Coding as Another Language: Impact on Math and Literacy Achievement in Early CS Education

Abstract

This study examines the impact of an early computer science curriculum, Coding as Another Language and the growth of coding skills on students' math and literacy achievement. Results from a randomized controlled trial with 680 K-2 students showed that implementing the CAL curriculum did not yield a significant impact on literacy and math achievements, but growth in coding skills positively predicted both outcomes.These results allay concerns regarding potential trade-offs between computer science instruction and core subjects, highlighting the positive influence of coding skill development on math and literacy achievement. The implications of these findings are important for educators, policymakers, and researchers as they deliberate the integration of computer science education into the curriculum for young learners.

Introduction

In today's increasingly digital world, learning to code has become an essential skill for children. The growing demand for computer programming skills in the job market is creating a need for a larger workforce with coding expertise. According to the U.S. Bureau of Labor Statistics (2021), the employment of computer and information technology occupations is projected to grow 11% from 2019 to 2029, much faster than the average for all occupations. This translates to an estimated 531,200 new jobs in the United States alone in this field and an estimated 1.4 million computing jobs are expected to become available in the US by 2026.

In addition to the economic benefits, learning to code also has cognitive and academic advantages. Research shows that coding instruction may improve children's problem-solving, logical reasoning, creativity (Çiftci & Bildiren, 2020; Scherer et al., 2019) and executive functioning skills such as planning and response inhibition (Arfé et al., 2019). The positive effects of coding may be also transferred to children's learning in other subjects, such as math, science and literacy (Voogt et al., 2015).

Seeing the great demand and potential of computer science (CS) education, more than 45 states have adopted K-12 computer science standards or policies, and 32 states allow computer science to count towards high school graduation requirements by 2021. These efforts aim to provide all students, regardless of their background or interests, with the opportunity to learn coding and acquire the skills needed to succeed in the digital age.

However, incorporating and implementing computer science into existing school curricula can present challenges. The limited time allocated to subjects such as language arts, math, and science can create concerns among educators and policymakers, who may worry that adding another curriculum could reduce instruction time for other subjects and result in lower academic performance (Webb et al., 2017). Yet little research has explored how the implementation of a computer science curriculum in early elementary education affects academic achievement, particularly in literacy and mathematics. Therefore, this study aims to address this gap by examining the impact of a computer science curriculum - Coding as Another Language-ScratchJr (CAL-ScratchJr) on children's standardized assessment scores in these core subjects. The following research questions guided our study design and analysis:

1. To what extent does the CAL-ScratchJr curriculum impact students' literacy and math standardized assessment post-test scores controlling for their pre-test scores and grade level?
2. To what extent does the growth in students' coding skills impact students' literacy and math standardized assessment post-test scores controlling for their pre-test scores and grade level?

Methods

The CAL-ScratchJr Curriculum

The Coding as Another Language - ScratchJr (CAL-ScratchJr) Curriculum is an evidence-based computer science curriculum designed for kindergarten to second grade students (Bers et al, 2023). The goal of the CAL-ScratchJr curriculum is not to train students to become skillful software engineers, but rather to nurture global citizens of the next generation by developing their problem-solving skills, character strength, and collaborative attitudes. It was developed under the guidance of the "Coding as another language" pedagogy, which emphasizes the relationship between coding and literacy development (Bers, 2019). To achieve this, the curriculum introduces the powerful ideas from computer science (e.g. algorithms), in conversations with the corresponding powerful ideas from literacy (e.g., alphabet) in a playful and structured way. The CAL-ScratchJr curriculum utilizes ScratchJr, a block-based programming language created specifically for young children (Flannery et al., 2013). The curriculum is composed of 24 lessons that amount to 18 hours of instruction, tailored to cater to students at varying grade levels. A visual representation of a CAL-ScratchJr lesson is illustrated in Figure 1.

Study Procedures

The authors of the current study utilized data from a larger study (Authors, year, blinded for review) that investigated the impact of the CAL-ScratchJr curriculum on coding and computational thinking skills of students in kindergarten to second grade. The study was conducted in one state on the east coast of the United States, involving a total of 12 public elementary schools with kindergarten to second-grade classrooms. These schools were randomly assigned to either the treatment group or the business-as-usual control group, with the treatment group receiving the CAL-ScratchJr curriculum in the first school year and the control group receiving it in the second school year.

