TYPE Original Research
PUBLISHED 29 November 2024
DOI 10.3389/feduc.2024.1484921

Check for updates

OPEN ACCESS

EDITED BY
David Lansing Cameron,
University of Agder, Norway

REVIEWED BY
Radoslava Stankova Kraleva,
South-West University "Neofit Rilski", Bulgaria
Liudmyla Gryzun,
Simon Kuznets Kharkiv National University of Economics, Ukraine

*CORRESPONDENCE
Francesco Domenico Di Blasi
✉ fdiblasi@oasi.en.it

RECEIVED 22 August 2024
ACCEPTED 14 November 2024
PUBLISHED 29 November 2024

CITATION
Stimoli MA, Di Blasi FD, Maccarrone S, Costanzo AA, Occhipinti P and Buono S (2024) Teaching basic computer programming to young adults with intellectual disability.
Front. Educ. 9:1484921.
doi: 10.3389/feduc.2024.1484921

COPYRIGHT
© 2024 Stimoli, Di Blasi, Maccarrone, Costanzo, Occhipinti and Buono. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Teaching basic computer programming to young adults with intellectual disability

Maria Agatina Stimoli, Francesco Domenico Di Blasi*, Silvestro Maccarrone, Angela Antonia Costanzo, Paola Occhipinti and Serafino Buono

Unit of Psychology, Oasi Research Institute-IRCCS, Troina, Italy

Introduction: Teaching computer programming can be challenging, especially for individuals with intellectual disability (ID) who exhibit a wide range of learning abilities and behavioral characteristics. This study aimed to investigate the effectiveness of an intervention designed to teach computer programming skills to individuals with ID.

Method: Four women with ID, aged 27 to 54 years, were selected to participate in the study. Participants were taught computer programming using authoring software to create multimedia educational activities. A discrete trial teaching (DTT) approach was employed to teach specific skills and to gradually fade prompts to promote independent learning. A *multiple-probe design* across subjects was used to evaluate the effectiveness of the intervention. This design involved a baseline phase, a training phase with a most-to-least prompting procedure, and a 1-month follow-up phase to assess skill maintenance.

Results: The results demonstrated that all participants were able to acquire the necessary programming skills and complete the assigned tasks independently.

Conclusion: Computer programming can provide valuable learning and development opportunities for individuals with ID. However, it is essential to tailor the instruction to individual needs and provide appropriate support.

KEYWORDS

intellectual disability, explicit instruction, software authoring tool, inclusion, teaching

Introduction

Since the second half of the last century, we have been witnessing a process of social transformation defined as computer literacy driven by the widespread adoption of personal computers and other digital devices in a wide range of work, training, and teaching activities, as well as in daily life. In the present century, programming has emerged as a new form of literacy, empowering individuals to learn to code or program in order to become active creators of digital information, rather than merely passive consumers (Kereluik et al., 2013).

Programming can be integrated into various educational and training contexts as it fosters the development of crucial cognitive skills (Papert, 2005). Programming sessions can therefore be integrated into a number of training, school, and educational activities as they can contribute to improving thinking skills in learning processes (Papert, 2005). Such skills involve the use of cognitive mechanisms (problem formulation, problem-solving, recursive thinking, abstraction, decomposition, error correction, and reasoning) essential for success in solving problems encountered in everyday life (Wing, 2006).

Individuals with intellectual disability (ID) are often excluded from learning complex skills like computer programming due to their unique needs and challenges (Wille et al., 2017). ID is in fact characterized by significant limitations in both intellectual and adaptive functioning, encompassing various degrees of cognitive ability and often co-occurring with other neurodevelopmental disorders and various genetic syndromes (Schalock et al., 2021). As a result, teaching programming to students with ID can be particularly challenging considering their different IDs, ages, learning levels, and behavioral characteristics. To date, there are in fact limited studies on teaching programming addressed to people with ID or autism spectrum disorder (ASD) (Luxton-Reilly et al., 2018). Research has shown that children and adolescents with intellectual disability (ID) or Down's syndrome can acquire basic programming and computational skills through explicit instruction, the use of concrete manipulatives (e.g., physical coding blocks), and tangible interfaces (e.g., robots). These approaches have been found to improve cognitive abilities, including episodic memory, executive functions, visuospatial skills, social interaction, and emotional–motivational regulation (Besio et al., 2008; Businaro et al., 2014; Taylor et al., 2017; Taylor, 2018; Bargagna et al., 2019; González-González et al., 2019; Di Lieto et al., 2020; Kolne et al., 2021; Kert et al., 2022; Pérez et al., 2021; Shahid et al., 2022; Ellis et al., 2023; Baek et al., 2024; Graßl and Fraser, 2024; Kim et al., 2024). Adults with ID can also benefit from learning programming through guided activities, video tutorials, and explicit instruction, as demonstrated by Koushik and Kane (2019). Similarly, children with autism spectrum disorder (ASD) and ID can develop computational thinking skills through explicit instruction, guided activities, and interaction with programming software platforms (Snodgrass et al., 2016; Albo-Canals et al., 2018; Ketenci et al., 2022; Sola-Özgüç and Altın, 2022; Gkiolnta et al., 2023). While various technologies and methodologies have been used to teach programming to individuals with ID and ASD, there is still a lack of consensus on the most effective assessment methods and instructional approaches. Currently, evaluation often relies on subjective observations and interviews with students and instructors (De Araújo and Andrade, 2021). "Explicit instruction" has emerged as a promising approach for teaching IT skills to this population (Israel et al., 2015a, 2015b; Sola-Özgüç and Altın, 2022; Baek et al., 2024). This approach involves breaking down complex tasks into smaller, more manageable steps, providing clear demonstrations, offering guided practice, providing immediate feedback to reinforce positive responses and correct negative ones, and promoting generalization of the learned skill to independent use (for a review see Hughes et al., 2017). Discrete trial teaching (DTT) aligns with the principles of an explicit instruction by promoting skill acquisition in a structured setting; instructor support (prompts) prevents students from making mistakes and therefore gradually learn new skills; as students gain proficiency, the instructor gradually reduces the amount of support, eventually allowing them to perform the skill independently and in different conditions; finally, this systematic approach maximizes learning and reinforces correct responses by providing frequent opportunities for practice and feedback (Hughes et al., 2017).

