Effects of Unplugged Programming Education on Computational Thinking and Social Skills in Individuals with Developmental Disabilities

Seung-Mee Lee[1], Seok-Ju Chun[1]

[1] Seoul National University of Education, Republic of Korea

HOW TO CITE:
Lee, S-M., & Chun, S-J. (2025). Effects of Unplugged Programming Education on Computational Thinking and Social Skills in Individuals with Developmental Disabilities. *International Journal of Special Education, 40*(2), 67-78.
CORRESPONDING AUTHOR:
Seok-Ju Chun;
chunsj@snue.ac.kr
DOI:
https://doi.org/10.52291/ijse.2025.40.22

ABSTRACT:

This study aimed to examine the effects of unplugged programming education on computational thinking and social skills in individuals with developmental disabilities. As computational thinking is recognized as an essential competency in the digital age, it is important to promote inclusive access to programming education so that people with developmental disabilities have equitable opportunities to develop this skill.

A modified multiple-probe design across participants was used in this study. The study involved six individuals with developmental disabilities in the Republic of Korea. The intervention spanned 14 sessions, and unplugged programming education was implemented using tangible programming tools.

As a result, all participants showed improvements in computational thinking compared to the baseline, and these effects were maintained during the maintenance phase. In particular, the PND results showed strong effects on computational thinking (100%). In addition, all participants exhibited increased social skills, with PND values ranging from 50% to 100%, suggesting moderate to strong effects depending on the individual. These findings suggest that instructional strategies that consider the characteristics of individuals with developmental disabilities can effectively enhance both computational thinking and social skills. Future research should include a broader range of age groups and disability types, and it should explore the long-term effects of such educational interventions.

Keywords: Computational Thinking; Developmental Disabilities; Inclusive Programming; Unplugged Programming; Social Skills

INTRODUCTION

Recently, various software-based technologies have become key elements of national competitiveness and wealth creation. Accordingly, major countries are considering computational thinking as one of the future core competencies, and the importance of programming education that actively reflects this in the curriculum is being emphasized.

However, people with disabilities, including individuals with developmental disabilities such as autism spectrum disorder (ASD) or intellectual disability (ID), are experiencing many difficulties in the process of using digital technology, which is creating an information gap between disabled and non-disabled people. This information gap causes imbalances in educational, employment, and social participation opportunities, which also negatively affect the quality of life of individuals with developmental disabilities. The International Society for Technology in Education (ISTE) discusses what educators should do to help students develop computational thinking skills to solve problems they face, but students with special needs tend to be excluded from these discussions. In addition, research on programming education and computational thinking skills development for individuals with disabilities, especially those with ASD or ID, is still lacking (Karaman & Seferoğlu, 2024; Kim et al., 2025).

Individuals with developmental disabilities experience a range of developmental challenges, including language acquisition, social interaction, and behavior regulation, and have historically been disadvantaged in education (Elshahawy et al. 2020; Kim et al. 2024). Recognizing the potential of the ASD community, many companies are actively recruiting individuals with ASD to perform programming and software testing tasks or offering intensive, immersive programming instruction, but these limited efforts fall short of meeting the programming education needs of individuals with disabilities (Eiselt et al., 2018). When individuals with ASD or ID have opportunities to develop STEM (science, technology, engineering, and mathematics) interests with their peers, they will not only improve their scientific literacy in the short term, but also increase their chances of success in the job market in the long term (Knight et al., 2019).

The K-12 Computer Science Framework was designed to make computer science education accessible to all, including individuals with disabilities, and to support teachers in doing so effectively (Taylor et al., 2017). Individuals with ASD need to develop computational thinking skills to play productive roles in technology fields, rather than simply being consumers of technology (Sola-Özgüç & Altın 2022). Therefore, future research and policy should be directed toward supporting individuals with ASD, ID, and other disabilities to receive inclusive education with their typically developing peers, primarily by increasing access to STEM education (Wright et al., 2021). Individuals with developmental disabilities should also be provided with educational opportunities on the same basis as general education, and it is important to create a more comprehensive learning environment (Taylor 2018). Therefore, efforts are needed to create a disability-friendly educational environment and provide policy support, enabling programming education for individuals with developmental disabilities to spread more systematically in the future.

This study aims to develop and apply an unplugged programming education for individuals with developmental disabilities to increase their access to programming education. In addition, we aim to verify the effects of this education on the development of computational thinking and social skills.

To address these aims, the following research questions were formulated:

1. What effects does the unplugged programming education developed in this study have on the computational thinking of individuals with developmental disabilities?
2. What effects does the unplugged programming education developed in this study have on the social skills of individuals with developmental disabilities?

LITERATURE REVIEW

Computational thinking was first introduced by Papert (1980), and Wing (2006) defined computational thinking as a combination of problem formulation, problem solving, recursive thinking, abstraction, decomposition, error correction, and reasoning skills. Computational thinking is a process of solving problems encountered in daily life by utilizing the thinking methods developed in computer science. It includes designing computational solutions and applying algorithmic thinking. Based on this concept, the educational approach is shifting from "learning to code" to "coding to learn" (Kert et al., 2022). In particular, programming education plays an important role in developing competencies such as problem-solving, project design, and communication skills, which are considered important in today's society. These skills can be utilized as a tool to improve various thinking skills in the learning process (Papert, 2005).