The analytical sample in the current study included 747 students who completed the study and were provided with either literacy or math test scores from the schools. Students were assessed on their coding and computational thinking skills before and after the intervention. Given that the CAL-ScratchJr intervention was conducted during the winter and spring quarters in 2022, students' math and literacy assessment scores in fall 2021 (pre-intervention) and spring 2022 (post-intervention) were obtained from the schools. Teachers in the treatment group received professional development training delivered by trained research assistants before the intervention, which included two sessions totaling four hours on both the ScratchJr app and the CAL-ScratchJr curriculum.

Participants

In total, the analysis sample included 747 students - 398 from the control group and 349 students from the treatment group. Detailed student demographic information by the treatment and control groups can be found in Table 1.

Coding Stages Assessment (CSA)

The Coding Stages Assessment (CSA) is a validated test that assesses the programming skills of young children in kindergarten to lower elementary grades, using the ScratchJr programming language (de Ruiter & Bers, 2021). It comprises a total of 25 question items, organized into five stages with five questions per stage. The split-half reliability for the CSA, as determined by Guttman's Lambda 6, is .94 (de Ruiter & Bers, 2021).

Literacy and Math Standardized Assessments

Students' literacy and math performance before and after implementing the study was measured using the standardized assessments administered by the teachers and educators at their schools in fall and spring quarters. There were in total four types of standardized assessments used by the participating schools and in the current study, standardized scores (z scores) were used in the analysis. The four assessments were: Fastbridge reading and math assessments (used in three schools), STAR reading and math assessments (used in five schools), Aimsweb reading and math assessments (used in three schools), iReady Diagnostic reading and math assessment (used in one school).

Results

Descriptive statistics of students' test scores are demonstrated in Table 2. Correlation matrix of the test scores, intervention condition and grade level is presented in Table 3. Baseline equivalence was established by examining whether math and literacy pre-test scores were significantly different between the treatment and control group controlling for the grade level. Analysis results showed that neither math (β = -.11, *SE* = .14, *p* = .42) nor literacy (β = -.14, *SE* = .14, *p* = .33) pre-test scores differed significantly between the treatment group and control group.

To answer the first research question, we conducted multilevel regression analysis on math and literacy post-test scores controlling for students' math or literacy pre-test scores and their grade level. Model details are presented in Math Model 1 and Literacy Model 1 in Table 4. To investigate the extent to which children's development of coding skills predict their math and literacy outcomes, we added the differences between the post CSA and pre CSA scores. Model details are presented in Math Model 2 and Literacy Model 2 in Table 4.

Discussion

Balancing Computer Science Education and Core Subjects

As computer science education receives increased attention from federal and state governments, it becomes imperative for states and districts to establish policies that enhance the capacity for CS education within the existing K-12 system. Despite the undeniable importance of CS education, concerns have been raised by policymakers, educators, and teachers regarding the potential trade-off between instructional time allocated to CS education and academic performance in core subjects like math and literacy. However, our study's null finding regarding the impact of a CS curriculum on students' standardized math and literacy achievement suggests that implementing a CS curriculum alongside the existing curriculum did not lead to significant improvements in these outcomes, nor did it result in a decline in students' overall academic achievement. It is important to interpret these results in light of the fact that students in the treatment group received less instructional time dedicated to math and literacy compared to the control group. Despite the reduced instructional time, students in the treatment group demonstrated similar academic achievement in other key subjects.

One possible explanation for this finding is that although students in the treatment group received less time for explicit instruction in literacy and math, they received additional instruction in a computer science curriculum that integrated elements of literacy and math. This approach allowed students to engage with key concepts in literacy and math from a computer science perspective. Therefore, even though explicit instruction in literacy and math was not provided during this period, students were still exposed to and approached these subjects through the lens of computer science. As a result, although the total amount of instructional time dedicated to literacy and math was less than that of the control group, students in the treatment group achieved comparable overall academic performance.

Enhancing Math and Literacy Achievement Through Coding with ScratchJr

We also found that the growth in students' coding skills using ScratchJr positively predicted their academic achievement in mathematics and literacy. This may be attributed to the direct benefits of coding with ScratchJr, as well as the indirect benefits through cognitive development.

