Impact Study of the Coding as Another Language Curriculum

Authors: Zhanxia Yang, Patricia Moore Shaffer, Courtney Hagan,

Parastu Dubash, Marina Bers

March 2023

Abstract

The aim of this study was to explore how the Coding as Another Language (CAL) curriculum, developed by Boston College's DevTech Research Group and utilizing the ScratchJr app, impacted students' computational thinking, coding skills, and reading comprehension. To accomplish this, the research team randomly assigned thirteen schools in a northeastern state of the United States to teach the Coding as Another Language curriculum or to a "business as usual" control condition. These thirteen schools were randomly assigned to either the treatment group (CAL condition) or control group, with six schools designated to the CAL condition and seven schools designated to the control group. Participants in the study, referred to as Impact Study A, initially included 37 kindergarten, first-, and second-grade teachers, including supporting teachers, and 464 kindergarten, first- and second-grade students from 28 classrooms in the treatment group and 44 teachers including supporting teachers and 488 Kindergarten, first, and second-grade students from 36 classrooms in the control group. Hierarchical linear modeling was used to determine the impact of the CAL curriculum on first and second grade students' computational thinking, coding skills, and reading comprehension. Results showed that the CAL curriculum intervention had a significantly positive impact on students' coding performance while no notable difference was found on students' computational thinking. Additionally, an examination of students' standardized literacy achievement across the two conditions found no notable difference on students' standardized literacy achievement. This implied that even though the treatment group students allocated regular class time for the CAL curriculum, the students in the treatment group showed comparable growth on the standardized assessments with the students in the control group.

This research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through grant #U411C220202 to DevTech Research Group at Boston College.

Contents

Background

In the automated economy, computer programming is essential across diverse disciplines. Occupations that value programming skills provide as much as 20% of "career-track" job openings (Burning Glass Technologies, 2016), and the number of jobs in information technology will grow 12.5% from 2014 to 2024 (Fayer, Lacey, & Watson, 2017). To meet this growing need, there has been an increase in new educational policies and frameworks at the federal and state level to prepare K-12 students for CS related professions.

However, while most of the educational implementation and research is happening at the late elementary, middle school, high school, and college (Guzdial, 2008; Wilson, Sudol, Stephenson & Stehlik, 2010), the frameworks, standards, and best practices mandate to start in kindergarten (Barron et al., 2011; International Society for Technology in Education, 2007; NAEYC and Fred Rogers Center for Early Learning and Children's Media, 2012; U.S. Department of Education, 2010; White House, 2016; U.S. Department of Education & U.S. Department of Health and Human Services, 2016; Paciga & Donohue, 2017).

There are both economic and developmental reasons for the choice to start early. Research shows that educational interventions that begin in early childhood are associated with lower costs and more durable effects than interventions that begin later on (e.g., Cunha & Heckman, 2007; Heckman & Masterov, 2007). Two National Research Council reports—Eager to Learn (2001) and From Neurons to Neighborhoods (2002)—detail the importance of early experiences for later school achievement. Furthermore, research shows how children who are exposed to STEM curriculum at an early age demonstrate fewer gender-based stereotypes regarding STEM careers, increased interest in engineering and computer science (Sullivan & Bers, 2018; Metz, 2007; Steele, 1997), and fewer obstacles entering these fields later in life (Madill et al., 2007; Markert, 1996). Research also suggests that addressing the under-representation of women in computer science is critical to improve early education experiences (Varma, 2009).

However, if computer science education is to start in the early years, when children are just starting to develop literacy and numeracy skills as well as learn "schooling," there is a need for pedagogical approaches, curriculum, and programming languages that are developmentally appropriate for young children (Bers, 2018; Bers et al, 2022). The need to fulfill the work pipeline is not enough of a rationale for the introduction of computer science in early childhood education and thus it must be integrated with foundational content areas such as math and literacy. If we are going to start computer science education in kindergarten, the rationale shouldn't be the creation of the future workforce, but the future citizenry (Bers, 2022).

The work conducted in this study is grounded on Bers' previous work that understands "Coding as a Another Language (CAL)" (Bers, 2018, 2019). Within this framework, those who learn how to code from a young age will not only be able to participate in the automated economy but will also have a civic voice. As children learn how to code, they also develop their creativity to grow a society of innovators (Resnick, 2018). A literate person knows that reading and writing are tools for meaning making and, ultimately, tools of power because they support new ways of thinking (Papert, 1980). The same is true for computer programming and computational thinking.

Researchers have coined the term "computational thinking" to refer to an analytical process rooted in the discipline of computer science. It involves thinking recursively, applying abstraction, breaking up a complex problem in smaller tasks, and using heuristic reasoning to discover a solution (Wing, 2006; 2011). There is debate whether computational thinking can be classified as a unique category of thought

(Gadanidis, 2017; Pei, Weintrop, & Wilensky, 2018). However, the term has grown popular at a time when schools are incorporating computer science in massive ways and developing frameworks (K–12 Computer Science Framework Steering Committee, 2016). While computational thinking is not the same as coding, the act of coding can facilitate the spread of computational thinking (Bers, 2021; Relkin et al, 2021). CAL addresses the teaching of computational thinking through both unplugged activities and on-screen coding with ScratchJr, integrated with other content areas, in particular math and literacy, in the K-2 segment.

ScratchJr is the first programming language explicitly designed for young children, 5 to 7 years, which meets their developmental needs. ScratchJr is the result of a long-lasting collaboration between the DevTech Research Group, now at Boston College, and the MIT Lifelong Kindergarten Group funded by the National Science Foundation and the Scratch Foundation (Bers & Resnick, 2015). ScratchJr enables children to create interactive stories and games by snapping together graphical programming blocks to make characters move, jump, dance, and sing. Through ScratchJr young children learn how to code and how to engage in computational thinking while creating personally meaningful projects. Since its launch in 2014, there have been 40 million users with an average of 2.4 million users per month over the past 6 months and has actively been used in every country in the world (except North Korea). The app can be freely downloaded to iPads, Android tablets, iPhones and Android phones, Amazon Kindle tablets, and Chromebook devices, and it has been translated to Spanish as well as a dozen other languages.

The ScratchJr team began collecting analytics data in 2016. Since then, as of February 2023, over 167 million projects have been created, and existing projects have been edited over 275 million times, indicating that users are improving and debugging their projects over time. The DevTech Research Group has developed curricula and teaching materials to integrate ScratchJr with other content areas in early childhood in both formal and informal learning settings. Three twenty-hour curriculum units have been developed to accompany the ScratchJr app: Animated Genres, Playground Games, and Reinforcing Common Core. In addition, several activities were developed in the form of coding cards (Bers & Sullivan, 2018) as well as the Coding as Another Language curriculum to support literacy integration. The CAL curriculum builds on previous work by also incorporating math, low-tech materials, and unplugged games to address powerful computational ideas, skills, and habits of mind that promote computational thinking.