Computational thinking is an important skill for everyone, regardless of disability (Bouck & Yadav, 2022). In this trend, research emphasizing the participation of individuals with disabilities in robotics and STEM education is increasing, and interest in how to address their special needs in educational design is growing (Hwang & Taylor, 2016; Yuen et al., 2014). Among these, studies focusing on individuals with developmental disabilities have shown that STEM activities and robotics education can positively impact enhancing social skills and increasing learning motivation.

Computational approaches to improving the skills of individuals with ASD have been developed since the 1960s, and various educational interventions have been implemented recently (Muñoz et al., 2018). Individuals with ASD tend to have strong visual processing abilities, spatial understanding, construction skills, attention to detail, and persistence, which lead to a high interest in STEM activities (Michalek et al., 2020). Accordingly, research on social skills education using STEM activities is being conducted. Studies have shown that when individuals with disabilities use robots, their motivation and interest in STEM increases (Lindsay & Lamb, 2018). In particular, all participants, including individuals with autism, experienced a high level of social interaction in robot-based collaborative projects, which proved effective in enhancing their social skills (Yuen et al., 2014). When programming a robot, you must learn to think before you act, which stimulates your ability to reason and plan. Since the robot provides feedback on planned actions, you can infer and self-analyze errors and performance. In addition, activities using educational robots promote cooperative learning in a playful environment (Bargagna et al., 2019). This educational approach plays an important role in developing not only academic skills but also social and communication skills.

Gkiolnta et al. (2023) reported that education using the robot 'Codey Rocky' helped improve the social and communication skills of students with ASD and ID. González-González et al. (2019) showed that the KIBO robot was effective in teaching programming and computational thinking skills to students with Down syndrome and in inducing their participation. Ratcliff and Anderson (2011) reported that Logo programming contributed to increasing self-esteem and learning motivation by providing interactive challenges and problem-solving experiences to students with disabilities. Knight et al. (2019) reported that the generalization and maintenance effects of coding skills were confirmed through robot coding education for elementary school students with ASD and severe problem behaviors. A study by Michalek et al. (2020) found that collaborative STEM activities, such as robot building, positively affected the social communication skills of adolescents with ASD and helped reduce social anxiety. However, it was pointed out that there are limits to improving verbal communication skills, and more structured interventions may be needed.

Education for all learners provides teachers with opportunities to promote values such as empathy, respect, and equity, which enrich the educational environment (Araujo et al., 2022). In particular, as computer science becomes included in the general education curriculum, educators must consider how to meaningfully include diverse learners (Gribble et al., 2017). Ensuring inclusive accessibility in programming education is an important task. In line with this recent trend, this study aims to evaluate the impact of unplugged programming education on individuals with developmental disabilities, focusing on enhancing computational thinking and social skills. Unplugged programming is a way to learn computational thinking and computer science concepts without computing devices, and includes role-playing, object manipulation, and physical activities (Demir, 2021). This approach helps them intuitively understand and participate in programming skills. In the study, participants apply computational thinking skills during the programming learning process and interact in various social situations.

METHODS

Unlike the traditional approach, where intervention is introduced at staggered time points across participants, this study implemented a fixed baseline period for all participants while conducting multiple probe assessments to ensure baseline stability. The training phase was initiated simultaneously for all participants after confirming stable performance across two consecutive baseline sessions. This approach was chosen not only to maintain experimental control and allow for structured comparisons across participants, but also to reflect the group-based instructional context in which programming education typically takes place. By modifying the traditional multiple-probe design in this way, the study ensured that all individuals received the same intervention period while still adhering to the principles of baseline stability assessment.

Participants

This study conducts an experiment using the modified multiple-probe design to verify the effectiveness of unplugged programming education for individuals with developmental disabilities. A total of six participants took part in the study, and all were registered as individuals with developmental disabilities according to the Welfare of People with Disabilities Act (Republic of Korea). In the selection process for research subjects, voluntary consent from participants and their caregivers was mandatory. Among those who met these requirements, participants suitable for the experiment were finally selected.

The study was conducted at a local welfare centre located in Gyeonggi-do, Republic of Korea, and six research participants were finally selected from this institution. The gender composition of the research participants consisted of two men and four women, and their ages ranged from their 20s to 30s. In terms of disability type, there were three individuals with ID and three individuals with ASD.

Several criteria were applied when selecting the research participants. First, individuals had to be able to participate in all 14 sessions of the study. Second, they needed sufficient cognitive and linguistic abilities to understand and follow the instructor's instructions. Third, participants without motor impairments were selected to operate the robot used in the study directly. Finally, individuals with no prior programming education experience were selected. These criteria were applied to increase the reliability and validity of the study.

Specific demographic information on the participants is presented in Table 1. Only gender, age, and disability type were collected. More sensitive information, such as IQ scores, detailed cognitive profiles, or communication ability, was not obtained due to ethical considerations.

Structure of the unplugged programming intervention

In this study, COBOBLOCKS was selected as an educational tool to support effective unplugged programming activities for individuals with developmental disabilities. As a tangible programming education tool, COBOBLOCKS enables users to learn programming anytime and anywhere without the need for a computer or monitor. Since individuals with developmental disabilities may have difficulty acquiring basic computer skills before learning programming, a simple and intuitive tool was adopted to ensure accessibility.