Several approaches can be used to teach programming, including robotics, block-based programming, unplugged programming, and authoring software. For this study, we chose to use authoring software, as it enables individuals without prior programming experience to create multimedia educational activities by combining various elements such as text, images, audio, and animations. JClic, (Xarxa Telemàtica Educativa de Catalunya, 1992) an open-source authoring software platform distributed under the GNU GPL license, provides a user-friendly interface for creating various types of interactive multimedia content.

Given that research is limited but growing, and is characterized by heterogeneity in methodologies, technologies used, and sample characteristics (e.g., age, cognitive-behavioral profile, and sample size), it presents limitations that currently do not allow for the sharing of solid evidence useful for educational practice. However, the potential benefits for individuals with ID resulting from learning computer programming encourage us to address this issue, as it can represent, along with other interventions, a means to increase self-esteem and efficacy, to strengthen the perception of oneself as a competent person, and to create new inclusive opportunities.

This study aimed to teach four young adults with ID how to use JClic authoring software to create multimedia projects. The primary goal was to develop their basic IT skills through a discrete trial teaching approach.

Materials and methods

Participants, settings, and materials

The study participants were four young women aged 27 to 54 years who regularly participate in habilitation sessions at the institute. These sessions focused on developing autonomy, environmental management skills, cognitive stimulation, and occupational therapy. The participants were well-integrated into the residential context, adhering to daily routines and actively engaging in their individualized habilitation programs.

All participants presented with intellectual disability, varying in severity, and often co-occurring with other neurodevelopmental disorders (Table 1). Prior to the study, the participants had demonstrated enthusiasm and motivation for technology-based activities. Written informed consent was obtained from the participants' parents to publish this article.

To determine the participants' preferences, interviews were conducted with their instructors, who had extensive knowledge of their

TABLE 1 Cognitive and behavioural characteristics of participants.

	Sex	Chronological age	Diagnosis	Cognitive level	Characteristics
M	Female	27	Autism spectrum disorder	Mild ID	Cognitive rigidity; verbal perseverations; emotional lability
S	Female	54	Personality disorder NOS, obsessive-compulsive disorder	Borderline intellectual functioning	Mood instability; anxious-depressive symptoms
G	Female	35	Personality disorder NOS; epilepsy.	Mild ID	Irritability; low frustration; tolerance
D	Female	30	Prader–Willi syndrome	Moderate ID	Cognitive rigidity; skin-picking

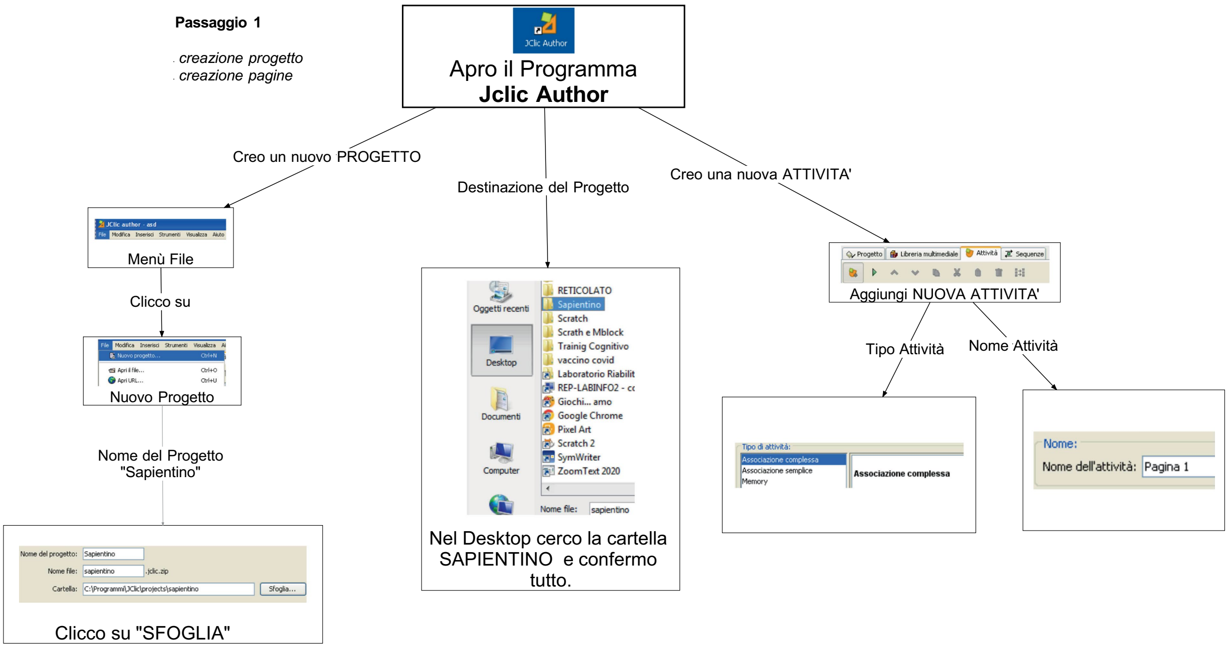

FIGURE 1
Textual prompt (concept map).

individual needs and interests. Given the participants' high level of motivation for computer-based activities, a formal preference assessment was deemed unnecessary. Positive reinforcement, in the form of verbal praise, was sufficient to maintain engagement throughout the training sessions. For participant G, additional motivating reinforcers, such as listening to music or playing games on the computer, were occasionally used to encourage continued engagement and adherence to the training protocol. This was necessary because participant G sometimes exhibited tendency to deviate from the established procedures after achieving initial success.