Pilot studies found that children in K-2 can master ScratchJr, which in turn supports learning of problem solving, foundational programming, and discipline-specific content in math and literacy (Flannery et al., 2013). Combined pilot work representing a total sample of N = 333 children (aged 5-7 years) revealed that children used ScratchJr to make creative projects, which supported literacy practices of exploring and utilizing narrative structures, decoding symbols, and reading and writing digital media. (Flannery et al., 2013; Portelance & Bers, 2015). Further, pilot work demonstrated that ScratchJr can support learning outcomes when educators have diverse teaching approaches, although positive learning outcomes are more pronounced when the learning is child-directed and open-ended (Strawhacker, Lee, & Bers, 2017).

Despite programming becoming popular and ScratchJr and its resources being widely utilized, there is a lack of well-researched, evidence-based integrated early childhood computer science curriculum and professional development strategies. Technology and pedagogy are not the same thing. As new programming languages that are developmentally appropriate emerge and are widely used, such as ScratchJr, there is a need to conceptualize pedagogical approaches for teaching computer science in the early years. These approaches must be consistent with developmentally appropriate practice (Bredekamp, S, 1987) and must embrace the maturational stages of children by inviting play and discovery, socialization and creativity (Bers, 2018a).

Study Description

Research Questions for the study

Research Question 1: What is the impact of three months of CAL curriculum on kindergarten, first, and second grade student's computational thinking compared to the business-as-usual condition?

Research Question 2: What is the impact of three months of CAL curriculum on kindergarten, first, and second grade student's coding skills compared to the business-as-usual condition?

Research Question 3: What is the impact of three months of CAL curriculum on first and second grade students' standardized literacy performance compared to the business-as-usual condition?

Intervention Condition

The CAL curriculum builds on DevTech's previously developed pilot units (Bers, 2018) and is aligned with the K-12 Computer Science Framework (K-12 Computer Science Framework Steering Committee, 2016) and the Standards for Technological Literacy (International Technology and Engineering Education Association, 2007), as well as the Common Core Frameworks for Math and Literacy (National Governors Association Center for Best Practices & Council of Chief State School Officers, 2010), and Virginia Department of Education's Standards of Learning for English and Standards of Learning for Computer Science (Virginia Department of Education, 2017).

The CAL curriculum is organized around powerful ideas that are fundamental to computational thinking and, at the same time, are developmentally appropriate for young children. The curriculum introduces coding and computational thinking in a playful, developmentally appropriate way by integrating powerful ideas of computer science with literacy skills.

Students were intended to receive a total of 24 lessons of 45 minutes each.

The CAL curriculum was implemented using a delayed treatment design. Following a pilot study during winter 2021 at selected schools in the U.S., thirteen schools were randomly assigned to either the treatment condition (CAL curriculum) or the control condition. In the six treatment schools, teachers were trained during summer 2021 in the delivery of the CAL curriculum and provided resources and support to implement the curriculum during a 12-week period (Fall 2021). In the seven control schools, teachers delivered business-as-usual during School Year (SY) 2021-2022 and delayed implementation of the intervention until SY 2022-2023; Teachers in control schools did not have access to training or the curriculum during SY 2021-2022.

For Impact Study A held during SY 2021-2022, the DevTech Research Group trained the teachers delivering the curriculum in CAL schools. The impact of the intervention is measured by comparing student outcomes in the treatment and the control group. Student outcomes are measured before and after the implementation of the curriculum/business-as-usual.

Program Intervention

Before implementation of the CAL curriculum, 24 kindergarten, first-, and second- grade teachers in 7 CAL schools received professional development (PD) training on the implementation of ScratchJr in the classroom and the CAL curriculum. This included attending two 2-hour PD workshops (total of 4 hours) and completing their own ScratchJr project.

Students were intended to receive a total of 24 CAL curriculum lessons of 45 minutes each. Teachers implemented the CAL curriculum lessons during regular school hours in between regular curricula. For Impact Study A, primary teachers, as well as specialists, implemented the curriculum to the whole class, in-person.

In Impact Study A, 24 kindergarten, first-, and second- grade teachers (100% female; average number of years teaching = 21 years) delivered the CAL curriculum to 564 students; among these students, 464 consented to participate in the study. These students were enrolled in the treatment schools as of September 2021. The average number of CAL lessons completed by the end of implementation was 20.2. The total number of CAL lessons to complete was 24 lessons. Teachers were able to begin implementing the lessons in their classroom after pre-testing and continued until the end of the school year.

Fidelity of implementation to the program was assessed using a fidelity of implementation measure and thresholds. This study was assessed for fidelity along three different key components (CAL curriculum, DevTech Research Group teacher training, and coaching), with seven different indicators (curriculum dissemination, group training participation, group training content, embedded onsite coaching–availability, embedded onsite coaching–satisfaction, virtual coaching–availability, and virtual coaching–satisfaction) nested across those three components (see table 10).

Setting

This multi-site study took place at 13 different schools within different districts in a northeastern state in the United States. Each school district was considered to be public, high-poverty, and in an urban environment. The CAL curriculum was deployed to child participants within the classroom context by teachers (who had been trained with the CAL curriculum) at variable rates across the length of the intervention. Methods for randomization at each site were standardized.

Comparison Condition

The CAL curriculum was implemented using a delayed treatment design. Following a pilot study in SY 2020-2021 at other U.S. schools, Group 1 (treatment group) was trained during summer 2021 in the delivery of the CAL curriculum and provided resources and support to implement the curriculum during a 12-week period beginning in Fall 2021. During SY 2021-2022, Group 2 schools (control group) delivered treatment-as-usual and delayed implementation of the intervention until SY 2022-2023; Group 2 teachers did not have access to training or the curriculum during SY 2021-2022.

Study Participants

In Impact Study A during spring 2021, public schools were invited by the state department of education to submit applications to participate in the CAL curriculum study. Title I schools were particularly encouraged to apply. Participation was voluntary. All schools that applied for participation were accepted except for one school; lack of funding resources from the state department of education was the reason the one school was not accepted for participation. However, two other schools in that same school district were selected to participate. School principals or district supervisors from the schools where these participating teachers taught were asked to send letters of support along with the original applications. Following school recruitment, random assignment of schools was done in May 2021 before any outcome testing. At the beginning of Fall 2021 students' parental consents to participate in the research study were then sent to be collected and coordinated by a site coordinator. All parental consents were voluntary. Students who joined the schools after the parental consent process was completed are not included in the sample.

