindstorms	[13], [14]
Unit 3D	[15]
Scratch	[5], [11], [13], [14], [17], [20]
Code.org	[7], [11], [17], [20]
Khan Academy	[20]
Hello Ruby	[11]
Dash Robot	[11], [19], [21]
Blockly	[19], [21]
Alice	[13], [20]
Phogo	[11]
Magic Cubes Toolkit	[18]

D. RQ3: How are students evaluated?

Assessing student learning is often challenging due to the individual nature of each student's background. Good planning is essential for the teacher to identify whether the student learned that content. This review made it evident that the evaluation of students with disabilities is still something that needs to be studied and discussed, especially in the context of learning how to code.

The papers found in this review use mainly extracurricular courses, where the focus of the evaluation is the courses and not as a way to measure student's performance formally. Also, the evaluations are based on the student's perceptions, opinions, and personal data obtained through observations and interviews as can be seen in Table VII. Only two studies presented a formal strategy to measure student performance through a frequency checklist, analyzing the number of mistakes and successes of each study participant [19], [21]. There is a strong suggestion that formal evaluation of programming performance is an open field of study.

TABLE VII
EVALUATION

Evaluation	Papers
Observation	[5], [10], [11], [13]–[18]
Interview	[11], [17]
Focus Group	[11], [13], [14], [16]
Student or Teacher Feedback	[13], [15]
Class Records	[7], [10], [11]
Photo or Video Analysis	[14], [18]
Checklist	[19], [21]

V. DISCUSSION

Through this SLR, it was possible to notice several types of research involving the teaching of programming and people with cognitive disabilities. People who learn differently was

also a term used, seeking to emphasize that anyone can learn, as long as they have the correct support [7].

A. Can anyone learn programming?

One of the significant challenges in teaching programming to people with cognitive disabilities is how these students receive and absorb the content. The technical language seems to be a barrier, which leads us to question the best strategies to adapt the language to facilitate understanding [7].

Studies have shown that close support and mentoring for the individual student can facilitate the involvement of students with cognitive disabilities in the CT tasks assigned to their class [17].

The results found in this research show strong evidence that students with cognitive disabilities can develop the skills needed to learn programming provided they have adequate support, which implies training for teachers, adapted materials, and, on some occasions, collaborative work.

In line with content adaptations, the UDL methodology appears as a popular approach, which seeks to meet the needs of students in a different way by making content available in multiple formats, so that each student chooses the best way to obtain knowledge.

B. Programming as a profession

An important issue raised by this research is that the vast majority of papers in the area focus on children, as shown in Figure 5, where 12 of the 14 selected papers cover up to k-12. Programming is then presented playfully as another interesting knowledge for anyone to learn in these cases.

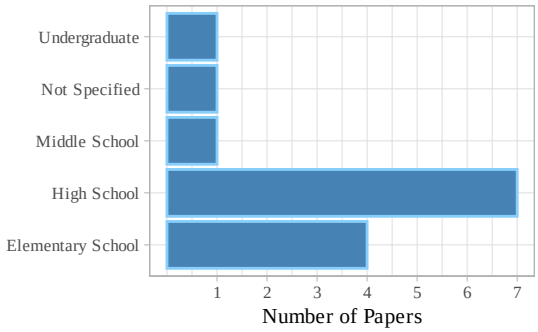

Fig. 5. Frequency by level

We cannot gauge how many of these students will pursue careers in computing. Hence, a possible area to be explored is the teaching of programming for people with disabilities in technical and higher education courses. Students with specific needs are often entering these courses without having prior knowledge on the field.

Teaching programming to students who have cognitive disabilities is already a reality with inclusive education policies. Teachers need to adapt their way of exposing knowledge and evaluating learning for each student, with and without disabilities, as diversity in learning is already common in the classroom. Teachers face a significant challenge at any level of education: welcoming students who learn in different ways, looking for different methodologies to teach them according to their specific needs. In the context of higher education, however, the teacher's responsibility becomes significantly bigger, as these students are preparing for a professional career most of the time.

C. Types of cognitive impairment

The focus of this review was on papers involving research on teaching programming to people with some cognitive impairment. In the scope of cognitive disabilities, we can have several types, and the diversity of terms used in the papers reflects this, as we can see in Figure 6. Most of the terms were used in only one work, except for "Learning Difficulties" used in three papers and "Students with Disabilities" used in two papers. Most papers generically deal with cognitive impairment, leaving the question of whether the strategies used can be generalized or not since each disability has its characteristics and needs.

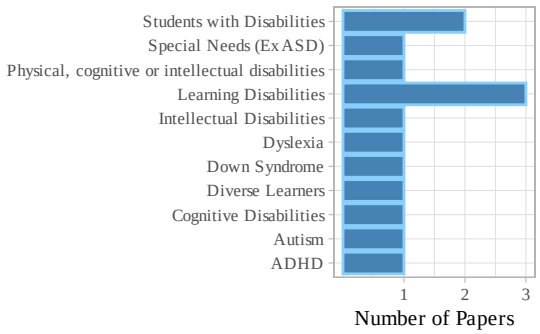

Fig. 6. Frequency by disability

VI. Threats to Validity

Even having followed the SLR protocol, we can identify threats to the research's validity. We designed the search key to cover the maximum number of terms related to the topic, but the lack of standard terminology may result in a smaller set of potential papers in the results. Each search base has specific features that may influence the results obtained and may not have returned all relevant studies for this work. The screening work is manual and human factors can influence data extraction and analysis when reading papers.