The study was conducted in a multimedia rehabilitation laboratory equipped with a desk, cabinets, and a computer. The computer was installed with JClic software, a Java-based platform that is used on various operating systems (Musilli, 2014). JClic comprises four modules: (1) JClic Author: For creating, editing, and publishing JClic projects; (2) JClic Player: For running JClic off-line; (3) JClic Applet: For embedding JClic projects on web page; (4) JClic Reports: For collecting and analyzing user data.

Each participant was seated individually at a computer station, and the instructor provided the following instruction "Create a JClic complex associations project."

A task analysis was created, outlining five key steps (see Figure 1) required to complete a complex associations project.

For the unplugged activities, a visual representation of an electrical circuit was used. This representation consisted of a physical circuit diagram and a set of colored pictograms paired with corresponding written labels. Correctly connecting the pictograms and labels would illuminate an LED bulb. The materials used for this activity included adhesive copper tape, paper, an LED bulb, and a 3-volt button cell (see Figure 2).

The following materials were used for the connection box (Figure 3): a circuit board with 200 holes; two 50-cm red and yellow wires; two 50-cm yellow wires with red and yellow alligator clips; a green LED; a 3-volt button cell; a battery holder for a 3-volt button cell; eight sample holders; various slips of paper with colored pictograms and corresponding written labels (e.g., Apple, Dog, Sun, House); a storage box to hold the electrical components; a 30×25 cm

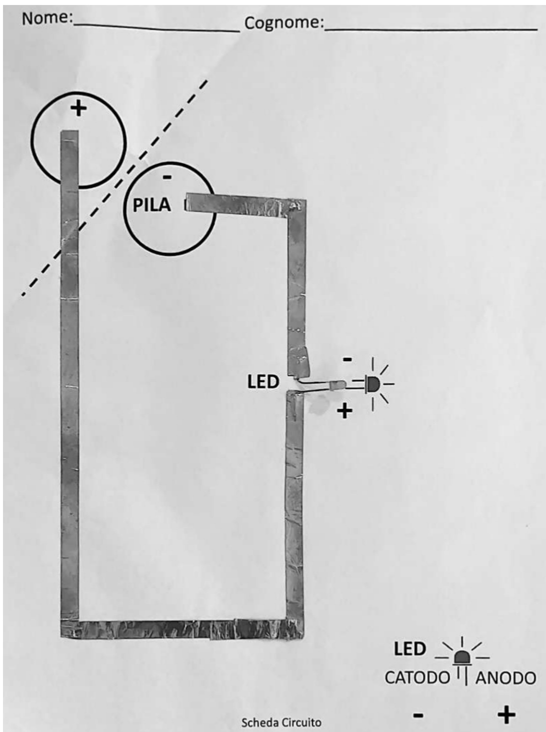

FIGURE 2
Electrical circuit scheme.

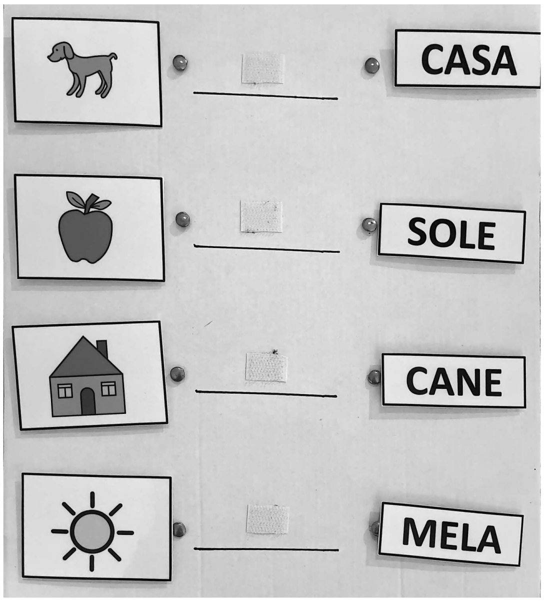

FIGURE 3
Connection box.

cardboard base to hold the slips and labels; 50-cm hook-and-loop fastener (Velcro); adhesive copper tape.

Procedures

Unplugged activity

The training was preceded by two unplugged activities, without computers or other electronic devices. One was carried out to introduce the concept of electrical circuits and one to introduce the concept of complex associations.

For each activity, conducted at separate times, participants worked in a group setting with an instructor to assemble the necessary materials for creating an electrical circuit and a connection box. The instructor first demonstrated the assembly process, and then, each participant followed along, replicating the steps. When needed, the instructor provided additional guidance, such as repeating instructions or offering verbal prompts.

The group setting fostered a collaborative learning environment, allowing participants to learn from each other by observing and interacting.

Pre-training

To ensure that participants possessed the necessary basic computer skills, a pre-assessment was conducted. All four participants demonstrated the required competence.

Experimental design

A multiple-probe design was used to evaluate the effectiveness of an instructional procedure involving prompting and reinforcement. The intervention aimed to teach four individuals with intellectual disabilities how to create complex associations projects using computer programming software. Each trial involved five steps (concept maps), and participant performance was measured by calculating the percentage of correct responses. A baseline phase was conducted for all four participants, during which they were not provided with any assistance or reinforcement. Training began with the first participant, who received instruction using a graduated prompting procedure (most-to-least prompts). Once the first participant achieved mastery, defined as 80% correct responses across three consecutive sessions, training commenced for the second participant and so on. The training continued until each participant could independently perform the entire sequence of steps without errors or prompts on two consecutive trials.

