Student Centered Computational Thinking for Children with Disabilities

Abstract
Student-centered pedagogies such as Universal Design for Learning and Coding as Another Language may provide opportunities for students with disabilities to access high-quality computational thinking education, but the outcomes of such curricula have not yet been assessed. In this paper, students with and without disabilities were assessed on CT knowledge before and after a CT curriculum based on the Coding as Another Language Framework. Students with disabilities had lower CT scores, but there was no interaction effect between pre-CT score and disability on post-CT score suggesting that the trajectory of CT learning was equal for students with and without disabilities. Model fit appeared to vary by students' specific disability, suggesting that CT learning from the curricula may vary by disability.

Objectives

Twenty-first century skills including engineering, coding, and computational thinking are becoming increasingly important for students to learn. The Computer Science for All initiative, the development of the K-12 Computer Science Framework, and Code.org's national hour of code all highlight the importance of these learning domains in state and national conversations. However, students with disabilities are not equally accessing these educational programs as compared to their nondisabled peers leading to knowledge gaps in computer science and computational thinking for students with disabilities. Nearly half of students with disabilities scored below proficient on the National Assessment of Educational Progress Technology and Engineering Literacy content area, compared to approximately 10% of students without disabilities (National Center for Education Statistics, 2021). This inequity in education leads to underrepresentation in the STEM field, where only 5% of STEM doctorate holders under 40 have a disability compared to 10% of the general working population (National Center for Science and Engineering Statistics, 2021). Currently, there is a lack of research on how students with disabilities learn CT and how to best support students with disabilities in CT learning. This paper examined how first and second grade students with disabilities with Individualized Education Plans (IEPs) responded to the Coding as Another Language Curriculum, an early-childhood curriculum for coding and computational thinking in the general education classroom. The paper has implications for the development of best practices for providing CT education to students with disabilities and for the development of future CT programs that can equitably reach these students.

Theoretical Framework

Special Education and Computational Thinking

Many of the previously developed and researched computer science and coding programs for students with disabilities have, like many special education curricula, been grounded in behaviorist practices and pedagogies. These pedagogies include explicit instruction, which has been used to teach coding and CT to students with Down syndrome, autism, and intellectual disability (Pivetti et al., 2020). However, these programs do not emphasize the expressive language aspect of coding languages, and skills taught through instructionist curricula do not necessarily generalize to new settings or programs. For example, one study using explicit instruction successfully taught preschool, kindergarten, and first grade students with intellectual disabilities to code using the Dash robot, but no student was able to apply the coding skills learned to complete a novel coding challenge (Taylor, 2018).

Universal Design for Learning

Universal Design for Learning (UDL) is an educational framework that incorporates accommodations within the curriculum by providing multiple means of accessing and exploring the material. This framework has been used to develop and plan multiple curricula and is encouraged by disability activists as a supportive and accessible method of accommodating and including students with a diverse range of disabilities in their classroom communities (Capp, 2017). Prior work has suggested that incorporating UDL practices in coding and computer science programs can help students with disabilities access computer science education programs (Israel et al., 2015, 2020). However, studies have not yet examined the CT outcomes for students with and without disabilities of curricula with UDL features.

Coding as Another Language

The Coding as Another Language framework is a student-driven pedagogy that promotes coding as a form of expression (Bers, 2019, 2020). According to this framework, coding languages, like written languages, are tools for students to tell stories, create art, and communicate ideas. Students become fluent in the language of code through use and practice, and the role of the teacher is to provide a scaffolded environment in which children can express themselves in purposeful ways. As students become more knowledgeable about the language of code, they become more fluent and can use the language for more abstract projects and communications.

The Coding as Another Language curriculum is an early-elementary coding and robotics curriculum designed for a general education setting and includes many elements associated with the UDL pedagogy, for example providing for multiple means of engagement by including unplugged CS activities, songs, and dances alongside teacher modeling of new skills and time spent in coding exploration (Relkin & Bers, 2020). The Coding as Another Language curriculum's emphasis on personally motivating projects including both artistic and code elements also offers multiple means of expression to students.

This paper examined whether disability status affects students' CT learning over the course of the CAL curriculum. We hypothesized that although students with disabilities would have lower CT scores compared to their nondisabled peers, there would not be a significant interaction effect of disability and pre-curriculum CT, suggesting a similar model of learning for students with and without disabilities.

Method

Participants

Participants were first and second grade students in public school classrooms enrolled in a research study evaluating the CAL curriculum (Relkin & Bers, 2020). Although the research study included both students receiving the curriculum and students in a control setting, this analysis only examined the 779 students (267 in first grade, 512 in second grade) in the experimental condition who completed the curriculum including all related assessments. Of these students, the school district identified 76 as having a disability. There was an even gender distribution, and sample was also racially diverse (Table 1).