Firstly, the direct benefits of learning to code using ScratchJr may have contributed to the improvement of math and literacy skills for our participants in kindergarten to second grade. Coding with ScratchJr involves key math and literacy concepts such as basic operations and narrative structure. For instance, the use of parameters and loop blocks in ScratchJr naturally reinforces number counting, basic operations, and abstraction. Additionally, ScratchJr provides a platform for creating, refining, and enacting stories with multiple characters and settings, enabling children to practice vital literacy concepts like narrative structure and story writing.

Second, existing research suggests that learning to code has a positive impact on children's cognitive development, which in turn may have contributed to improved performance in various academic domains, including mathematics and literacy (Voogt et al., 2015). ScratchJr, with its distinctive characteristics of a low floor, high ceiling, and wide walls (Flannery et al., 2013), provides a conducive environment for fostering cognitive growth.

Within ScratchJr, children can create both simple and complex programs to bring their stories to life, involving single or multiple characters in various settings. This process requires them to hold information in mind while sequentially planning and executing each step of the program to enact the desired story. Engaging in trial and error, they run the program, compare it with their envisioned story, debug any issues, and continuously refine their programs until satisfaction. This iterative process actively engages and strengthens a range of cognitive skills, including working memory, attentional skills, goal setting, planning, problem-solving, computational thinking, and creativity (Chu et al., 2016; Voogt et al., 2015).

Implications

These findings have implications for policymakers and educators as they consider the integration of computer science education within the K-12 curriculum, highlighting the potential for a balanced approach that incorporates computer science instruction while maintaining the necessary instructional time for core subjects such as math and literacy. Meanwhile, the findings underscore the benefits of early computer science education, not only in terms of students' CS-related skills but also their cognitive development and overall academic achievements. The current lack of computer science programs tailored to early childhood and early elementary grades calls for action. Educators and policymakers might consider integrating early CS instruction that takes a literacy approach into the early education curriculum. This integration holds potential for fostering interdisciplinary skills and equipping students with the cognitive foundations necessary for success in mathematics, literacy, and beyond.

References

Arfé, B., Vardanega, T., Montuori, C., & Lavanga, M. (2019). Coding in primary grades boosts children’s executive functions. *Frontiers in psychology*, *10*, 2713.

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.

Bers, M. U., Blake-West, J., Kapoor , M. G., Levinson, T., Relkin, E., Unahalekhaka, A., & Yang, Z. (2023). Coding as another language: Research-based curriculum for Early Childhood Computer Science. *Early Childhood Research Quarterly*, *64*, 394-404. https://www.sciencedirect.com/science/article/pii/S0885200623000571

Bureau of Labor Statistics. (2021). Computer and Information Technology Occupations. Retrieved from https://www.bls.gov/ooh/computer-and-information-technology/home.html

Chu, F. W., VanMarle, K., & Geary, D. C. (2016). Predicting children's reading and mathematics achievement from early quantitative knowledge and domain-general cognitive abilities. *Frontiers in psychology*, *7*, 775.

Çiftci, S., & Bildiren, A. (2020). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. *Computer science education*, *30*(1), 3-21.

de Ruiter, L. E. & Bers, M. U. (2021). The Coding Stages Assessment: development and validation of an instrument for assessing young children’s proficiency in the ScratchJr programming language. *Computer Science Education.* DOI: 10.1080/08993408.2021.1956216

Flannery, L.P., Kazakoff, E.R., Bontá, P., Silverman, B., Bers, M.U., and Resnick, M. (2013). *Designing ScratchJr: Support for early childhood learning through computer programming: Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13)* (pp. 1-10). New York, NY: ACM. doi:10.1145/2485760.2485785

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*, *111*(5), 764.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational Thinking in Compulsory Education: Towards an Agenda for Research and Practice. Education and Information Technologies, 20(4), 715-728. doi: 10.1007/s10639-015-9445-8

Webb, M., Bell, T., Davis, N., Katz, Y. J., Reynolds, N., Chambers, D. P., ... & Mori, N. (2017). Computer science in the school curriculum: Issues and challenges. In *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing: 11th IFIP TC 3 World Conference on Computers in Education, WCCE 2017, Dublin, Ireland, July 3-6, 2017, Revised Selected Papers 11* (pp. 421-431). Springer International Publishing.