Inclusion criteria for the study are schools at least serving one of the targeted grade levels of Kindergarten, first grade, and second grade. No specific exclusion criteria were applied. The schools were randomized with blocks. Blocks were formed using two criteria: poverty quartile and number of estimated participating students in order to ensure comparable demographics and comparable sample sizes across the treatment and the control group: 1) schools in the lowest and second-to-lowest poverty quartile that expect fewer than 100 students to participate; 2) schools in the lowest and second-to-lowest poverty quartile that expect more than 100 students to participate; and 3) schools in the highest and second-to-highest poverty quartile. As a result of the random assignment, six schools were assigned in the intervention group and seven in the comparison group. Figure 1 and 2 shows the CONSORT flow of participants in Study A.

Figure 1. CONSORT Flow of K-2nd Grade Participants for the CSA and TechCheck Outcomes

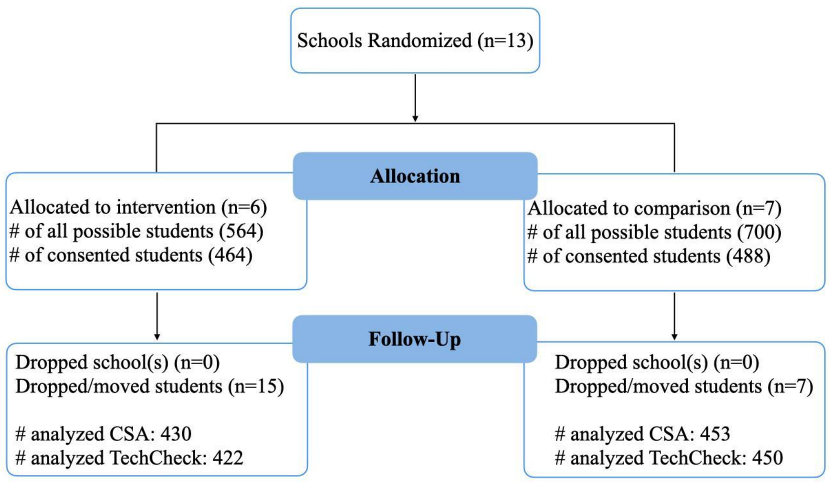
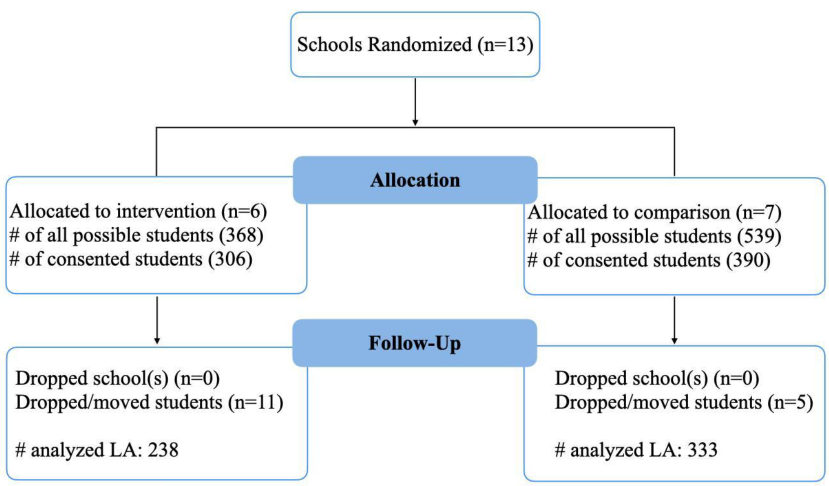


Figure 2. CONSORT Flow of 1st and 2nd Grade Participants for the Literacy (LA) Outcome



Sample Alignment with Those Served by the Program

The evaluation sample for the creative coding proficiency as measured by CSA and computational thinking as measured by *TechCheck* includes all consented students who were offered the intervention over the duration of the evaluation and for whom each outcome measure was not missing.

The evaluation sample for literacy assessment included first and second grades only because the majority of schools in Study A did not administer literacy assessment for Kindergartens. Additionally, among the 13 participating schools, one school in the treatment condition was excluded for the evaluation of literacy assessment due to the type of assessment administered by the school district. The literacy assessment in the excluded school used ordinal categories whereas the literacy assessments in the included schools were continuous and standardized as a z-score. Also, there is no corresponding type of literacy assessment used in the control group as it is in the treatment group.

As a result of this exclusion, 37 students were excluded from the literacy assessment analysis, 36 of which had consented for research participation. The excluded 36 consented students account for 11.8% of all first and second grade consented students in the treatment evaluation sample and 5.2% of consented students across treatment and control conditions.

Design and Measures

Independence of the Impact Evaluation

The evaluation was conducted by Shaffer Evaluation Group (SEG). Shaffer Evaluation Group worked closely with the Research and Evaluation team of the DevTech Research Group, which worked independently from the intervention development and implementation team. A firewall was implemented between the DevTech Research and Evaluation team and the intervention development and implementation team. All decisions regarding assignment, data collection, data analysis, and final reporting were made by the Shaffer Evaluation Group.

While impact analyses were conducted by the DevTech Research and Evaluation team, the Shaffer Evaluation Group conducted verification of the impact models for each research question of the study. During this verification process, SEG ensured that all data at the student level that was collected was included in the analysis. If student-level data was dropped from analysis, SEG verified that the reason for dropping the case was objective in nature. In addition to verifying the data cleaning process, SEG also verified the analysis. During this verification process, SEG verified that the code DevTech used included all appropriate variables in each unconditional and conditional model, verified the correct number of observations and schools in each unconditional and conditional model, verified each beta was in bounds, and finally ran RMarkdown to spot-check that the same results were found.

Pre-Registration of the Study Design

This study was originally registered in the Registry of Efficacy and Effectiveness Studies (REES) on August 22, 2022. The original registry ID number is 13200.1v1. The original study design was restricted to the second grade. The original research questions as well as findings that address the pre-specified research analyses are identified in this report by a ‘+’ symbol. The study design registration was updated on Friday February 24, 2023 to reflect the decision to expand the study to include kindergarten and first grade.

Design

The CAL curriculum was implemented using a delayed treatment design. During Impact Study A, the DevTech Research Group trained the teachers who delivered the curriculum in the schools. The impact of the intervention was measured by comparing student outcomes in the treatment group (Group 1) and the control group (Group 2). Student outcomes were measured before and after the implementation of the curriculum/business-as-usual.

The impact study used a randomized control trial design that assigned schools to either the treatment or to a control condition. Teachers in treatment schools were trained to teach the curriculum, so that the intervention is delivered at the classroom level. Group 1 implemented the CAL curriculum supported by training from the DevTech Research Group during the first year of implementation. Group 1 students were compared using three-level hierarchical linear modeling (HLM), controlling for covariates at the student, teacher, and school levels, to test for differences in the outcomes of computational thinking, coding skills, and literacy comprehension skills.

Measures

To address each of the research questions, this study used multiple instruments to assess the various outcomes (Table 1). Each instrument is described in more detail below.