Measurement and interobserver agreement

To measure participant performance, a trial-by-trial data collection sheet was used. This sheet outlined the five steps involved in completing the task (based on the concept maps). Each step was scored as either correct (+) or incorrect (−), and the percentage of correct responses was calculated. The observers recorded, each on their own sheet, the answers obtained according to the conditions established in the teaching procedure. The percentage of correct answers for each of the five steps (concept maps) completed by the participants was calculated. Interobserver agreement (IOA) was assessed by having a second observer independently record participant performance. IOA was calculated by dividing the number of agreements between the two observers by the total number of trials and multiplying by 100 (Cooper et al., 2007). The calculated IOA was 90%.

Baseline

During the baseline phase, participants were asked to independently start JClic program and attempt to create a new project. No prompts or reinforcement was provided. While participants were able to initiate the program, they were unable to progress further in the project creation process.

Training

A discrete trial training approach was used, employing a total task chaining procedure. This involved breaking down the task into smaller steps (based on the task analysis) and providing instruction and support for each step. The instructor gradually reduced assistance as the participant demonstrated mastery of each step, ultimately enabling independent performance of the entire task.

Two training sessions were conducted per week. Initially, each session consisted of a single 45- to 50-min trial. As participants became more proficient, the session duration decreased to 10–15 min, allowing for the inclusion of two trials per session. The training started with the first participant. The instructor provided verbal instruction to create a complex associations project on JClic authoring, showing the program on the PC; the instructor carried out the procedure on the PC (modeling) and at the same time provided a textual prompt (five maps to follow for the realization of the project); the participant was then invited to carry out the steps indicated in the maps, and in the first phase, she was accompanied by total gestural and verbal prompts from the instructor for the implementation of the project until reaching the established mastery criterion (i.e., three consecutive sessions per day at 80%) reinforcing the correct execution of the steps with praise. The teaching was divided into six phases (conditions), in each of which the prompts were reduced (fading), to facilitate the acquisition of the procedure in complete autonomy (Table 2). Therefore, in the first condition, the participant received modeling, verbal prompts, gestural prompts, and written instructions. In the second condition, verbal and textual prompts were provided. In the third condition, only written instructions were given. The fourth condition involved no prompting. In the final two conditions, generalization was promoted by introducing new images, varying the content of the tasks, and having different instructors provide instruction. The participants were used to working with the IT expert educator with whom they have established a positive and supportive relationship. During the generalization phase, a different instructor, who was known to the participants but not in a close capacity, was introduced to the training setting.

An error correction procedure was incorporated into the training. Whenever a participant made a mistake, the instructor provided immediate feedback and guidance, including verbal instructions, physical prompts, and written cues. This process was repeated until the participant successfully completed the step independently.

The teaching was divided into six phases (conditions): in each phase, prompts were gradually reduced, starting with the least intrusive ones (such as gestures and visuals) to minimize intervention and promote autonomy.

- Condition 1: modeling and total gestural and verbal prompt, textual prompt (map display).
- Condition 2: textual prompt, verbal prompt.
- Condition 3: textual prompt.
- Condition 4: no prompt.
- Condition 5: generalization with different images.
- Condition 6: generalization with different instructors.

TABLE 2 Teaching procedure.

Total task chaining
• Introduce the PC and JClic program
• Provide verbal instruction
• Run the activity on JClic with most-to-least prompts type:
1. Modeling, total gestural and verbal prompt, textual prompt
2. Textual and verbal prompt
3. Text prompt
4. No prompts
5. Generalization with different images
6. Generalization with different instructors
• Reinforce the performance at the end of the chain with praise
Error correction
• Every time the participant makes a mistake, take her back to the last step of the correctly performed sequence

Each training session followed a structured sequence involving the participant, the instructor, the computer, and JClic software. The instructor began by verbally instructing the participant to create a new complex associations project in JClic. A modeling phase followed, where the instructor demonstrated each of the five steps on the computer. The participant was invited to replicate the five steps previously performed by the instructor and was immediately provided with verbal prompts, total gestures, and visual cues (e.g., task analysis maps). Correct answers were reinforced with verbal praise ("very good, you did well") and a positive "+" mark on the data sheet; incorrect responses or non-responses within 5 s were followed by an error correction procedure, a negative "-" mark on the data sheet and a re-demonstration with verbal, gestural, and textual prompts and additional prompts.

If the participant responded correctly within 5 s, they received less emphatic praise ("good"). This involved gradually reducing the level of prompting as the participant's performance improved. Once the participants achieved independent performance (condition 4), they were tasked with creating associations using new, unfamiliar images. Participants were asked to create new associations with new images (not those used during training). Subsequently, a generalization phase was planned with different instructors delivering instructions in the absence of the initial trainer. This aimed to determine whether the participant could apply their learned skills to new situations and with different individuals.

Follow-up

A follow-up assessment was conducted approximately 1 month after the conclusion of the training. No activity related to JClic software was carried out this month. During this assessment, the participants were asked to complete JClic project without any prompts or reinforcement, similar to the baseline phase.

Results

Figure 4 shows an overall view of the effectiveness of the treatment. All four participants successfully met the established training criteria, thus eliminating the need for further instruction.

In the training phase, an average of four sessions were conducted for each condition. In particular, we can observe that the first participant D maintained a fairly constant trend, she performed the task with great enthusiasm; initially, she tried to avoid following the maps and the procedure as proposed by the instructor, but gradually, she accepted the guidance provided through the prompts provided: 1 (modeling and total gestural and verbal prompt, textual prompt) and 2 (textual prompt, verbal prompt). Once she gained awareness of the task, already in condition 4 (no prompt) she began to build the program deviating from the prescribed steps and incorporating her own ideas. She demonstrated creativity by suggesting the use of new images. By the end of the training, she exhibited improved focus and organizational skills, planning her work and utilizing notes.