The intervention focused on problem-solving through unplugged programming education using COBOBLOCKS. Rather than merely introducing programming concepts, the educational program was designed to

Table 1. **Information of participants**

Participants	Gender	Age	Disability type
A	Female	23	ASD
B	Female	27	ASD
C	Male	24	ASD
D	Female	24	ID
E	Female	39	ID
F	Male	23	ID

Table 2. **Contents of programming education**

No.	Contents
1	- Understanding LED, Sound, and Dancing blocks
2	- Creating a sequential algorithm using LED, Sound, and Dancing blocks
3	- Understanding and adjusting the number of the Move Forward block
4	- Understanding and adjusting the number of Turn Left/Right blocks
5	- Understanding how changing the block sequence affects robot movement
6	- Sequencing movement blocks
7	- Creating a repeating algorithm
8	- Understanding infinite loops

help participants actively solve problem situations using block-based commands. A variety of programming activities from previous studies (Lee et al., 2024) were referenced and adapted into a structured program that gradually increased in difficulty. Each session progressively introduced more complex tasks, starting from basic programming concepts and moving toward more advanced problem-solving. The educational program also combined individual and cooperative learning to promote social interaction among participants with developmental disabilities. The specific contents of the intervention program are presented in Table 2.

Assessment tool

In this study, a rating scale was used to measure the computational thinking and social skills of individuals with developmental disabilities. Since the participants had difficulty reading the items and writing answers independently, the evaluation was conducted by two individuals: a disability welfare specialist familiar with the participants and a researcher involved in the study. To ensure objectivity and accuracy, both evaluators closely observed and recorded the participants' behaviors throughout the sessions.

The computational thinking assessment aimed to measure core sub-elements of computational thinking, as discussed in previous studies (Wing, 2006; Wing, 2008; Brennan & Resnick, 2012). Drawing from Wing's emphasis on abstraction, decomposition, and algorithmic reasoning, and Brennan and Resnick's categorization of computational thinking into concepts (e.g., sequences, loops, conditionals, data), practices, and perspectives, this study focused on evaluating key concepts. To secure the content validity of the assessment, it was reviewed by two computer education teachers and two special education teachers. The reliability verification result showed a very high Cronbach's α value of 0.972.

The assessment consisted of a total of 10 items and was subdivided into problem understanding (3 items), abstraction (2 items), algorithm (3 items), and automation (2 items). Each item was evaluated using a 5-point Likert scale. Representative items include: 'Identifying the current state of a problem and the goal state to be achieved', and 'Modifying and refining solutions to successfully operate the robot when the initial attempt fails'. The total score was calculated out of 50 points, and the average value based on the 5-point scale was presented during the analysis process. The computational thinking assessment was conducted across all sessions from the baseline to the maintenance phase. While maintaining the same level and content for each session, some items were adjusted according to the learning topic to enable more precise measurement.

The social skills assessment was organized into three domains: self-assertion, cooperation, and self-control. A total of 10 items were developed: 4 items each for self-assertion and cooperation, and two items for self-control. The same observational procedures used for the computational thinking assessment were applied during each session. To ensure content validity, the assessment tool was reviewed by two special education teachers. The items were adapted and modified from the Social Skills Rating System (Gresham & Elliott, 1990) to suit the characteristics of the participants in this study. All items were checked based on the number of times the researcher observed them, enabling a quantitative analysis of the social skills changes in the research participants. The internal consistency of the adapted scale was high, with a Cronbach's α of 0.965.

Experiment

In this study, an experiment was conducted to analyze the effectiveness of unplugged programming education using COBOBLOCKS. The experiment was conducted once a week for a total of 14 sessions, and each session lasted 40 minutes. The experimental procedure was divided into four phases: baseline, training, intervention, and maintenance. At each phase, their computational thinking and social skills were assessed through observation.

The first phase of the experiment, the baseline phase, lasted two weeks. During this period, no intervention was provided. Participants were given COBOBLOCKS and a play map to explore freely without researcher involvement. If participants showed stable assessment scores for two consecutive sessions, they proceeded to the next phase—the training phase.

The training phase lasted for two weeks, during which basic programming methods using COBOBLOCKS were taught. The participants engaged in activities that involved controlling the movement of the robot car using various coding blocks. During this learning process, the researcher explained the role and use of the command blocks.

The intervention phase consisted of eight sessions and was where the core intervention of this study took place. This phase mainly focused on understanding the sequential structure of programming. In addition, during the intervention process, the researcher provided appropriate feedback to facilitate the participants' learning and guided them to reinforce the problem-solving strategies that emerged during the performance process.

Table 3. **Inter-rater Reliability for Computational Thinking and Social skills**

Assessment	Kappa Value (κ)	Range (Min–Max)
Computational Thinking	0.85	0.78-0.92
Social Skills	0.80	0.74-0.88

The final maintenance phase began one week after the intervention ended and lasted a total of two weeks. In this phase, the researcher did not provide additional interventions to the participants, and observations were conducted under the same conditions as the baseline.