Table 1. List of Instruments Used by Outcome Measures

Domain	**Name of Instrument**	**Subtest(s) of instrument used, if any**	**Timing of measurements**	**Baseline measure**	**Variable construction**
Computational Thinking	TechCheck	n/a	Right after completion of curriculum (Group 1, 2)	Score before start of curriculum (fall/winter)	Raw score
Creative Coding Knowledge	Coding Stages Assessment	n/a	Right after completion of curriculum (Group 1, 2)	Score before start of curriculum (fall/winter)	Raw score
Comprehension	STAR Reading	Literary text	Spring/summer after curriculum (Group 1, 2)	Score before start of curriculum (fall/winter)	Raw domain score
Comprehension	iReady	Comprehension: literature	Spring/summer after curriculum (Group 1, 2)	Score before start of curriculum (fall/winter)	Domain scale score
Comprehension	AimsWeb Plus	Reading Comprehension	Spring/summer after curriculum (Group 1, 2)	Score before start of curriculum (fall/winter)	Raw domain score
Reading	Developmental Reading Assessment	n/a	Spring/summer after	Score before start of	Scale score

			curriculum (Group 1, 2)	curriculum (fall/winter)	
Reading	Fastbridge Reading Screener	n/a	Spring/summer after curriculum (Group 1, 2)	Score before start of curriculum (fall/winter)	Scale score

TechCheck (Relkin et al., 2020) - TechCheck assesses students' computational thinking. This measure was developed by the intervention developer and validated with 769 five-to-nine-year-old students. In previous studies, it had a reliability of $\alpha = 0.68$. TechCheck was correlated with TACTIC-KIBO ($r = .53$). This measure has 15 questions that are in a multiple-choice format. Data for the current study was collected November 2021 - June 2022. Example items include: "What comes next" after showing a series of shapes.

Coding Stages Assessment (CSA; de Ruiter & Bers, 2021 – CSA assesses students coding knowledge and programming language. This measure was developed by the intervention developer and validated with 118 five-to-eight-year-olds. This assessment has 27 items presented in three stages. Data for the current study was collected November 2021 - June 2022. Example items include: "Can you show me where you tap on the screen to make the program go?". In previous studies, internal consistency was $\lambda_6 = .94$.

Comprehension - literacy assessment included standard assessments of:

Fastbridge - This assessment includes three universal screening tests to capture students who may need more support throughout the school year. Fastbridge assesses five reading components: phonemic awareness, phonics, fluency, vocabulary, and comprehension. The internal consistency across K through eighth grade ranged from 0.91 (K) to 0.96 (2nd, 7th, and 8th).

STAR - The STAR assessment encompasses five components: word knowledge and skills, comprehension strategies and construction meaning, understanding author's craft, analyzing literary text, and analyzing argument and evaluating text. This assessment had an overall reliability of 0.98.

DRA - The Developmental Reading Assessment provides benchmark assessment, word analysis, and progress monitoring. The DRA assesses oral reading fluency and comprehension. Oral reading fluency had a median internal consistency of 0.88. Comprehension had a median internal consistency of 0.82.

Aimsweb - The Aimsweb reading assessment captures letter naming fluency, oral reading fluency, phoneme segmentation, print concepts, auditory vocabulary, initial sounds, letter word sounds fluency, and word reading fluency. The internal reliability across K - eighth grade for Aimsweb ranges from 0.87 - 0.95.

iReady - The iReady Diagnostic Reading assessment covers phonological awareness, phonics, high-frequency words, vocabulary, comprehension of informational text, and comprehension of literature.

Sample Sizes and Attrition

Study A sample size information is reported in Table 2, both for clusters and individuals, by condition and for each outcome. Outcomes for coding skill are labeled as Post-CSA, computational thinking as Post-TechCheck, and standardized literacy comprehension performance as Post-LA. Although specified in the registered evaluation study plan, teacher attrition was not incorporated into the analysis since during actual implementation a decision by an individual teacher to not participate in the study often did not equate to loss of students from the study.

Table 3 summarizes the observed overall attrition and differential attrition and assigns an attrition level by outcome level according to WWC boundaries for defining high versus low attrition for randomized control trials (What Works Clearinghouse, 2022). Attrition at the cluster level may be defined as low with the exception of the post literacy assessment outcome. As discussed earlier in the Alignment of the Sample section, this exception was due to the exclusion of one treatment school from the sample because the school's literacy assessment used ordinal categories that were incompatible with the planned z-score analysis. Student-level attrition may be defined as high for the post-CSA and post-Tech Check outcomes, due primarily to the parent consent rate (82.27%) and student absences during testing. The student-level attrition for the post literacy assessment may be defined as low.

Data Analysis and Findings

Baseline Equivalence

The baseline mean difference between the treatment and control groups was calculated using a statistical model that adjusts for clustering. Multilevel modeling, with student-level being the first-level and school-level being the second level, was used. Specifically, the baseline measures of outcomes at the student level were used as the predicted variable, and the condition variable (e.g., intervention or comparison) was used as the predictor while allowing the school intercept to vary. The participants differed from those included in each analytic sample slightly because only students who completed both pre and post-assessments were included in the impact analyses.

Table 4 presents descriptive statistics by condition for each analytic sample in the study: baseline measures of coding skills as measured by CSA, computational thinking as measured by TechCheck, and literacy comprehension as measured by standardized assessments in Impact Study A. The standardized difference (Hedges' g) between control and treatment groups on all outcome assessments is lower than 0.05 and therefore satisfies the WWC standard for baseline equivalence (What Works Clearinghouse, 2022.).

Background characteristics not required for assessing baseline equivalence are also presented in Table 5 by each condition, including gender (Female or Male), Individualized Education Plan status (Yes or No), Limited English Language Proficiency status (Yes or No), and free/reduced lunch program status (Yes or No).

Program Effects

<u>Approach to estimating program effects:</u>
The method used to estimate the impacts of CAL-ScratchJr Curriculum implementation is multilevel modeling at two levels. See the statistical models below:

- **Level-1: Student Level**

$$Y_{ij} = \beta_{0j} + \beta_{1j}\left(Y_{ij}^{*}\right) + \sum_{m=1}^{M} \beta_{2.mj} X_{mij} + \varepsilon_{ij}$$

- **Level-2: Cluster (School) Level**

$$\beta_{0j} = \gamma_{00} + \gamma_{01}(T_j) + \sum_{q=1}^{Q} \gamma_{02.q} W_{qj} + \mu_{0j}$$
$$\beta_{1j} = \gamma_{10}$$
$$\beta_{2.mj} = \gamma_{2.m0}$$

Where,

- Y_{ij} = the outcome score for the i^{th} student in the j^{th} school.
- β_{0j} = the intercept for school *j*.
- β_{1j} = the effect of pretest in school *j*.
- Y_{ij}^{*} = a pre-test measure for the i^{th} student in the j^{th} school.
- $\beta_{2.mj}$ = the effects of student covariates in school *j*.
- X_{mij} = the m^{th} of M additional covariates for student *i* in school *j*.
- ε_{ij} = a residual error term for student *i* in school *j*.
- γ_{00} = the mean intercept
- γ_{01} = estimated treatment impact
- T_j = 1 if school *j* is assigned to treatment (CAL), and = 0 if school *j* is assigned to comparison.
- $\gamma_{02.q}$= the effect of school-level covariate (percent of students receiving free/reduced-price lunch);
- W_{qj}= the q^{th} of Q covariates for school *j*.
- μ_{0j} = random intercept term – deviation of cluster *j*'s mean from the grand mean, conditional on covariates; assumed to be normally distributed with mean 0 and variance τ_{00}^{2}.
- γ_{10} = mean effect of pretest
- $\gamma_{2.m0}$= mean effect of student covariate m.