The second participant S also maintained a consistent level of engagement throughout the training. She closely followed the

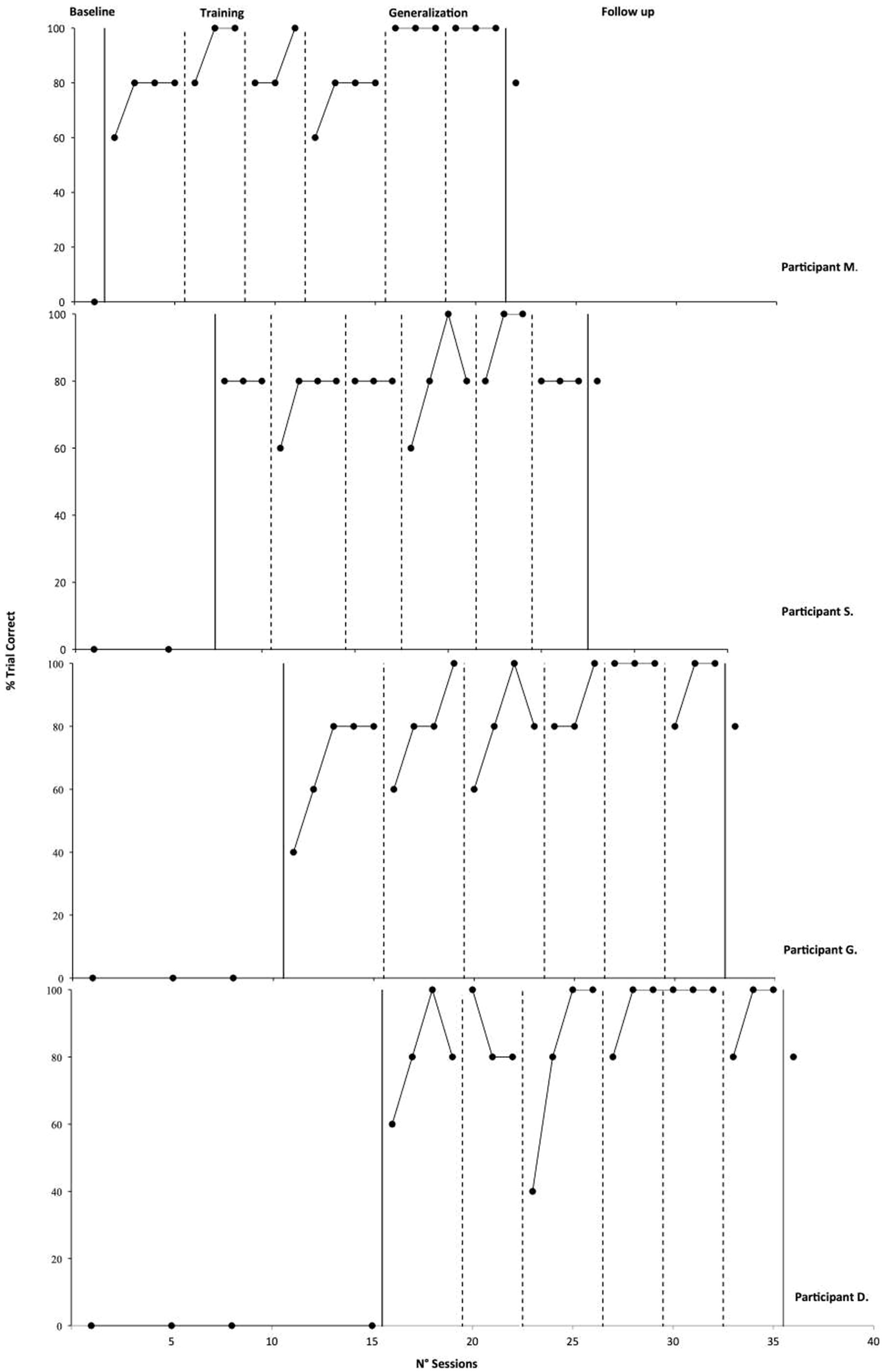

FIGURE 4
Percentages of correct trials in baseline, training, and follow-up for the four participants. Legend: Multiple-probe design across subjects showing the percentage of correct answers for each test by four young adults with intellectual disabilities (participants D, S, M, and G). The solid vertical lines indicate the phase change (baseline, training, and follow-up). The dashed vertical lines indicate the change in condition which coincides with the fading steps (prompts most to least).

provided instructions and demonstrated enthusiasm for the activity.

The third participant M exhibited a more variable performance, often appearing insecure and reliant on prompts. She frequently displayed anxiety and agitation, requiring reassurance to complete the tasks. In condition 4 (no prompt), she became frustrated with errors but eventually found strategies to resolve them independently, ultimately meeting the mastery criterion.

The fourth participant G also met the training objectives, although her performance was more inconsistent. She often became frustrated with repetitive tasks and required additional motivation, such as computer games or music, to maintain engagement. With these additional reinforcers, she was able to complete the training successfully.

A follow-up assessment was conducted 1 month after the completion of the training. All four participants achieved a score of 80% or higher on the assessment, indicating maintenance of the acquired skills.

Discussion

From the results obtained, we can state that teaching basic computer programming using JClic authoring software proved to be effective as the four participants successfully learned to create a project of complex associations aimed at creating multimedia educational activities.

While JClic was not specifically designed for individuals with disabilities, it was able to facilitate their access to computer programming and overcome certain barriers. This aligns with the International Classification of Functioning, Disability, and Health (ICF) framework, which emphasizes the importance of providing support to enable participation in various activities (World Health Organization (WHO), 2001).