Intervention Fidelity Measurement

This study assessed intervention fidelity to ensure that the intervention program was implemented consistently according to the planned procedures. The evaluation was conducted by a disability welfare specialist with over five years of experience, using video recordings and session-specific instructional plans as reference materials.

Each item was rated as "Yes (1 point)" or "No (0 points)," and fidelity was evaluated for three randomly selected sessions, accounting for 30% of the total eight sessions. The results showed a high level of intervention fidelity, with an average score of 96.7% (range: 90-100%).

Inter-rater Reliability

To ensure the reliability of computational thinking and social skills assessments, which serve as the dependent variables in this study, inter-rater reliability (IRR) was calculated separately for each measure. The primary researcher served as Rater 1, while Rater 2 was a disability welfare specialist with over six years of experience. Both raters independently assessed all sessions before comparing their evaluations.

Prior to the observational assessments, the researcher conducted a pre-training session to align both raters on the study objectives, intervention methods, and assessment criteria for both computational thinking and social skills. During this training, both raters reviewed recorded video footage and refined their evaluation criteria to ensure consistency. The training concluded once the percentage agreement exceeded 90%.

Subsequently, three sessions (20% of the total intervention sessions) were randomly selected for inter-rater reliability assessment. Since the evaluation methods for computational thinking and social skills differed, their inter-rater reliability was analyzed separately. Cohen's Kappa (κ) coefficient was used for reliability analysis, and the detailed inter-rater reliability values for each assessment are presented in Table 3. Both results indicate substantial agreement, with computational thinking assessments showing slightly higher reliability (Landis et al., 1977).

Social Validity

This study assessed the social validity of the intervention program to evaluate its educational usefulness and acceptability. Social validity was measured based on goal appropriateness, procedural acceptability, and outcome usefulness (Kazdin, 1977). The evaluation consisted of a 10-item survey using a 5-point Likert scale, administered to the disability welfare specialist, study participants, and their caregivers.

The survey results indicated an average score of 4.8 out of 5, with participants and caregivers responding that the program was practically beneficial. These findings suggest that the unplugged programming education program developed in this study is both educationally valuable and highly feasible for individuals with developmental disabilities.

RESULTS

Computational Thinking Improvement

The mean scores and ranges for each phase of the study participants' computational thinking scores are presented in Table 4.

Table 4 presents the mean computational thinking scores of participants A through F across each experimental phase. All participants showed noticeable increases in their scores from the baseline to the intervention phase. The baseline mean scores ranged from 1.85 to 2.5, while the intervention phase scores increased to a range of 2.61 to 3.85. In the maintenance phase, Participants A through D maintained high levels of performance, indicating strong retention of learning. Participants E and F also showed continued growth, though to a lesser extent. Additionally, all participants achieved a PND (Percentage of Non-overlapping Data) of 100%, demonstrating a highly effective intervention across individuals. The trends in computational thinking scores for participants A through F are illustrated in Figure 1.

Table 4. **Participants' computational thinking scores**

Participants	A	B	C	D	E	F
Baseline M(SD)	2.5 (2.5-2.5)	2.45 (2.4-2.5)	2.35 (2.3-2.4)	2 (2-2)	1.85 (1.8-1.9)	2.25 (2.2-2.3)
Training M(SD)	2.9 (2.8-3.0)	3.1 (2.9-3.3)	3.55 (3.5-3.6)	2.35 (2.3-2.4)	2 (2-2)	2.35 (2.3-2.4)
Intervention M(SD)	3.49 (3.2-3.8)	3.56 (3.1-4)	3.85 (3.5-4)	3.12 (2.3-3.9)	2.61 (2-3.3)	3.22 (2.6-3.8)
Maintenance M(SD)	3.7 (3.7-3.7)	3.9 (3.8-4)	4 (4-4)	3.85 (3.7-4)	2.8 (2.8-2.8)	3.5 (3.3-3.7)
Intervention PND	100%	100%	100%	100%	100%	100%
Maintenance PND	100%	100%	100%	100%	100%	100%