Measures

Computational Thinking

CT was evaluated using the TechCheck assessment (Relkin et al., 2020). The assessment consists of 15 multiple choice questions each worth one point. Scores were summed for each student at each timepoint. CT was assessed both before and after completing the curriculum.

Disability Status

Students were identified as having a disability if they had an IEP during the year they received the curriculum. Disability status was provided by the school district. Students without IEPs were categorized as not having a disability. We combined disability categories for analysis due to differences in diagnostic classification for first and second grade students and due to the relatively low proportion of students with disabilities compared to the overall student population.

Procedure

As part of the experimental condition for the CAL study, first and second grade classrooms from eight elementary schools in a Mid-Atlantic school district were given the CAL curriculum developed by the DevTech research group at Tufts University. The CAL curriculum consisted of 12 to 15 1-hour lessons centered around storybooks and KIBO, a developmentally appropriate coding and tangible robotics tool.

CT was evaluated before and after curriculum implementation. The public school district supplied demographic data and disability status for all students participating in the study.

Analysis

We estimated a linear regression with an interaction term to evaluate if CT knowledge prior to completing the curriculum predicted CT knowledge after completing the curriculum, and if disability status moderated this relationship. All analyses were done in R using the Tidyverse package (R Core Team, 2020; Wickham et al., 2019).

Results

Preliminary Analyses

We used t-tests and Pearson correlations to examine the relationship between pre- and post-curriculum CT. Post-curriculum CT was significantly higher than pre-curriculum CT ($M_{pre} = 10.17$, $M_{post} = 11.11$, $t(1353.6) = -6.84$, $p < 0.001$). Pre-curriculum CT knowledge was positively correlated with post-curriculum CT knowledge ($r = 0.57$, $p < .001$). We used t-tests to determine if students with and without disabilities performed differently on the TechCheck assessment. Across both timepoints, students without disabilities showed significantly more CT knowledge than students with disabilities (pre-curriculum: $t(91.76) = -2.6455$, $p = .001$; post-curriculum: $t(79.52) = -2.8866$, $p = .005$).

Computational Thinking

Our research question was whether students with disabilities showed different CT learning than students without disabilities. We estimated an ordinary least squares regression examining whether pre-curriculum CT, student disability status, and their interaction predicted post-curriculum CT. Approximately 30.9% percent of variance in post-curriculum CT, a statistically significant amount, was predicted by pre-curriculum CT, disability status, and the interaction of these variables, $F(3,775) = 115.6$, $p < .001$. The interaction of pre-curriculum CT and disability status was not a statistically significant predictor of post-curriculum CT suggesting that CT learning did not statistically differ for students with disabilities and students without disabilities, $\beta = 0.05$, $p = .603$.

As there was not a significant interaction, we estimated an ordinary least squares regression examining whether pre-curriculum CT and student disability status predicted post-curriculum CT (Figure 1). As the interaction did not significantly predict post-curriculum CT, there was still 30.9% percent of variance in post-curriculum CT predicted by pre-curriculum CT and disability status, $F(2,776) = 173.4$, $p < .001$. Both pre-curriculum CT and disability status significantly predicted later CT knowledge. For each one-point difference in pre-curriculum CT, the model predicted a 0.54-point higher CT score after the curriculum, holding disability status constant, $p < .001$. A student with a disability had a 0.78 predicted lower post-curriculum CT score than a student without a disability, holding pre-curriculum CT constant, $p = .003$.

Disability-Specific Model Fit

Although there was not enough statistical power to separately examine the effects of different categories of disability, we used a scatterplot to examine how the residuals for students with different disabilities compared to each other and to students without disabilities (Figure 2). As seen in Figure 2, some disabilities may have different effects on CT learning than other disabilities. Although the residuals of students with some disabilities, such as specific learning disability (SLD), appeared to be evenly distributed, the residuals of students with other disabilities appeared more clustered. For example, students with developmental delays (DD) appeared to have lower residuals and lower CT scores, whereas students with specific language impairments (SLI) appeared to have higher residuals. This suggests that the model of CT learning fit less well for students with these disabilities.

Discussion

This paper examined the relationship between pre-curriculum CT knowledge, student disability status, and post-curriculum CT knowledge in order to determine whether CT outcomes from the CAL curriculum for students with disabilities were equal to those for students without disabilities. For all students, higher CT knowledge before beginning the curriculum was associated with higher CT knowledge at the end of the curriculum. Students with disabilities had lower CT scores than students without disabilities, but there was no effect of an interaction between pre-curriculum CT and disability on post-curriculum CT, suggesting that the trajectory of CT learning was equal for students with and without disabilities.