VII. Conclusion and Future Works

Learning how to code has been considered an important skill with several initiatives for its popularization, starting as early as in elementary school or even before that. In classes with students of varying profiles, including people with cognitive disabilities, it can be difficult for teachers to teach the many concepts of Computer Science, often the abstract ones. With this SLR, it was possible to understand that several initiatives

seek the inclusion of people with cognitive disabilities in programming classes. We answered three research questions relating to programming and people with cognitive disabilities. Initially, we saw that the most used approaches were Programming in Blocks, Unplugged Programming, and Robotics, possibly as a way to bring a more playful approach within the learning context. Then it was identified that there is a variety of learning technologies explored, Scratch being the most mentioned in the papers, followed by the Code.org platform. It was also observed that there is no well-established learning assessment process. the observation and opinion of students and teachers through interviews is still the most adopted strategy to validate the proposals.

As future research topics, we suggest studies dedicated to helping students with cognitive disabilities in the initial programming disciplines of a technical or higher education course in the field of computing. These students can enter courses without prior knowledge in technology, with a deficit in logic and mathematics, needing additional help to assimilate content and grow to pursue a career in this area. Another critical open question is how to assess these students to see what skills they acquired during the course, a practice that becomes even more critical when these students are enrolled in CS courses since the programming disciplines of later stages require the knowledge gained in the initial classes. The introductory topics presented in those classes are crucial for students to progress in their studies and build a successful career.

REFERENCES

[1] R. Munoz, R. Villarroel, T. S. Barcelos, F. Riquelme, Á. Quezada and P. Bustos-Valenzuela, "Developing Computational Thinking Skills in Adolescents With Autism Spectrum Disorder Through Digital Game Programming," in IEEE Access, vol. 6, pp. 63880-63889, 2018, doi: 10.1109/ACCESS.2018.2877417.

[2] El Kah, A., Lakhouaja, A. Developing effective educative games for Arabic children primarily dyslexics. Educ Inf Technol 23, 2911–2930 (2018), doi: https://doi-org.ez291.periodicos.capes.gov.br/10.1007/s10639-018-9750-2

[3] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. In Technical report, Ver. 2.3 EBSE Technical Report. EBSE. sn, 2007.

[4] T. Zaki et al., "Towards developing a learning tool for children with autism," 2017 6th International Conference on Informatics, Electronics and Vision & 2017 7th International Symposium in Computational Medical and Health Technology (ICIEV-ISCMHT), Himeji, 2017, pp. 1-6, doi: 10.1109/ICIEV.2017.8338597.

[5] Varsha Koushik and Shaun K. Kane. 2019. "It Broadens My Mind": Empowering People with Cognitive Disabilities through Computing Education. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 514, 1–12. DOI:https://doi.org/10.1145/3290605.3300744

[6] S. Wille, J. Century and M. Pike, "Computer Science Principles (CSP) and students with learning differences: Expanding opportunities for a hidden underrepresented group," 2016 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), Atlanta, GA, 2016, pp. 1-8, doi: 10.1109/RESPECT.2016.7836160.

[7] S. Wille, J. Century and M. Pike, "Exploratory Research to Expand Opportunities in Computer Science for Students with Learning Differences", in Computing in Science & Engineering, vol. 19, no. 3, pp. 40-50, May-June 2017, doi: 10.1109/MCSE.2017.43.

[8] P. da Silva Neves Lima, L. das Almas Silva, J. L. dos Santos Oliveira, A. A. Franco Brandão and L. de Oliveira Brandão, "Computational games in STEM courses: a systematic review of the literature", 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1-8, doi: 10.1109/FIE44824.2020.9274071.

[9] Brasil. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep). Censo da Educação Básica 2019: resumo Técnico. Brasília, 2020.

[10] Meg J. Ray, Maya Israel, Chung eun Lee, and Virginie Do. 2018. A Cross-Case Analysis of Instructional Strategies to Support Participation of K-8 Students with Disabilities in CS for All. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). Association for Computing Machinery, New York, NY, USA, 900–905. DOI:https://doi-org.ez291.periodicos.capes.gov.br/10.1145/3159450.3159482

[11] Maya Israel, Gakyung Jeong, Meg Ray, and Todd Lash. 2020. Teaching Elementary Computer Science through Universal Design for Learning. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery, New York, NY, USA, 1220–1226. DOI:https://doi-org.ez291.periodicos.capes.gov.br/10.1145/3328778.3366823

[12] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14). Association for Computing Machinery, New York, NY, USA, Article 38, 1–10. DOI:https://doi.org/10.1145/2601248.2601268

[13] R. Dorsey and A. M. Howard, "Examining the Effects of Technology-Based Learning on Children with Autism: A Case Study", 2011 IEEE 11th International Conference on Advanced Learning Technologies, Athens, GA, USA, 2011, pp. 260-261, doi: 10.1109/ICALT.2011.81.