All outcome measures were collected at the student level and the treatment indicator was at the school level. Clustering was addressed by allowing random intercept among schools.

As presented in the statistical model, all student's pre-test scores were included as a control variable. In addition, grade level, demographics including gender, IEP status, LEP status, Free/Reduced Lunch Program status, school-level pretest score, and school level free/reduced lunch percentage were included as control variables.

No participants or units were excluded from the analysis except for the ones that are missing data or had data incompatible for analysis (see Alignment of the Sample section).

Approach to handling missing data:

Only students with data present at both the pre-test and the post-test were included in the analytical sample. Baseline equivalence from all outcome measures was established based on the analytical sample, which list-wise deleted those students who had missing data at either the pre or the post time points.

Table 6 presents the extent of missing data by outcome and condition. One should note that the numbers used in this table include all possible roster students. The number of consented students for research is smaller than the number of students from the roster, thus, the actual missing rate from the consented students is smaller than the ones in the table.

Findings

Regarding research question 1, which inquired about the impact of three months of the CAL curriculum on kindergarten, first, and second grade student development of computational thinking, the treatment group showed no notable difference compared to the business-as-usual control group (Hedge's g = 0.04). Table 7 presents descriptive statistics (means, standard deviations) and sample sizes by condition for each outcome measure at each time point (pre and post assessments).

Regarding research question 2, which inquired about the impact of three months of CAL curriculum on the Kindergarten to second grade student participants' development of coding skills, the treatment group showed a significantly higher increase of coding skills compared to the business-as-usual control group (Hedge's g = 0.47). This Hedge's g indicated a large effect of the impact of the curriculum. The CAL-ScratchJr curriculum was successful in terms of improving students' coding performance.

Regarding research question 3, which inquired about the impact of three months of CAL curriculum on the first and second grade student participants' literacy performance compared to the business-as-usual condition, the treatment group showed no notable difference compared to the business-as-usual comparison group (Hedge's g = -0.01). Table 8 presents sample sizes, means, standard deviations, and the beta coefficients for the grouping variable (treatment or control) by condition for each outcome measure.

Table 9.1 to Table 9.3 report the sample sizes, means, and standard deviations in both conditions for samples with and without missing data.

Discussion

Considering the time taken away from regular classes in the treatment group compared to the control group, the comparable growth in the standardized literacy assessment in the treatment group is a success. The results implied that even with some class time reserved for the CAL curriculum, the treatment group students' standardized literacy performance is comparable with their peers in the control group. The results also provide empirical evidence that future CAL curriculum implementation may not negatively impact students' performance on standardized literacy assessments.

Compared to the control group, the treatment group did not show a notable difference in students' computational thinking. Few research studies have examined the effect of early childhood computer science curriculum interventions on students' computational thinking. Among the few we found, research reported mixed findings regarding the impact of a computer science intervention on students' computational thinking. For example, Grillo-Hill, Mahoney, Chow, and Li (2019) examined the effect of codeSpark Academy on one domain of computational thinking and found no significant effect. Oluk and Cakir (2021) examined the effect of code.org on sixth grade students and reported a significant increase on students' algorithm development and computational thinking skills in the intervention group. However, the grade level and limited details regarding the number of schools or the number of classrooms investigated left little generalizability of the study. Additionally, code.org activities include the unplugged computational thinking activities, which made the curricula not comparable with other curricula that do not explicitly teach computational thinking such as codeSpark Academy or the CAL curricula. Our research team, Relkin et al. (2021) also conducted a quasi-experimental study and found a significant effect of the CAL-KIBO curriculum on first graders' computational thinking but not on second graders. However, the significant effect among the first graders was associated with a significantly higher baseline in the intervention group, which made it difficult to attribute the effect to the intervention.

In summary, the CAL-ScratchJr showed a positive impact on students' coding performance. While no significant impact was found on students' computational thinking, recent research does not offer valid evidence that similar computer science curricula affect computational thinking either. The results also provided empirical evidence that the CAL curriculum can be implemented in authentic classroom settings without negatively impacting students' performance on standardized literacy assessments even though some class time was allocated to the CAL curriculum.

The study was conducted during the COVID-19 pandemic when many schools, teachers, and students were challenged with absences and quarantines. As a result, the CAL curriculum was implemented at various levels of fidelity. Teachers do report that time was one of their main challenges during the pandemic. Nevertheless, teachers shared how excited their students were to engage with the CAL curriculum.

The DevTech Research Group plans to expand the program to more schools, including engaging additional schools in CAL professional development training and providing support for implementation of the CAL curriculum. Planning is underway to make automated interactive assessments to reduce cost and improve efficiency in research, since the current assessment of coding skills and computational thinking were administered one-on-one by trained research assistants, which is labor- intensive.

Fidelity of Implementation Study

Fidelity Measurement

The fidelity of CAL curriculum implementation was assessed by examining three key components: (1) revised CAL curriculum, (2) DevTech Research Group teacher training, and (3) teacher coaching. In Impact Study A, the revised CAL curriculum component encompassed one indicator: curriculum dissemination. The DevTech training component encompassed two indicators: group training participation and group training content. The coaching component encompassed four indicators: onsite coaching availability, onsite coaching satisfaction, virtual coaching availability, and virtual coaching satisfaction. The scoring model for assessing fidelity of implementation is presented in Table 10.

Fidelity Findings

Overall, Impact Study A achieved fidelity for some of the three components. In key component one, curriculum dissemination, 18 teachers (out of 24 total) accessed the curriculum, and therefore earned a score of '1' at the teacher-level. At the school-level, four schools had 90% or more teachers earn a score of '1', while there were two schools that had 51-75% of teachers earn a score of '1' and one school had 26-50% of teachers earn a score of '1'. This resulted in an overall sample-level score of '2', with 57% of schools earning a score of '1' or inadequate implementation. However, there may have been stronger implementation of curriculum dissemination, as all teachers received copies of the curriculum multiple times. There were low rates of responses for lesson log entries by teachers, potentially impacting the fidelity score.