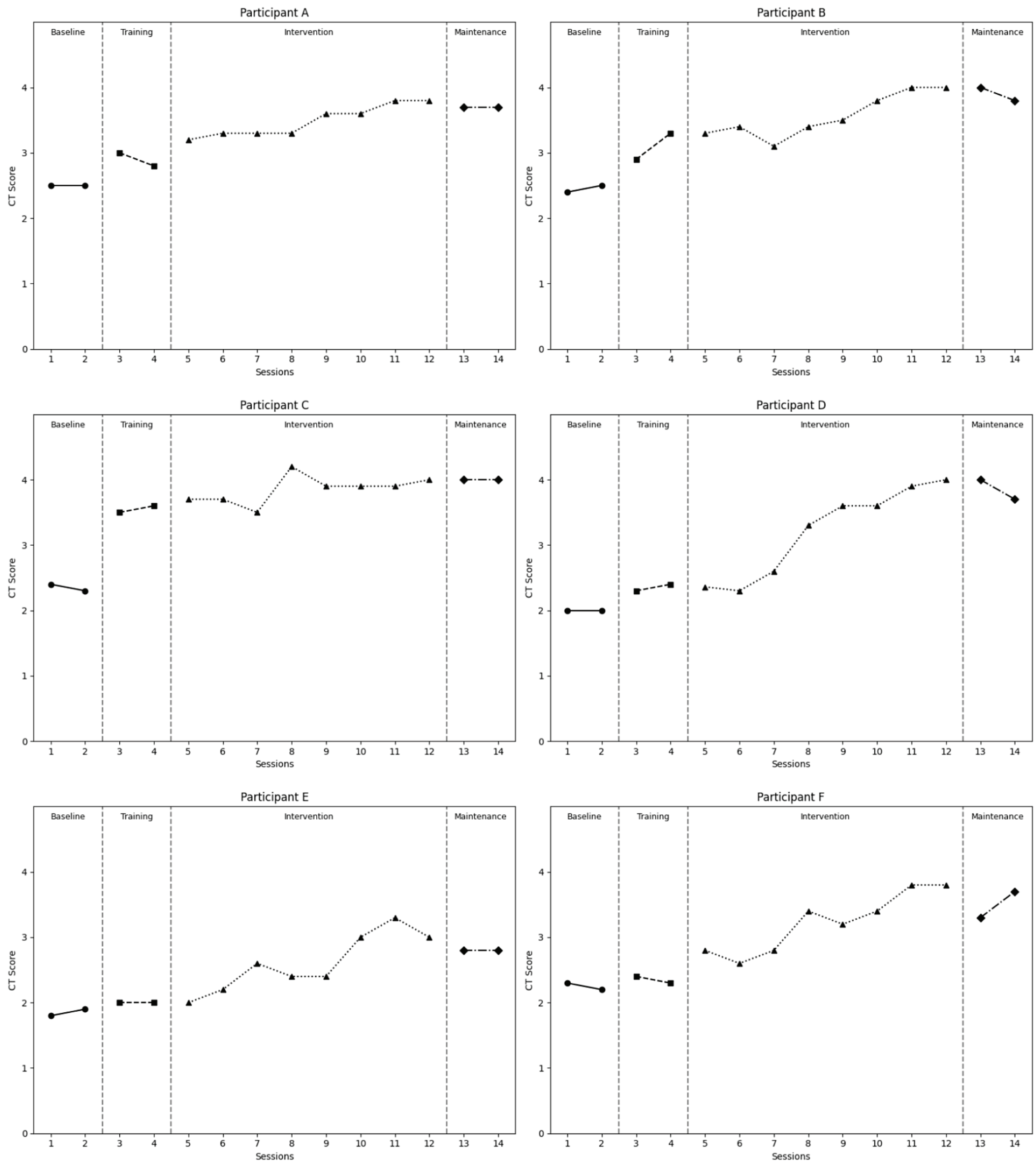

Figure 1. Changes in computational thinking scores for Participants A to F across all phases

Table 5. **Participants' Social Skills Frequencies**

Participants	A	B	C	D	E	F
Baseline M(SD)	4 (3-5)	1.5 (1-2)	3 (3-3)	10 (10-10)	8.5 (7-10)	7 (6-8)
Training M(SD)	5 (5-5)	2 (2-2)	4.5 (4-5)	14 (13-15)	12.5 (12-13)	8 (8-8)
Intervention M(SD)	7.75 (3-10)	7.75 (2-12)	7.25 (4-10)	21.86 (18-25)	19.25 (16-25)	13.63 (10-16)
Maintenance M(SD)	5.5 (5-6)	7 (6-8)	5.5 (5-6)	19 (18-20)	18.5 (15-20)	16 (16-16)
Intervention PND	75%	75%	100%	100%	100%	100%
Maintenance PND	50%	100%	100%	100%	100%	100%