[14] C. Conchinha, P. Osório and J. C. de Freitas, "Playful learning: Educational robotics applied to students with learning disabilities", 2015 International Symposium on Computers in Education (SIIE), Setubal, Portugal, 2015, pp. 167-171, doi: 10.1109/SIIE.2015.7451669.

[15] C. Galeos, K. Karpouzis and G. Tsatiris, "Developing an educational programming game for children with ADHD," 2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA, Zakynthos, Greece, 2020, pp. 1-6, doi: 10.1109/SMAP49528.2020.9248458.

[16] Carlos Gonzalez-Sacristan, Carlos Garcia-Saura, and Pablo Molins-Ruano. 2016. Phogo: a low cost, engaging and modern proposal to learn how to program. In Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM '16). Association for Computing Machinery, New York, NY, USA, 67–71. DOI:https://doi.org/10.1145/3012430.3012498

[17] Snodgrass, Melinda. R., Israel, Maya., & Reese, George. C. (2016). Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis.Computers & Education, 100, 1-17.

[18] Zuzanna Lechelt, Yvonne Rogers, Nicola Yuill, Lena Nagl, Grazia Ragone, and Nicolai Marquardt. 2018. Inclusive Computing in Special Needs Classrooms:Designing for All. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, 517

[19] Taylor, Matthew S. "Computer Programming With Pre-K Through First-Grade Students With Intellectual Disabilities". The Journal of Special Education, vol. 52, no. 2, Aug. 2018, pp. 78–88, doi:10.1177/0022466918761120.

[20] Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S., & Tapia, T. (2015). Empowering K–12 Students With Disabilities to Learn Computational Thinking and Computer Programming. TEACHING Exceptional Children, 48(1), 45–53. https://doi.org/10.1177/0040059915594790

[21] M. S Taylor, E. Vasquez, and C. Donehower. 2017. Computer Programming with Early Elementary Students with Down Syndrome.Journal of Special Education Technology32, 3 (2017), 149–159. https://doi.org/10.1177/0162643417704439

[22] Kai Petersen, Sairam Vakkalanka, Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, Volume 64, 2015, Pages 1-18, ISSN 0950-5849. https://doi.org/10.1016/j.infsof.2015.03.007.

[23] Cristina Santos e Denilson Silva e Maria Silveira e Giana Ferreira. 2017. Desafio de Programação para Meninas do Ensino Médio: Um Relato de Experiência.Anais do Workshop de Informática na Escola 23, 1 (2017), 137. https://doi.org/10.5753/cbie.wie.2017.137

[24] American Psychiatric Association. 2013.Diagnostic and Statistical Manual ofMental Disorders. American Psychiatric Association. https://doi.org/10.1176/appi.books.9780890425596

[25] K. Eiselt and P. Carter, “Integrating Social Skills Practice with Computer Programming for Students on the Autism Spectrum”, 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 2018, pp. 1-5, doi: 10.1109/FIE.2018.8659252.

[26] Maya Israel, Jamie N. Pearson, Tanya Tapia, Quentin M. Wherfel, George Reese.Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis,Computers & Education,Volume 82,2015,Pages 263-279,ISSN 0360-1315,https://doi.org/10.1016/j.compedu.2014.11.022. (https://www.sciencedirect.com/science/article/pii/S0360131514002759)

[27] Silke Stienen-Durand, Jennifer George. Supporting Dyslexia in the Programming Classroom. Procedia Computer Science. Volume 27. 2014. Pages 419-430. ISSN 1877-0509. https://doi.org/10.1016/j.procs.2014.02.046.

[28] Sylvia Stuurman, Harrie J.M. Passier, Frédérique Geven, and Erik Barendsen. 2019. Autism: Implications for Inclusive Education with respect to Software Engineering. In Proceedings of the 8th Computer Science Education Research Conference (CSERC ’19). Association for Computing Machinery, New York, NY, USA, 15–25. DOI:https://doi.org/10.1145/3375258.3375261

[29] Jeannette M. Wing. 2006. Computational thinking. Commun. ACM 49, 3 (March 2006), 33–35. DOI:https://doi.org/10.1145/1118178.1118215

[30] R. Thompson,“Teaching coding to learning-disabled children with Kokopelli’s World,” 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Cambridge, UK, 2016, pp. 258-259, doi: 10.1109/VLHCC.2016.7739705.

[31] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. Introductory programming: a systematic literature review. In Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018 Companion). Association for Computing Machinery, New York, NY, USA, 55–106. DOI:https://doi-org.ez291.periodicos.capes.gov.br/10.1145/3293881.3295779

[32] Damasio, J., Jurak, D., Bittencourt, R., Campos, M., & Amory, A. (2019). Programming teaching with robotic support for people who are visually impaired: a systematic review. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), 30(1), 1231. doi:http://dx.doi.org/10.5753/cbie.sbie.2019.1231