In key component two, DevTech Research Group teacher training, 24 teachers (out of 24 total) participated in the provided group training. This included attending training in synchronous or asynchronous format. Further, eleven out of twelve indicators were observed during the first part of the group training and seven out of seven indicators were observed during part 2. Overall, across indicators in component two, there was a score of ‘4’ indicating adequate implementation.

For coaching, the third key component, seven teachers out of the 24 requested either an onsite or virtual coach. Of the two teachers who requested a virtual coach, two received a response to the request. There was an average satisfaction of 4.0 (on a rating scale of 1 = needs a lot of improvement - 5 = couldn’t be better) with virtual coaching. Of the seven teachers who requested an onsite coach, all seven received a response to their request. There was an average satisfaction of 4.3 with onsite coaching. Overall, this component earned a sum score of ‘9’ indicating adequate implementation.

Details of fidelity of implementation scores are provided in Table 11.

References

Barron, B., Cayton-Hodges, G., Bofferding, L., Copple, C., Darling-Hammond, L., & Levine, M. H. (2011). *Take a giant step: A blueprint for teaching young children in a digital age*. New York: Joan Ganz Cooney Center at Sesame Workshop.

Bers, M. U. (2018a). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education*, *3*(3), 8.

Bers, M. U. (2018b). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. In *2018 IEEE global engineering education conference (EDUCON)* (pp. 2094-2102). IEEE.

Bers, M. (Ed.). (2021). *Teaching Computational Thinking and Coding to Young Children*. IGI Global. http://doi:10.4018/978-1-7998-7308-2

Bers, M. U., & Resnick, M. (2015). *The official ScratchJr book: Help your kids learn to code*. No Starch Press.

Bers, M. U., & Sullivan, A. (2018). *ScratchJr coding cards: Creative coding activities*. No Starch Press.

Bredekamp, S. (1987). *Developmentally appropriate practice in programs serving children from birth through age eight*. Washington, DC: National Association for the Education of Young Children.

Burning Glass Technologies. (2016). *Beyond point and click: the expanding demand for coding skills*. Retrieved from: http://hdl.voced.edu.au/10707/429586.

Cunha, F., & Heckman, J. (2007). The technology of skill formation. *American economic review*, *97*(2), 31-47.

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*, *32*(4), 388-417.

Fayer, S., Lacey, A., & Watson, A. (2017). STEM occupations: Past, present, and future. *Spotlight on Statistics*, *1*, 1-35.

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. In *Proceedings of the 12th international conference on interaction design and children (IDC '13)*. ACM, New York, NY, USA, 1-10.

Gadanidis, G. (2017). Five affordances of computational thinking to support elementary mathematics education. *Journal of Computers in Mathematics and Science Teaching*, *36*(2), 143-151.

Grillo-Hill, A., Mahoney, C., Chow, E., & Li, L. (2019). StoryCoder Classroom Feasibility Study [White Paper]. WestEd. Retrieved from: https://edcuration.com/resource/vendor/348/codeSpark_Feasibility%20Memo_Draft.pdf

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25-27.

Heckman, J. J., & Masterov, D. V. (2007). The productivity argument for investing in young children. NBER working paper 13016. Retrieved from: https://www.nber.org/papers/w13016.

Institute of Education Sciences. (undated a). *Module 2: Attrition*. Retrieved from: https://ies.ed.gov/ncee/wwc/Docs/OnlineTraining/wwc_training_m3.pdf.

Institute of Education Sciences. (undated b). *Module 3: Baseline equivalence*. Retrieved from: https://ies.ed.gov/ncee/wwc/Docs/OnlineTraining/wwc_training_m3.pdf.

International Society for Technology in Education. (2007). *National educational technology standards for students*. Retrieved from: https://www.iste.org/standards/.

K-12 Computer Science Framework Steering Committee. (2016). *K-12 computer science framework*. Retrieved from: https://k12cs.org/.

Madill, H., Campbell, R. G., Cullen, D. M., Armour, M. A., Einsiedel, A. A., Ciccocioppo, A. L., et al. (2007). Developing career commitment in STEM-related fields: myth versus reality. In R. J. Burke, M. C. Mattis, & E. Elgar (Eds.), *Women and minorities in science, technology, engineering and mathematics: Upping the numbers* (pp. 210–244). Northampton, MA: Edward Elgar Publishing.

Markert, L. R. (1996). Gender related to success in science and technology. *The Journal of Technology Studies, 22*(2), 21–29.

Metz, S. S. (2007). Attracting the engineering of 2020 today. In R. Burke & M. Mattis (Eds.), *Women and minorities in science, technology, engineering and mathematics: Upping the numbers* (pp. 184–209). Northampton, MA: Edward Elgar Publishing.

NAEYC and Fred Rogers Center for Early Learning and Children's Media. (2012). *Technology and interactive media as tools in early childhood programs serving children from birth through age 8. Joint position statement*. Washington, DC: NAEYC. Latrobe, PA: Fred Rogers Center for Early Learning at Saint Vincent College. Available at: www.naeyc.org/files/naeyc/file/positions/PS_technology_WEB2.pdf. (Accessed 28 March 2018).

National Academies of Science. (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington DC: National Academies Press.

National Governors Association Center for Best Practices & Council of Chief State School Officers. (2010). *Reacher Higher: The Common Core State Standards Validation Committee*.

Oluk, A. & Çakir, R. (2021). The effect of code. org activities on computational thinking and algorithm development skills. *Journal of Teacher Education and Lifelong Learning, 3*(2), 32-40.

Paciga, K. A., & Donohue, C. (2017). *Technology and interactive media for young children: A whole child approach connecting the vision of Fred Rogers with research and practice.* Latrobe, PA: Fred Rogers center for early learning and children's media at Saint Vincent College.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, Inc.

Pei, C., Weintrop, D., & Wilensky, U. (2018). Cultivating computational thinking practices and mathematical habits of mind in lattice land. *Mathematical thinking and learning*, *20*(1), 75-89.

Portelance, D. J., & Bers, M. U. (2015). Code and Tell: Assessing young children's learning of computational thinking using peer video interviews with ScratchJr. In *Proceedings of the 14th international conference on interaction design and children* (pp. 271-274).

Strawhacker, A., Lee, M., & Bers, M. U. (2018). Teaching tools, teachers' rules: Exploring the impact of teaching styles on young children's programming knowledge in ScratchJr. *International Journal of Technology and Design Education*, *28*, 347-376.

Steele, C. M. (1997). A threat in the air: How stereotypes shape intellectual identity and performance. *American Psychologist, 52*, 613–629.

Sullivan, A., & Bers, M. U. (2018). The impact of teacher gender on girls' performance on programming tasks in early elementary school. *Journal of Information Technology Education. Innovations in Practice*, *17*, 153.

Relkin E., & Bers, M. (2021). TechCheck-K: A Measure of Computational Thinking for Kindergarten Children. In *2021 IEEE Global Engineering Education Conference (EDUCON).* IEEE. pp. 1696-1702, https://doi.org/10.1109/EDUCON46332.2021.9453926.