In other words, this suggests that although students with disabilities scored lower on the assessments overall, they made similar gains in knowledge to their peers without disabilities. As this curriculum was not an intervention for students with disabilities, we would not expect to see students with disabilities learn more than their peers without disabilities. However, this curriculum was developed for and implemented in a general education classroom, making it of note that students with disabilities learned an equivalent amount to their peers without disabilities, a finding we attribute to the UDL aspects incorporated in to the intervention curriculum.

The Coding as Another Language pedagogical framework is in line with many UDL pedagogical practices, including providing multiple means of engagement and expression (Bers, 2019, 2020; Israel et al., 2020). Our finding that disability did not impact CT learning trajectory suggests that a computer science curriculum that incorporates these practices can lead to successful CT outcomes for young students with disabilities, even when in the general education classroom. This possibility offers implications for future developers of best practices and computational thinking curricula for students with disabilities and can lead the way to both more equitable CS education and a more equal STEM workforce.

References

Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, *6*(4), 499–528. https://doi.org/10.1007/s40692-019-00147-3

Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.

Capp, M. J. (2017). The effectiveness of universal design for learning: A meta-analysis of literature between 2013 and 2016. *International Journal of Inclusive Education*, *21*(8), 791–807. https://doi.org/10.1080/13603116.2017.1325074

Israel, M., Jeong, G., Ray, M., & Lash, T. (2020). Teaching Elementary Computer Science through Universal Design for Learning. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 1220–1226. https://doi.org/10.1145/3328778.3366823

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

National Center for Education Statistics. (2021). *Digest of Education Statistics: 2019*. U.S. Department of Education. https://nces.ed.gov/programs/digest/d19/

National Center for Science and Engineering Statistics. (2021). *Women, Minorities, and Persons with Disabilities in Science and Engineering: 2021* (Special Report NSF 21-321). National Science Foundation. https://ncses.nsf.gov/wmpd

R Core Team. (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Relkin, E., & Bers, M. (2020). *Exploring the Relationship Between Coding, Computational Thinking and Problem Solving in Early Elementary School Students.* Annual Meeting of the American Educational Research Association (AERA), San Francisco, CA.

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology*, *29*(4), 482–498. https://doi.org/10.1007/s10956-020-09831-x

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1683. https://doi.org/10.21105/joss.01686