The combination of JClic authoring software and explicit instruction proved effective in teaching computer programming skills to individuals with intellectual disabilities. This finding aligns with previous research demonstrating the efficacy of explicit instruction in teaching various skills, including academic skills (Butler et al., 2001; Knight et al., 2012; Bakken et al., 2021; Çapraz, 2023; Rodgers and Loveall, 2023; Schöld et al., 2023; Sulu et al., 2023), also for computer skills (Israel et al., 2015a, 2015b; Sola-Özgüç and Altın, 2022). Explicit instruction, which involves breaking down complex tasks into smaller, more manageable steps and providing immediate feedback, is a key component of effective teaching for individuals with intellectual disabilities. This approach aligns with behavioral chaining techniques, where tasks are broken down into smaller steps and taught sequentially (Hughes et al., 2017). The use of the aforementioned methodology has been previously proposed in educational approaches for individuals with disabilities, and the results of our study are a further confirmation of the effectiveness of *how* to teach this special population. They represent a novelty applied to *what* to teach, namely the acquisition of skills in the specific field of information technology, and *to whom* adult people with intellectual disabilities with associated behavioral problems.

The multimedia products created by the four participants with ID were subsequently used in rehabilitation activities for children and adolescents with various levels of disability. Their direct involvement, albeit in an executive role, in the creation of this basic multimedia software represents an innovative approach. This experience has empowered individuals with ID, enhancing their self-awareness and sense of competence in the field of IT. By contributing to the creation of educational materials, they have assumed a role of responsibility and gained recognition for their abilities. This aligns with the principles of inclusive education and promotes the dignity and wellbeing of individuals with ID. Accessing these conditions allows individuals with ID to assume more adult-like roles, facilitating the transition to adulthood and challenging the traditional image of individuals with ID as passive recipients of care. This approach, rooted in the principles of person-centered care, respects individual differences and promotes autonomy (Caldin and Scollo, 2018). It is also important to underline that the educational intervention described in our study was motivated by the participants' genuine interest in learning computer programming. This intrinsic motivation served as a powerful driver for their engagement and progress. By tailoring the intervention to their individual needs and interests, we were able to further stimulate their desire to become more independent and self-directed. There is currently a broad consensus on the role for digital and information technologies in promoting inclusion for individuals with ID. The United Nations (2006) has recognized access to these technologies as a fundamental human right. It is equally true, however, that the process of digital literacy, and we could add, computer literacy of people with ID is still very far from being fully realized. Making people with ID computer literate is a challenging task, as the rapid advancements in technology often exacerbate the digital divide between those who can access and use technology and those who cannot (Lussier-Desrochers et al., 2017; Chadwick et al., 2019; Johansson et al., 2021). From a psycho-pedagogical point of view, basic computer literacy can also have important implications in the area of cognitive and adaptive rehabilitation to support people with various forms of intellectual and behavioral disabilities. In fact, computer programming can offer learning opportunities proposed through multifaceted stimuli in a context of greater originality, and cognitive and relational flexibility. While it should not replace traditional clinical and rehabilitative interventions, it can be integrated into a comprehensive support plan to enhance the quality of life for individuals with ID. All this can have an inclusive value that still needs to be better explored today.

Conclusion

Our study presents research on the effectiveness of an intervention applied to four women with ID, with varying learning levels and behavioral characteristics, on basic computer programming skills using JClic authoring software. The results obtained provide further evidence of the feasibility of interventions to promote computer programming education for individuals with intellectual disabilities. The reasons for these results are likely due to a combination of the programming approach using authoring software, the teaching methodology based on discrete trial teaching, which falls within the explicit instruction approach, and the characteristics of the participants, for whom this experience was both motivating and empowering.

However, our study also has a number of limitations. The sample used in this study was limited to only four female participants. To generalize the findings of the study, future research should replicate these results across diverse populations with ID (e.g., genetic syndromes) and various age groups.

Regarding the methodology, the maintenance probe was administered 1 month after the final independent performance. In the future, it would be necessary to plan for multiple follow-up periods (e.g., 6, 12, 18 months) to observe the long-term maintenance of acquired skills.

Considering that specific software for individuals with ID is often unavailable and that the choice is usually dictated by cognitive-behavioral characteristics of the individuals in relation to the software rather than the other way around, future research should include the use of different software for learning basic computer programming.

The challenge posed by this new area of learning can also have significant implications in the area of cognitive and adaptive rehabilitation to support people with various forms of intellectual and behavioral disabilities. Future research in this area should incorporate basic programming activities into individualized plans, tailored to the cognitive-behavioral characteristics and the level of support required, as computer programming can offer learning and inclusive opportunities.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Ethics statement

The studies involving humans were approved by the Comitato Etico IRCCS Sicilia-Oasi Maria SS." Prot. CE/193, as of 5 April 2022, approval code: 2022/04/05/CE-IRCCS-OASI/52. The studies were conducted in accordance with the local legislation and institutional requirements. Written informed consent for participation in this study was provided by the participants' legal guardians/next of kin. Written informed consent was obtained from the individual(s), and minor(s)' legal guardian/next of kin, for the publication of any potentially identifiable images or data included in this article.

Author contributions

MS: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. FB: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. SM: Data curation, Writing – review & editing. AC: Data curation, Writing – review & editing. PO: Data curation, Writing – review & editing. SB: Supervision, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This study was supported by a grant from the Italian Ministry of Health—Ricerca Corrente 2022–2024.