Whyte, R., Ainsworth, S., & Medwell, J. (2020). Designing Multimodal Composition Activities for Integrated K-5 Programming and Storytelling.

Table 1.
Demographic information for the students by intervention condition.

	Control (398)		Treatment (349)	
	N	%	N	%
Gender (Female)	211	53.02	176	50.43
Gender(Male)	187	46.98	173	49.57
Ethnicity				
White	278	69.85	224	64.18
Hispanic	59	14.82	69	19.77
African American	27	6.78	16	4.58
Asian	15	3.77	19	5.44
Other	19	4.77	21	6.02
Grade Level				
Kindergarten	52	13.07	107	30.66
Grade 1	187	46.98	99	28.37
Grade 2	159	39.95	143	40.97
Disability	51	12.81	39	11.17
Free/Reduced Lunch	72	18.09	67	19.20

Table 2.
Descriptive statistics of students' test scores by intervention condition.

	Control Group				Treatment Group			
	Mean	SD	Min	Max	Mean	SD	Min	Max
Math Pre-test	0.46	0.94	-2.39	2.70	0.36	0.97	-2.33	2.33
Math Post-test	0.59	0.99	-3.16	2.33	0.60	0.94	-2.66	2.33
Literacy Pre-test	0.10	1.07	-2.33	3.36	-0.06	1.12	-2.33	3.27
Literacy Post-test	0.37	1.00	-2.43	2.33	0.26	1.03	-2.33	2.33
CSA Pre-Test	3.42	2.07	0.00	11.40	3.38	2.36	0.00	26.00
CSA Post-Test	5.49	3.38	0.00	26.00	11.56	6.43	2.20	39.00
CSA Gain	2.07	2.77	-3.40	15.20	8.18	5.86	-5.80	32.10

Table 3.
Correlation matrix of all the variables.

	Math Pre-test	Math Post-test	Literacy Pre-test	Literacy Post-test	CSA Pre-test	CSA Post-test	Grade
Math Pre-test	1.00						
Math Post-test	0.78***	1.00					
Literacy Pre-test	0.62***	0.57***	1.00				
Literacy Post-test	0.69***	0.69***	0.80***	1.00			
CSA Pre-test	0.13***	0.11**	0.15***	0.13***	1.00		
CSA Post-test	0.24***	0.23***	0.21***	0.21***	0.38***	1.00	
Grade	-0.06	-0.04	0.01	0.03	0.40***	0.23***	1.00
Condition	-0.05	0.004	-0.07	-0.05	-0.01	0.52***	-0.11**

Note: * $p < .05$, ** $p < .01$, *** $p < .001$.

Table 4.
Multilevel Regression Models Predicting Math and Literacy Post-tests.

	Math Model 1	Math Model 2	Literacy Model 1	Literacy Model 2
	B *(SE)*	*B* *(SE)*	*B* *(SE)*	*B* *(SE)*
Fixed Effect				
Intercept	0.24** (0.09)	0.25** (0.09)	0.30** (0.10)	0.30** (0.01)
Grade	0.03 (0.03)	0.01 (0.03)	0.003 (0.02)	-0.01 (0.02)
Condition	0.001 (0.09)	-0.04 (0.09)	-0.05 (0.10)	-0.09 (0.10)
Math Pre-test	0.74*** (0.02)	0.72*** (0.03)	—	—
Literacy Pre-test	—	—	0.73*** (0.02)	0.72*** (0.02)
CSA Gain	—	0.06* (0.03)	—	0.07** (0.03)
Random-effects				
Intercept	0.08 (0.04)	0.09 (0.04)	0.11 (0.05)	0.11 (0.05)
Residual	0.32 (0.02)	0.31 (0.02)	0.31 (0.02)	0.31 (0.02)
ICC	0.21 (0.08)	0.22 (0.08)	0.26 (0.09)	0.26 (0.09)
N	680	680	681	681

Note: Coding Stages Assessment (CSA).

Figure 1.

Visual Representation of a CAL-ScratchJr Lesson.