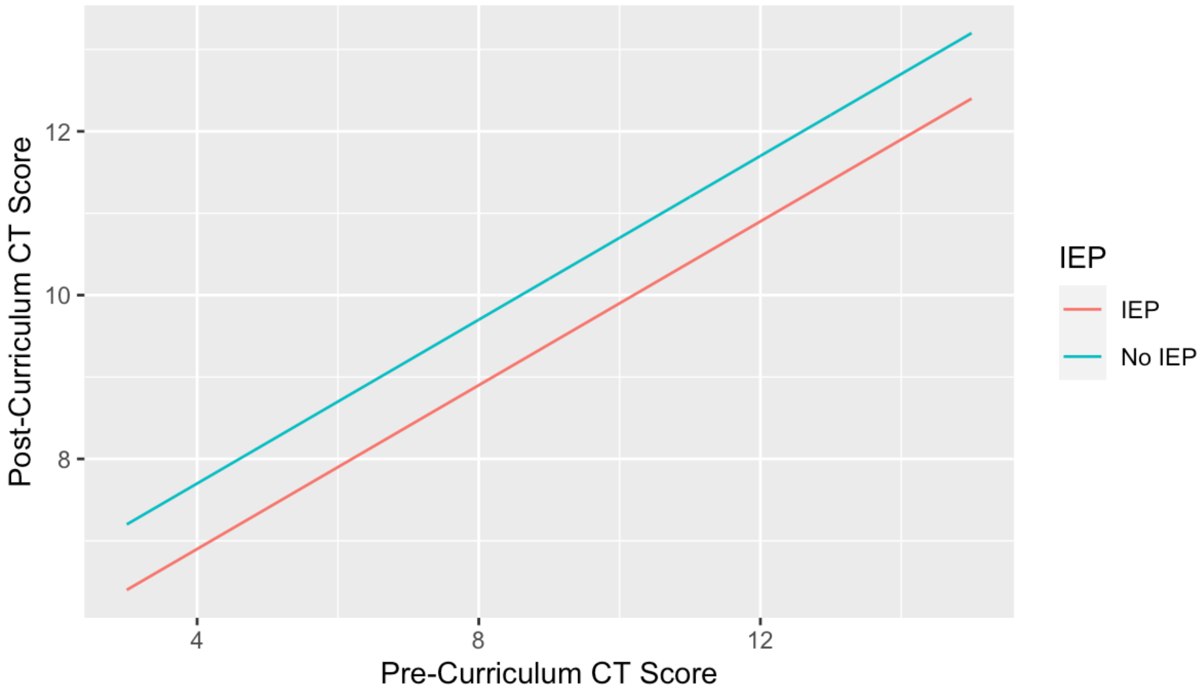
Table 1

Demographic Characteristics for Students Participating in the CAL Curriculum

Baseline Characteristic	Has IEP		No IEP		Full Sample	
	n	%	*n*	%	*n*	%
Full Sample	72	0.09	707	0.91	779	1.00
Gender						
Female	23	0.03	389	0.50	412	0.53
Male	49	0.06	318	0.41	367	0.47
Racial demographics						
White Non-Hispanic	28	0.04	226	0.29	254	0.33
African-American/Black	30	0.04	314	0.40	344	0.44
Hispanic/Latino	9	0.01	80	0.10	89	0.11
Asian	1	0.00	24	0.03	25	0.03
Other	4	0.01	63	0.08	67	0.09

Figure 1

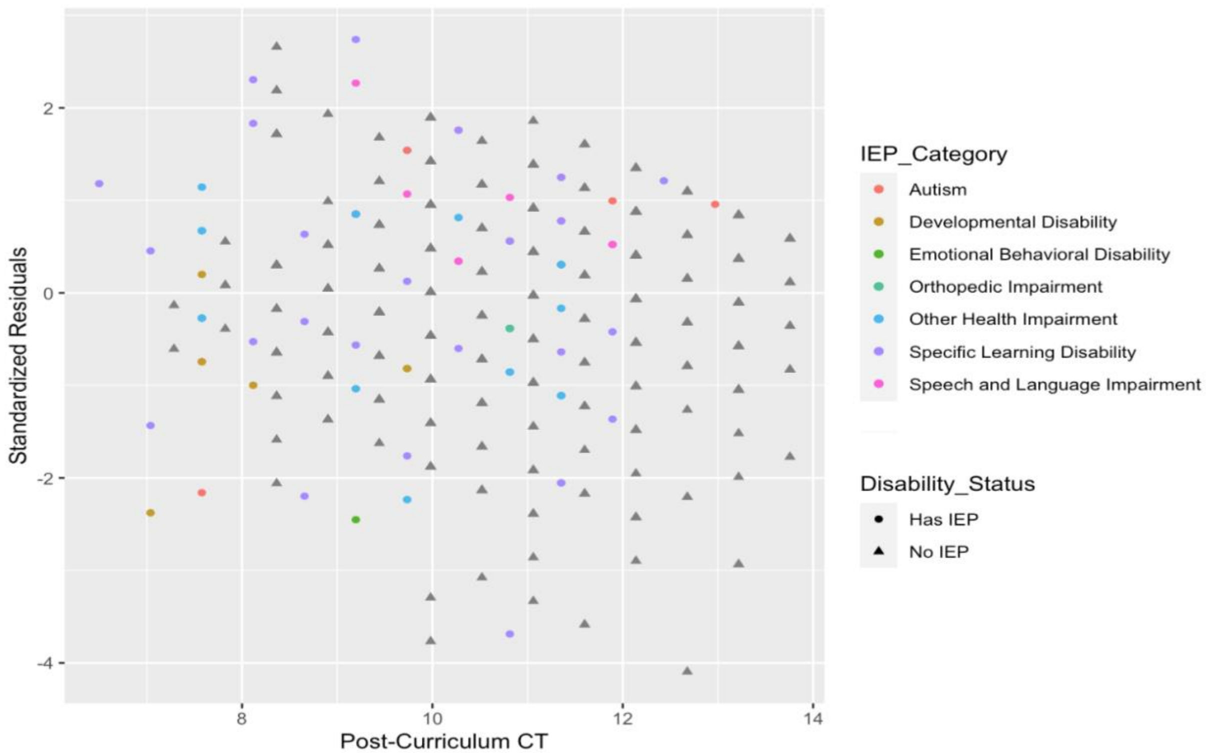
Pre- and Post-Curriculum Computational Thinking (CT) by Disability Status



Note. Computational Thinking (CT) was defined as students' scores on the TechCheck assessment. Among both students with and without disabilities, higher pre-curriculum computational thinking (red line) was associated with higher post-curriculum computational thinking (blue line). There was no interaction between pre-curriculum computational thinking and disability status.

Figure 2

Residuals for Predicted Post-Curriculum Computational Thinking (CT) by IEP Category



Note. Scores for post-curriculum computational thinking were defined as students' scores on the TechCheck assessment following implementation of the curriculum. Although each category did not have the required power to run analysis models by IEP category, the distribution of residuals for IEP categories appears uneven, suggesting future analyses should account for differences between students with different IEP categories.