Acknowledgments

Special acknowledgements for this paper are due to Eleonora Di Fatta for her valuable assistance in the translation, preparation, and formatting of the text. We thank Cristina Citerei, BCBA psychologist, for the supervision of the research design.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Albo-Canals, J., Martelo, A. B., Relkin, E., Hannon, D., Heerink, M., Heinemann, M., et al. (2018). A pilot study of the KIBO robot in children with severe ASD. *Int. J. Soc. Robotics* 10, 371–383. doi: 10.1007/s12369-018-0479-2

Baek, C., Saito-Stehberger, D., Nam, A., Lopez, G., Jacob, S., and Warschauner, M. (2024). Empowering Latine elementary school students with disabilities: computer programming through culturally sustaining curriculum. *Comput. Sci. Educ.* 1-26, 1–26. doi: 10.1080/08993408.2024.2396749

Bakken, R. K., Næss, K.-A. B., Lemons, C. J., and Hjetland, H. N. (2021). A systematic review and Meta-analysis of Reading and writing interventions for students with disorders of intellectual development. *Educ. Sci.* 11:638. doi: 10.3390/educsci11100638

Bargagna, S., Castro, E., Cecchi, F., Cioni, G., Dario, P., Dell'Omo, M., et al. (2019). Educational robotics in down syndrome: a feasibility study. *Tech. Know. Learn.* 24, 315–323. doi: 10.1007/s10758-018-9366-z

Besio, S., Caprino, F., and Laudanna, E. (2008). "Profiling robot-mediated play for children with disabilities through ICF-CY: the example of the European project IROMEC" in Computers helping people with special needs. ICCHP 2008. Lecture notes in computer science. eds. K. Miesenberger, J. Klaus, W. Zagler and A. Karshmer, vol. *5105* (Berlin, Heidelberg: Springer), 545–552.

Businaro, N., Zecca, L., and Castiglioni, M. (2014). Implicazioni psicologiche di un laboratorio di robotica educativa nella scuola primaria: riflessioni sul caso di un bambino con ritardo mentale. *Psicol. Clin. Svilupp.* 2, 311–318. doi: 10.1449/77641

Butler, F. M., Miller, S. P., Lee, K., and Pierce, T. (2001). Teaching mathematics to students with mild-to-moderate mental retardation: a review of the literature. *Ment. Retard.* 39, 20–31. doi: 10.1352/0047-6765(2001)039<0020:TMTSWM>2.0.CO;2

Caldin, R., and Scollo, S. (2018). Inclusione lavorativa, disabilità e identità. Riflessioni e rappresentazioni. [Inclusion at work, disability and identity. Considerations and Rappresentations]. *Studiun Educationis-Rivista semestrale per le professioni educative* 3, 49–60.

Çapraz, C. (2023). Teaching of solid, liquid and gas substances to students with intellectual disabilities through direct instruction method. *Int. e-Journal Educ. Stud.* 7, 170–179. doi: 10.31458/iejes.1238841

Chadwick, D. D., Chapman, M., and Caton, S. (2019). "Digital inclusion for people with an intellectual disability" in Oxford handbook of Cyberpsychology. eds. I. A. Attrill, C. Fullwood, M. Keep and D. Duss (Oxford: Oxford University Press), 261–284.

Cooper, J. O., Heron, T. E., and Heward, W. L. (2007) in Applied behavior analysis. New York: Pearson/Merrill-prentice hall. eds. S. Izzo and J. Dean (Milano: Consorzio Universitaio Humanitas).

De Araújo, E. C. J., and Andrade, W. L. (2021). "A systematic literature review on teaching programming to people with cognitive disabilities" in IEEE Frontiers in education conference (FIE) (Lincoln, NE, USA: IEEE), 1–8.

Di Lieto, M. C., Castro, E., Pecini, C., Inguaggiato, E., Cecchi, F., Dario, P., et al. (2020). Improving executive functions at school in children with special needs by educational robotics. *Front. Psychol.* 10:2813. doi: 10.3389/fpsyg.2019.02813

Ellis, K., Kruesi, L., Ananthanarayan, S., Senaratne, H., and Lindsay, S. (2023). Piece it together": insights from one year of engagement with electronics and programming for people with intellectual disabilities. *Proceed. 2023 CHI Conf. Human Fact. Comput. Syst.* 219, 1–17. doi: 10.1145/3544548.3581401

Gkiolnta, E., Zygopoulou, M., and Syriopoulou-Delli, C. K. (2023). Robot programming for a child with autism spectrum disorder: a pilot study. *Int. J. Dev. Disabil.* 69, 424–431. doi: 10.1080/20473869.2023.2194568

González-González, C. S., Herrera-González, E., Moreno-Ruiz, L., Reyes-Alonso, N., Hernández-Morales, S., Guzmán-Franco, M. D., et al. (2019). Computational thinking and down syndrome: an exploratory study using the KIBO robot. *Informatics* 6:25. doi: 10.3390/informatics6020025

Graßl, I., and Fraser, G. (2024). Coding to cope: teaching programming to children with emotional and behavioral disorders. In Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '24) (New York, NY, USA: Association for Computing Machinery), 127–138. doi: 10.1145/3639474.3640078

Hughes, C. A., Morris, J. R., Therrien, W. J., and Benson, S. K. (2017). Explicit instruction: historical and contemporary contexts. *Learn. Disabil. Res. Pract.* 32, 140–148. doi: 10.1111/ldrp.12142

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., and Reese, G. (2015a). Supporting all learners in school-wide computational thinking: a cross-case qualitative analysis. *Comput. Educ.* 82, 263–279. doi: 10.1016/j.compedu.2014.11.022

Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S., and Tapta, T. (2015b). Empowering K-12 students with disabilities to learn computational thinking and computer programming. *Teach. Except. Child.* 48, 45–53. doi: 10.1177/0040059915594790

Xarxa Telemàtica Educativa de Catalunya. (1992). JClic. Available at: https://projectes.xtec.cat/clic/en/

Johansson, S., Gulliksen, J., and Gustavsson, C. (2021). Disability digital divide: the use of the internet, smartphones, computers and tablets among people with disabilities in Sweden. *Universal Access Inf.* 20, 105–120. doi: 10.1007/s10209-020-00714-x

Kereluik, K., Mishra, P., Fahnoe, C., and Terry, L. (2013). What knowledge is of most worth: teacher knowledge for 21st century learning. *J. Digit. Learn. Teach. Educ.* 29, 127–140. doi: 10.1080/21532974.2013.10784716

Kert, S. B., Yeni, S., and Fatih Erkoç, M. (2022). Enhancing computational thinking skills of students with disabilities. *Instr. Sci.* 50, 625–651. doi: 10.1007/s11251-022-09585-6

Ketenci, T., Dwortz, C., and Chonoeva, N. (2022). "Online computing summer camp with non-verbal students with autism spectrum disorder" in 45th annual proceedings of the Association for Educational Communication and Technology (AECT 2022), (Las Vegas, NE, USA: AECT), vol. 1 & 2, 178–189.