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.

Relkin, E., de Ruiter, L.E., Bers, M. U. (2021). Learning to Code and the Acquisition of Computational Thinking by Young Children. *Computers & Education.* https://doi.org/10.1016/j.compedu.2021.104222

U.S. Department of Education, Office of Educational Technology. (2010). Transforming American education: Learning powered by technology. Washington, DC: U.S. Department of Education, Office of Educational Technology. Retrieved from: http://www.ed.gov/technology/netp-2010.

Varma, R. (2009). Why I chose computer science? Women in India. *Americas Conference on Information Systems (AMCIS) 2009 Proceedings.* Retrieved from: https://aisel.aisnet.org/amcis2009/413/.

Virginia Department of Education. (2017). *Computer Science Standards of Learning Resources*. Virginia Department of Education. Retrieved from: http://www.doe.virginia.gov/testing/sol/standards_docs/computer-science/index.shtml

What Works Clearinghouse. (2022). *What Works Clearinghouse procedures and standards handbook, version 5.0*. Washington, DC: U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance (NCEE). Retrieved from: https://ies.ed.gov/ncee/wwc/Handbooks.

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–36.

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. The Association for Computing Machinery. The Computer Science Teachers Association.

Appendix A: Tables

Table 2. Sample Sizes at Randomization and in Analytic Sample Needed to Assess Attrition for an RCT with Cluster-Level Assignment								
	Control Group				Treatment Group			
	Clusters[a]		Students[b]		Clusters[a]		Students[b]	
Outcome Measure	# Randomized	# Analytic Sample	# Randomized	# Analytic Sample	# Randomized	# Analytic Sample	# Randomized	# Analytic Sample
Post-CSA	7	7	700	453	6	6	564	430
Post-TechCheck	7	7	700	450	6	6	564	422
Post-LA	7	7	539	333	6	5	368	238

[a] Reported only for cluster-assignment evaluations. Not applicable for individual-assignment evaluations.
[b] Report the number of students in non-attrited clusters only, for cluster-assignment evaluations.

Table 3. Attrition Assessment for Impact Study A						
	Clusters			Students		
Outcome Measure	Overall Attrition	Differential Attrition	Optimistic Threshold	Overall Attrition	Differential Attrition	Optimistic Threshold
Post-CSA	0.00%	0.00%	Low	30.14%	-11.53%	High
Post-TechCheck	0.00%	0.00%	Low	31.01%	-10.54%	High
Post-LA	7.69%	16.67%	High	37.05%	-2.89%	Low

Table 4. Results from Baseline Equivalence Assessment								
	Control Group			Treatment Group				
Measure	Sample Size	Mean	Standard Deviation	Sample Size	Mean	Standard Deviation	Treatment – Control Difference	Standardized Difference
Pre-CSA	453	3.38	5.17	430	3.21	7.40	-0.16	-0.02
Pre-TechCheck	450	7.33	4.63	422	7.09	6.59	-0.24	-0.04
Pre-LA	333	0.37	4.92	238	0.21	3.63	-0.16	-0.04

Table 5. Background characteristics				
Background Characteristics	Sample Size	% in Control group	Sample Size	% in Treatment group
Female	488	51.4%	464	50.2%
Limited English Proficiency	488	6.8%	464	11.6%
Individualized Education Plan	488	12.3%	464	11.6%
Free/reduced-price lunch	488	17.8%	464	17.9%

Table 6. Missing Data by Outcome and Condition								
	Control Group				Treatment Group			
	Clusters[a]	Students[b]			Clusters[a]	Students[b]		
Outcome Measure	# Analytic Sample	# Randomized	# Analytic Sample	Missing%	# Analytic Sample	# Randomized	# Analytic Sample	Missing%
Post-CSA	7/7	700	453	35.3%	6/6	564	430	23.8%
Post-TechCheck	7/7	700	450	35.7%	6/6	564	422	25.2%
Post-LA	7/7	539	333	38.2%	6/6	368	238	35.3%

Table 7. Descriptive Statistics and Sample Sizes of the Baseline and Outcome Variables by Condition in Study A						
Measures	Control			Treatment		
	n	Mean	SD	n	Mean	SD

Pre-CSA	453	3.35	2.05	430	3.14	2.35
Post-CSA	453	5.36	3.26	430	11.34	6.51
Pre-TechCheck	450	7.35	2.41	422	7.12	2.47
Post-TechCheck	450	8.77	2.53	422	8.70	2.65
Pre-LA	333	0.10	1.07	238	-0.09	1.21
Post-LA	333	0.44	0.99	238	0.26	1.15

Table 8. Impact Analysis Results (Cluster-Level Assignment Study)

	Control Group				Treatment Group						
	Sample Size				Sample Size						
Outcome Measure	# clusters	# students	Mean	Standard Deviation	# clusters	# students	Model-adj. mean (comp+ beta)	Standard Deviation	Treatment – Control Difference	Standardized Difference	p-value
Post-CSA	7	453	5.44	11.06	6	430	11.63	14.46	6.19	0.47	<0.001

Post-TechCheck	7	450	8.67	7.00	6	422	8.92	3.33	0.25	0.04	0.18
Post-LA	7	333	0.42	4.92	6	238	238	0.49	-0.04	-0.01	0.86

Table 9.1. Additional Information for PostCSA and PreCSA with Missing Data in the Analytic Sample

	Control Group			Treatment Group		
Sample	# Individuals	Mean of Baseline Measure	Mean of Outcome Measure	# Individual s	Mean of Baseline Measure	Mean of Outcome Measure
Analytic sample (same as Tables 6 or 7)	453	3.35	5.36	430	3.14	11.33
Subsample of individuals with non-missing values for post-CSA and pre-CSA measures	453	3.35	5.36	430	3.14	11.33
Subsample of individuals with non-missing post-CSA measure and missing pre-CSA measure	2	Not applicable	1.65	0	Not applicable	
Subsample of individuals with non-missing pre-CSA measures and missing Post-CSA measure	27	3.45	Not applicable	24	2.98	Not applicable
Correlation between the baseline and outcome measures (calculated using only non-imputed data): _0.38____						

Table 9.2. Additional Information for Post-TechCheck and Pre-TechCheck with Missing Data in the Analytic Sample						
2nd Grade	Control Group			Treatment Group		
Sample	# Individuals	Mean of Baseline Measure	Mean of Outcome Measure	# Individual s	Mean of Baseline Measure	Mean of Outcome Measure
Analytic sample	450	7.35	8.77	422	7.11	8.70
Subsample of individuals with non-missing values for post-TechCheck and pre-TechCheck measures	450	7.35	8.77	422	7.11	8.70
Subsample of individuals with non-missing post-TechCheck measure and missing pre-TechCheck measure	4	Not applicable	8.5	4	Not applicable	7.25
Subsample of individuals with non-missing pre-TechCheck measures and missing Post-TechCheck measure	28	7.43	Not applicable	27	6.85	Not applicable
Correlation between the baseline and outcome measures (calculated using only non-imputed data): _0.51_____						