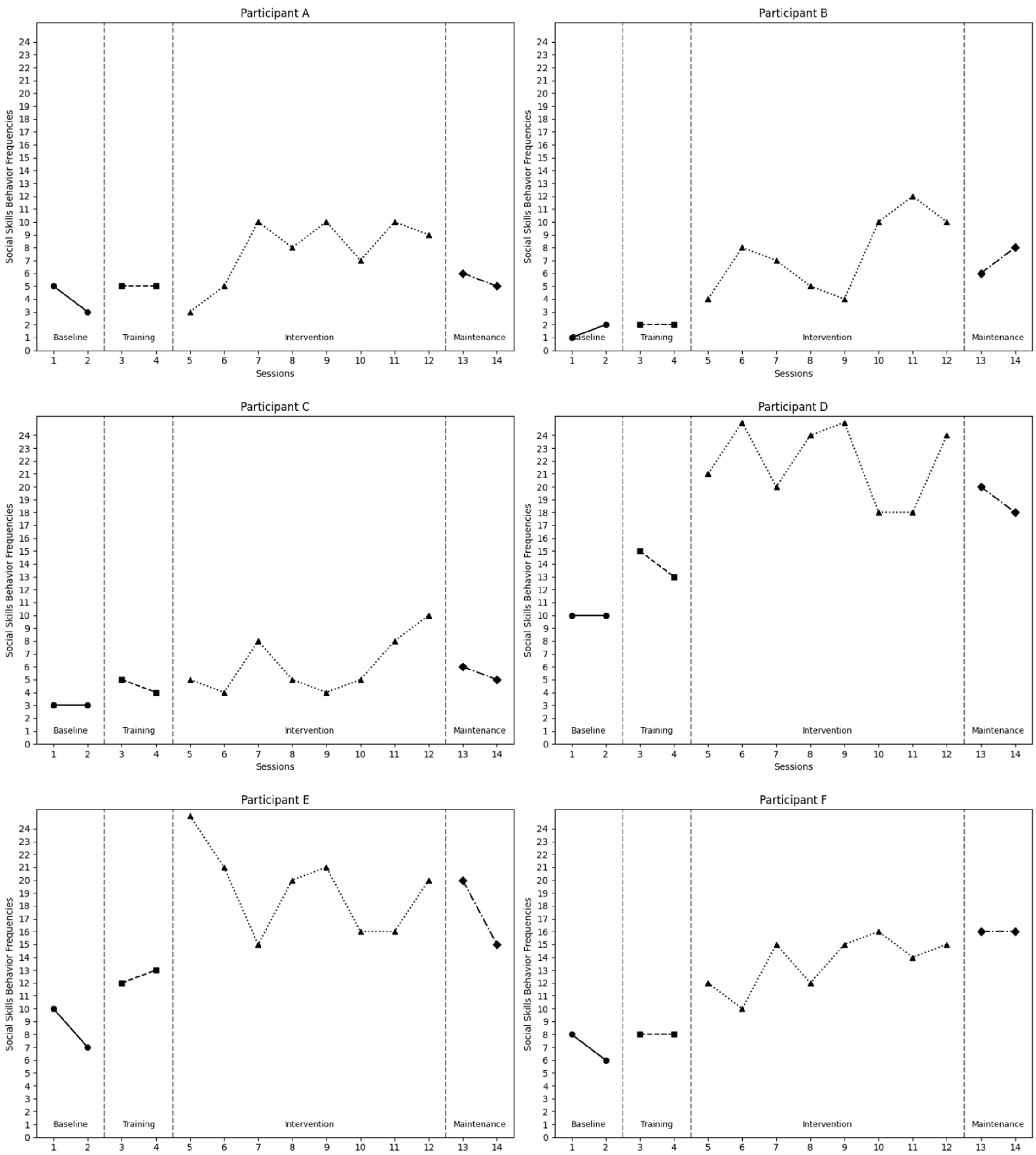

Figure 2. Changes in the frequency of social skills for Participants A to F across all phases

Participants A through D exhibited steady growth throughout the intervention, maintaining or slightly increasing their scores during the maintenance phase. Participant C, in particular, reached and sustained the maximum score, indicating the strongest response to the intervention. In contrast, Participants E and F, who began with lower baseline scores, showed gradual yet meaningful improvement over time. The visual patterns shown in the figure highlight not only the overall effectiveness of the unplugged programming intervention but also its potential to support learners with varying initial levels of computational thinking. (Figure 1.)

Social Skills Improvement

The results of social skills by phase for each participant are presented in Table 5.

All participants demonstrated increased social skills during the intervention phase compared to the baseline. Notably, participants D, E, and F exhibited particularly high frequencies, while A, B, and C showed more modest gains. The maintenance phase data indicated that most participants retained the improvements to some degree. PND analysis further confirmed the effectiveness of the intervention, with PND values reaching 100% for all participants except A and B, who showed partial maintenance. These results support the overall effectiveness and generalizability of the intervention.

Figure 2 illustrates the progression of social skills frequencies for each participant across the four phases. Participants D, E, and F started with moderate to high baseline frequencies and demonstrated strong and consistent improvement throughout the intervention and maintenance phases. In contrast, Participants A, B, and C, all diagnosed with ASD, began with lower baseline frequencies and showed more gradual improvement. However, even among these participants, behavioral gains were observed and sustained mainly during the maintenance phase. These patterns suggest that while response levels varied depending on individual characteristics, the intervention program positively influenced the social interaction skills of all participants. (Figure 2.)

CONCLUSION

In this paper, we developed and applied an unplugged programming education program for six individuals with developmental disabilities. The experiment results show that all participants have improved in both computational thinking and social skills compared to the baseline phase. Based on these findings, the conclusions and suggestions are as follows.

First, the intervention program implemented in this study was effective in enhancing the computational thinking of individuals with developmental disabilities. During the baseline phase, participants had difficulty understanding how to give commands to a robot car using coding blocks. However, as the program progressed, noticeable changes were observed. With repeated interventions, participants demonstrated the ability to select and sequence coding blocks to solve problems, going beyond merely following given instructions. They were able to identify and correct errors and explore more efficient solutions. This suggests that individuals with developmental disabilities can develop computational thinking through appropriate educational support.

Second, the unplugged programming education conducted in this study had a positive effect on improving participants' social skills. The unplugged activities naturally encouraged interaction with others. As the program progressed, participants showed increased interest in their peers, engaged in conversations, and helped one another. These findings indicate that unplugged programming education can serve as a meaningful educational tool to promote social interaction in individuals with developmental disabilities.

Third, the unplugged programming tool used in this study contributed to enhancing accessibility to programming education for individuals with developmental disabilities. By using tangible coding blocks instead of digital devices, participants were able to understand and apply programming concepts more intuitively. This suggests that instructional methods tailored to the cognitive and sensory characteristics of individuals with developmental disabilities can be effectively applied in programming education. In practical classroom contexts, teachers may further adapt unplugged programming activities to different ability levels by simplifying task steps, providing visual supports, or offering extension challenges for more advanced learners. Pairing students with diverse strengths may also foster peer learning and social interaction, ensuring that all learners can participate meaningfully.