Kim, M., Kim, J., and Lee, W. (2024). Enhancing computational thinking in students with autism spectrum disorder and intellectual disabilities: a robot programming approach. *Int. J. Dev. Disabil.* 1–16, 1–16. doi: 10.1080/20473869.2024.2394735

Knight, V. F., Smith, B. R., Spooner, F., and Browder, D. (2012). Using explicit instruction to teach science descriptors to students with autism spectrum disorder. *J. Autism Dev. Disord.* 42, 378–389. doi: 10.1007/s10803-011-1258-1

Kolne, K., Bui, S., and Lindsay, S. (2021). Assessing the environmental quality of an adapted, play-based LEGO® robotics program to achieve optimal outcomes for children with disabilities. *Disabil. Rehabil.* 43, 3613–3622. doi: 10.1080/09638288.2020.1743776

Koushik, K., and Kane, S. K. (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). *Assoc. Comput. Mach.* 514, 1–12. doi: 10.1145/3290605.3300744

Lussier-Desrochers, D., Normand, C. L., Romero-Torres, A., Lachapelle, Y., Godin-Tremblay, V., Dupont, M. È., et al. (2017). Bridging the digital divide for people with intellectual disability. *Cyberpsychology, Journal of Psychosocial Research of Cyberspace* 11, 1–20. doi: 10.5817/CP2017-1-1

Luxton-Reilly, A., Becker, B. A., Ott, L., Simon Giannakos, M., Paterson, J., Albluwi, I., et al. (2018). A review of introductory programming research 2003–2017 In Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018) (New York, NY, USA: Association for Computing Machinery), 342–343. doi: 10.1145/3197091.3205841

Musilli, G. (2014). I software autore per la didattica. Ledizioni: Percorsi creativi nella scuola primaria.

Papert, S. (2005). "Teaching children thinking" in Contemporary issues in technology and teacher education 5, 353–365 (Waynesville, NC USA: Society for Information Technology & Teacher Education).

Pérez, J., Azuaje, M., León, C., and Pedroza, O. (2021). Effects of social robotics on episodic memory in children with intellectual disabilities. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje* 16, 393–399. doi: 10.1109/RITA.2021.3125899

Rodgers, D. B., and Loveall, S. J. (2023). Writing interventions for students with intellectual and developmental disabilities: a Meta-analysis. *Remedial Spec. Educ.* 44, 239–252. doi: 10.1177/07419325221108896

Schalock, R. L., Luckasson, R., and Tassé, M. J. (2021). An overview of intellectual disability: definition, diagnosis, classification, and systems of supports. *Am. J. Intellect. Dev. Disabil.* 126, 439–442. doi: 10.1352/1944-7558-126.6.439

Schöld, D., Östergren, R., Levén, A., Hassler-Hallstedt, M., and Träff, U. (2023). App-based mathematical intervention for youth with intellectual disabilities: a randomised controlled trial. *Scand. J. Educ. Res.* 68, 689–701. doi: 10.1080/00313831.2023.2175250

Shahid, N. M., Law, E. L. C., and Verdezoto, N. (2022). Technology-enhanced support for children with down syndrome: a systematic literature review. *Int. J. Chil Comput. Interact.* 31:100340. doi: 10.1016/j.ijcci.2021.100340

Snodgrass, M. R., Israel, M., and Reese, G. C. (2016). Instructional supports for students with disabilities in K-5 computing: findings from a cross-case analysis. *Comput. Educ.* 100, 1–17. doi: 10.1016/j.compedu.2016.04.011

Sola-Özgüç, C., and Altın, D. (2022). Teaching block-based coding to a student with autism spectrum disorder. *Ankara Univ. Facul. Educ. Sci. J. Special Educ.* 23, 565–594. doi: 10.21565/ozelegitimdergisi.822554

Sulu, M. D., Martella, R. C., Aydin, O., Bolshokova, V., and Erden, E. (2023). A Meta-analysis of science education studies for students with intellectual and developmental disabilities. *J. Dev. Phys. Disabil.* 35, 917–950. doi: 10.1007/s10882-023-09890-z

Taylor, M. S. (2018). Computer programming with pre-K through first-grade students with intellectual disabilities. *J. Spec. Educ.* 52, 78–88. doi: 10.1177/0022466918761120

Taylor, M. S., Vasquez, E., and Donehover, C. (2017). Computer programming with early elementary students with down syndrome. *J. Spec. Educ.* 32, 149–159. doi: 10.1177/0162643417704439

United Nations (2006). Convention on the rights of persons with disabilities. New York: United Nations.

World Health Organization (WHO). (2001). International classification of functioning, disability and health (ICF). (Geneva: WHO). Italian Translation (2002) ICF: Classificazione Internazionale del funzionamento della disabilità e della salute. Trento: Erickson.

Wille, S., Century, J., and Pike, M. (2017). Exploratory research to expand opportunities in computer science for students with learning differences. *Comput. Sci. Eng.* 19, 40–50. doi: 10.1109/MCSE.2017.43

Wing, J. M. (2006). Computational thinking. *Comm. ACM* 49, 33–35. doi: 10.1145/1118178.1118215