Table 9.3. Additional Information for Post-LA and Pre-LA with Missing Data in the Analytic Sample						
	Control Group			Treatment Group		
Sample	# Individuals	Mean of Baseline Measure	Mean of Outcome Measure	# Individuals	Mean of Baseline Measure	Mean of Outcome Measure
Analytic sample (same as Tables 6 or 7)	333	0.1	0.44	238	-0.09	0.26
Subsample of individuals with non-missing values for post-LA and pre-LA measures	333	0.1	0.44	238	-0.09	0.26
Subsample of individuals with non-missing post-LA measure and missing pre-LA measure	41	Not applicable	-0.10	17	Not applicable	-0.06
Subsample of individuals with non-missing pre-LA measures and missing Post-LA measure	5	-0.75	Not applicable	3	-0.62	Not applicable
Correlation between the baseline and outcome measures (calculated using only non-imputed data): ___0.81___						

Table 10. Table Illustrating the Scoring that Defines Adequate Implementation of Each Key Component in a Program Logic Model				
Indicator	**Unit of measurement**	**Indicator Scoring at Unit Level**	**Indicator Scoring at School Level**	**Indicator Scoring at Sample Level**
Key Component 1. Revised CAL curriculum				
(1) Curriculum dissemination	Teacher	0 (low) = don't have it ever 1 (high) = have at some point in curriculum	School-level: 0 = < 25% teachers with score of "1" 1 = 26 – 50% teachers with score of "1" 2 = 51-75% of teachers with score of "1" 3 = 76-90% teachers with score of "1" 4 > 90% teachers with score of "1" Threshold for fidelity = score of 3	All teachers completing training and teaching curriculum
Key Component 2. DevTech Training				
(1) Group training participation	Teacher	0 (low) = attended 25% or less of training 1 (low-medium) = attended 26% to 50% of training	School-level: 0 = < 25% teachers with score of "3" or more	Sample level: 0 =< 25% schools with score of "3" 1 = 26–50% schools with score of "3"

		2 (medium) = attended 51% to 75% of training 3 (high-medium) = attended 76% to 90% of training 4 (high) > 90% of training	1 = 26 – 50% teachers with score of "3" or more 2 = 51-75% of teachers with score of "3" or more 3 = 76-90% teachers with score of "3" or more 4 > 90% teachers with score of "3" or more Threshold for fidelity = score of 3	2 = 51-75% schools with score of "3" 3 = 76-90% schools with score of "3" 4 >90% schools with score of "3" Threshold for fidelity = score of 3
(2) Group training content	Sample			0 (low) = covered 25% or less of topics 1 (low-medium) = covered 26% to 50% of topics 2 (medium) = covered 51% to 75% of topics 3 (high-medium) = covered 76% to 90% of topics 4 (high) > 90% of topics Threshold for fidelity = 3
Key Component 2 Total Score		**Adequate implementation**	**School-level:**	**Sample level: Range: 0-8**

DevTech Training		**at teacher level = score of "3"**	**0 = < 25% teachers with score of "3" or more 1 = 26 – 50% teachers with score of "3" or more 2 = 51-75% of teachers with score of "3" or more 3 = 76-90% teachers with score of "3" or more 4 > 90% teachers with score of "3" or more Threshold for fidelity = score of 3**	**Threshold for fidelity = score of 6**
Key Component 3. Coaching				
(1) Embedded onsite coaching - availability	Teacher	0 (low) = teacher did not receive a response to request in first or in second half of the curriculum 1 (medium) = teacher received a response to request in either first or second half, but not both		

		2 (high) = teacher received a response to all requests made N/A: teacher did not request onsite coaching		
(2) Embedded onsite coaching - satisfaction	Teacher	0 (low) = Likert scale 1 or 2 (needs improvement) 1 (medium) = Likert scale 3 (“meets expectations”) 2 (high) = Likert scale 4 or 5 (exceeds expectations) N/A: teacher did not request onsite coaching		
(3) Virtual coaching - availability	Teacher	0 (low) = teacher did not receive a response to request in first or in second half of the curriculum 1 (medium) = teacher received a response to request in either first or second half, but not both 2 (high) = teacher received a		

		response to all requests made N/A: teacher did not request onsite coaching		
(4) Virtual coaching - satisfaction	Teacher	0 (low) = Likert scale 1 or 2 (needs improvement) 1 (medium) = Likert scale 3 ("meets expectations") 2 (high) = Likert scale 4 or 5 (exceeds expectations) N/A: teacher did not request onsite coaching		
Key Component 3 Total Score		**Teacher level: adequate implementation with score of at least 3 (if one of the coachings is N/A) or 6). If no training is accessed, the threshold is N/A.**	**School-level:** **0 = < 25% teachers with score of "3" (one coaching type accessed) or "6" (two coaching types accessed) or more (excluding N/As)** **1 = 26 – 50% teachers with score of "3" or "6" or more (excluding N/As)** **2 = 51-75% of teachers with**	**Sample level:** **0 =< 25% schools with score of "3"** **1 = 26–50% schools with score of "3"** **2 = 51-75% schools with score of "3"** **3 = 76-90% schools with score of "3"** **4 > 90% schools with score of "3"** **Threshold for fidelity = score of 3**

			score of “3” or “6” or more (excluding N/As) **3 = 76-90% teachers with score of “3” or “6” or more (excluding N/As)** **8> 90% teachers with score of “3” or “6” or more (excluding N/As)** **Threshold for fidelity = score of 3**	

Table 11. Findings on Fidelity of Implementation by Component for RI Schools						
Key Components, Number of Indicators, Units, and Threshold				**Year 1 Results** ***(2020-21 School Year)***		
Key Component	**Total # of Measurable Indicators**	**Unit of Implementation**	**Sample-Level Threshold for Fidelity of Implementation**	**Number of Units in Which Component was Implemented**	**Number of Units in Which Fidelity of Component was Measured**	**Achieved Fidelity Score and Whether Program Met Sample-Level Threshold**
1.Revised CAL curriculum	1	Teacher	Adequate implementation at teacher level = score of "1"	1 program 24 teachers 7 schools	1 program 24 teachers 7 schools	Score is 2 ***Program fidelity = No***
2.DevTech Training	2	1 teacher-level indicator 1 program-level indicator	Adequate implementation at teacher level = score of "3"	1 program 24 teachers	1 program 24 teachers	Score is 4 ***Program fidelity = Yes***
3.Coaching	4	Teacher	Adequate implementation at teacher level = score of "3"	24 teachers	8 teachers	Score is 9 ***Program fidelity = Yes***