This study was conducted as a single-subject research with only six participants, limited to a specific region and age group. This limitation is consistent with the nature of single-case research widely used in special education (Kert et al. 2022; Knight et al. 2019). However, there are limitations in generalizing the findings. In addition, only limited demographic information was collected for the participants because of ethical concerns and the sensitivity of such information. More detailed descriptors, such as cognitive level, communication ability, or prior social skills experience, could have provided more profound insights into individual differences in program outcomes. Future studies should include a broader range of age groups and disability characteristics, along with the development of educational programming that incorporates various tools and instructional strategies. Follow-up research is also needed to verify the long-term effects of such programs.

Furthermore, collecting more comprehensive participant characteristics, where ethically and practically feasible, would allow for a richer interpretation of findings. Future studies should also examine potential differential effects between individuals with ASD and ID to provide a more nuanced understanding of intervention outcomes. In addition, it would be meaningful to explore how unplugged programming can be systematically integrated with digital programming tools to scaffold progression and support the transfer of learning to broader technological contexts.

Currently, programming education for individuals with developmental disabilities is relatively scarce compared to general education curricula. However, as demonstrated in this study, with appropriate teaching strategies and tools, individuals with developmental disabilities can gradually improve both computational thinking and social skills. Accordingly, this study is meaningful as it empirically demonstrates the potential of customized programming education for individuals with developmental disabilities. It is hoped that future efforts will focus on the systematic development and dissemination of such education, along with supportive educational policies.

ACKNOWLEDGEMENT

None

DECLARATION OF INTEREST STATEMENT

The authors reported no potential conflict of interest.

ETHICAL STATEMENT

This study was approved by the Institutional Review Board of Seoul National University of Education (IRB No. SNUE-IRB-2024-002) on March 24, 2024. All participants and their legal guardians provided written informed consent prior to participation. The study involved individuals with developmental disabilities, and all procedures were conducted with careful consideration of their cognitive and sensory characteristics. Participation was entirely voluntary, and participants were informed that they could withdraw at any time without any negative consequences. All data were anonymized to ensure confidentiality. The study was carried out in accordance with the ethical standards of the institution and the 1964 Helsinki Declaration and its later amendments or comparable ethical standards.

FUNDING

None

REFERENCES

Araujo, E. C. J. de, Andrade, W. L., & Oliveira, A. L. S. (2022). Identifying programming skills impacted in students with cognitive disabilities. *2022 IEEE Frontiers in Education Conference (FIE)*, 1–9. https://doi.org/10.1109/FIE56618.2022.9962703

Bargagna, S., Castro, E., Cecchi, F., Cioni, G., Dario, P., Dell'Omo, M., Di Lieto, M. C., Inguaggiato, E., Martinelli, A., Pecini, C., & Sgandurra, G. (2019). Educational robotics in Down syndrome: A feasibility study. *Technology, Knowledge and Learning, 24*(2), 315–323. https://doi.org/10.1007/s10758-018-9366-z

Bouck, E. C., & Yadav, A. (2022). Providing access and opportunity for computational thinking and computer science to support mathematics for students with disabilities. *Journal of Special Education Technology, 37*(1), 151–160. https://doi.org/10.1177/0162643420978564

Brennan, K., & Resnick, M. (2012, April). *Using artifact-based interviews to study the development of computational thinking in interactive media design*. Annual Meeting of the American Educational Research Association (AERA), Vancouver, BC, Canada.

Demir, Ü. (2021). The effect of unplugged coding education for special education students on problem-solving skills. *International Journal of Computer Science Education in Schools, 4*(3), 1–30. https://doi.org/10.21585/ijcses.v4i3.95

Eiselt, K., & Carter, P. (2018). Integrating social skills practice with computer programming for students on the autism spectrum. *2018 IEEE Frontiers in Education Conference (FIE)*, 1–5. https://doi.org/10.1109/FIE.2018.8659232

Elshahawy, M., Bakhaty, M., & Sharaf, N. (2020). Developing computational thinking for children with autism using a serious game. *2020 24th International Conference Information Visualisation (IV)*, 761–768. https://doi.org/10.1109/IV51561.2020.00135

Gkiolnta, E., Zygopoulou, M., & Syriopoulou-Delli, C. K. (2023). Robot programming for a child with autism spectrum disorder: A pilot study. *International Journal of Developmental Disabilities, 69*(3), 424–434. https://doi.org/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., & Infante-Moro, A. (2019). Computational thinking and Down syndrome: An exploratory study using the KIBO robot. *Informatics, 6*(2), 25. https://doi.org/10.3390/informatics6020025

Gresham, F. M., & Elliott, S. N. (1990). *Social skills rating system manual*. Circle Pines, MN: American Guidance Service.

Gribble, J., Hansen, A., Harlow, D., & Franklin, D. (2017). Cracking the code: The impact of computer coding on the interactions of a child with autism. In *Proceedings of the 2017 ACM Conference on Interaction Design and Children (IDC '17)* (pp. 445–449). ACM. https://doi.org/10.1145/3078072.3084307

Hwang, J., & Taylor, J. C. (2016). Stemming on STEM: A STEM education framework for students with disabilities. *Journal of Science Education for Students with Disabilities, 19*(1), 4. https://scholarworks.rit.edu/jsesd/vol19/iss1/4

Karaman, G., & Seferoğlu, S. S. (2024). Identification of performance, motivation, and support needs in coding education provided for the students with mild intellectual disabilities. *Educational Academic Research, 55*, 82–92. https://doi.org/10.33418/education.1487199

Kazdin, A. E. (1977). Assessing the clinical or applied importance of behavior change through social validation. *Behavior Modification, 1*(4), 427–452. https://doi.org/10.1177/014544557714001

Kert, S. B., Yeni, S., & Erkoç, M. F. (2022). Enhancing computational thinking skills of students with disabilities. *Instructional Science, 50*, 625–651. https://doi.org/10.1007/s11251-022-09585-6

Kim, M., Kim, J., & Lee, W. (2025). Intellectual disabilities and programming: Improving computational thinking based problem solving. *Education and Information Technologies, 30*, 12101–12141. https://doi.org/10.1007/s10639-024-13253-2

Kim, M., Kim, J., & Lee, W. (2024). Enhancing computational thinking in students with autism spectrum disorder and intellectual disabilities: A robot programming approach. *International Journal of Developmental Disabilities,* 1-16. https://doi.org/10.1080/20473869.2024.2394735

Knight, V. F., Wright, J., & DeFreese, A. (2019). Teaching robotics coding to a student with ASD and severe problem behavior. *Journal of Autism and Developmental Disorders, 49*, 2632–2636. https://doi.org/10.1007/s10803-019-03888-3

Knight, V. F., Wright, J., Wilson, K., & Hooper, A. (2019). Teaching digital, block-based coding of robots to high school students with autism spectrum disorder and challenging behavior. *Journal of Autism and Developmental Disorders, 49*, 3113–3126. https://doi.org/10.1007/s10803-019-04033-w

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics, 33*(1), 159–174.

Lee, S. M., Chun, S. J., Jo, Y., Hong, J. Y., & Seo, J. H. (2024). Tangible programming education program to improve collaborative problem-solving (CPS) competency of elementary school students. In J.-C. Hong (Ed.), *New technology in education and training* (pp. 391–408). Springer. https://doi.org/10.1007/978-981-97-3883-0_22

Lindsay, S., & Lamb, A. (2018). Exploring types of play in an adapted robotics program for children with disabilities. *Disability and Rehabilitation: Assistive Technology, 13*(3), 263–270. https://doi.org/10.1080/17483107.2017.1306595

Michalek, A. M. P., Phalen, L., Bobzien, J. L., Chen, C.-H., Urbano, M., Hartmann, K., Okwara, L., Gorrepati, P., Deutsch, S., & Williams, T. (2020). Using a STEM activity to improve social communication interactions in autism spectrum disorder. *Preventing School Failure: Alternative Education for Children and Youth, 65*(1), 38–47. https://doi.org/10.1080/1045988X.2020.1811627

Muñoz, R., Villarroel, R., Barcelos, T. S., Riquelme, F., Quezada, Á., & Bustos-Valenzuela, P. (2018). Developing computational thinking skills in adolescents with autism spectrum disorder through digital game programming. *IEEE Access, 6*, 63880–63894. https://doi.org/10.1109/ACCESS.2018.2877417

Sola-Özgüç, C., & Altın, D. (2022). Teaching block-based coding to a student with autism spectrum disorder. *Ankara University Faculty of Educational Sciences Journal of Special Education, 23*(3), 565–594. https://doi.org/10.21565/ozelegitimdergisi.822554

Papert, S. (2005). Teaching children thinking. *Contemporary Issues in Technology and Teacher Education, 5(3/4)*, 353–365.

Papert, S. (1980). *Mindstorms: Computers, children, and powerful ideas*. Basic Books.

Ratcliff, C. C., & Anderson, S. E. (2011). Reviving the turtle: Exploring the use of Logo with students with mild disabilities. *Computers in the Schools, 28*(3), 241–255. https://doi.org/10.1080/07380569.2011.594987

Taylor, M. S. (2018). Computer programming with pre-K through first-grade students with intellectual disabilities. *The Journal of Special Education, 52*(2), 78–88. https://doi.org/10.1177/0022466918761120

Taylor, M. S., Vasquez, E., & Donehower, C. (2017). Computer programming with early elementary students with Down syndrome. *Journal of Special Education Technology, 32*(3), 149–159. https://doi.org/10.1177/0162643417704439

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wright, J. C., Knight, V. F., Barton, E. E., & Edwards-Bowyer, M. (2021). Video prompting to teach robotics and coding to middle school students with autism spectrum disorder. *Journal of Special Education Technology, 36*(4), 187–201. https://doi.org/10.1177/0162643419890249

Yuen, T. T., Mason, L. L., & Gomez, A. (2014). Collaborative robotics projects for adolescents with autism spectrum disorders. *Journal of Special Education Technology, 29*(1), 51–62. https://doi.org/10.1177/016264341402900104