Impact Study of the Coding as Another Language Curriculum: Study B

June 2023

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

Parastu Dubash, Marina Bers

Abstract

The aim of this study was to explore how the Coding as Another Language using ScratchJr (CAL-ScratchJr) curriculum, developed by Boston College's DevTech Research Group utilizing the ScratchJr app, impacted second grade students' computational thinking, coding skills, and reading comprehension. To accomplish this, the research team randomly assigned 20 schools in a school district located in a northeastern state of the United States to teach the Coding as Another Language curriculum or to a "business as usual" control condition. As a result, ten schools were assigned to the treatment group and the remaining ten schools were assigned to the control group. Participants in this study, referred to as Impact Study B, initially included 13 teachers and 247 students from 17 classrooms in the treatment group, and 10 teachers and 103 students from 12 classrooms in the control group. Hierarchical linear modeling was used to assess the impact of the CAL-ScratchJr curriculum on these second grade students' computational thinking, coding skills, and reading comprehension. Results showed that the CAL-ScratchJr curriculum intervention had a significantly positive impact on students' coding performance while no notable difference was found on students' computational thinking as both groups showed significantly higher increases of computational thinking from the baseline. Additionally, an examination of students' standardized literacy achievement across the two conditions found no notable difference findings, suggesting that even though the treatment group students allocated regular class time for the CAL-ScratchJr curriculum, the students in the treatment group showed comparable growth with the students in the control group on standardized literacy achievement assessments.

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 computer science (CS) related professions.

While most of the CS education implementation is taking place at the late elementary, middle school, high school, and college levels (Guzdial, 2008; Wilson, Sudol, Stephenson & Stehlik, 2010), educational frameworks, standards, and best practices position CS instruction 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 to start CS education 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 in a child's education (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).

If CS 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). Since the need to support the STEM pipeline is a weak rationale for the introduction of CS in early childhood education, CS 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 CS 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-ScratchJr addresses the teaching of computational thinking through both unplugged activities and on-screen coding with ScratchJr, when 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, to meet 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-ScratchJr 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 use 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 CS 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 CS 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).

In response to the pedagogical needs, DevTech Research Group developed the CAL-ScratchJr curriculum through design-based research, iterating curriculum development, field testing, and curriculum revision (Bers et al., 2023). In this process, many considerations were included such as expert review of materials for content and selection of books and songs to reflect diversity. Additionally teachers' feedback was considered such as the need for socio emotional learning on top of integrating the teaching of coding with literacy. This 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. In this study, we examine the CAL-ScratchJr curriculum's impacts on students' coding skills, computational thinking, and literacy performance.

Study Description

Research Questions for the study

Research Question 1: What is the impact of three months of CAL-ScratchJr curriculum on second grade students' computational thinking compared to the business-as-usual condition?

Research Question 2: What is the impact of three months of CAL-ScratchJr curriculum on second grade students' coding skills compared to the business-as-usual condition?

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

Intervention Condition

The schools that were assigned to the treatment group implemented the CAL-ScratchJr curriculum to their students during School Year (SY) 2021-2022. Students were intended to receive a total of 24 lessons of 45 minutes each. 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.

Although the current study focuses on the impact examination during SY 2021-2022, which used cluster randomized control trials that included both treatment and control groups, the CAL-ScratchJr curriculum was implemented using a delayed treatment design. That is, schools in the control group will implement the CAL-ScratchJr curriculum in SY 2022-2023.

Teachers in the treatment schools were trained in February 2022 in the delivery of the CAL-ScratchJr curriculum and provided resources and support to implement the curriculum during a 12-week period (Spring 2022). Teachers in the control schools delivered business-as-usual during 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 B held during SY 2021-2022, the DevTech Research Group trained the teachers delivering the curriculum in CAL-ScratchJr intervention schools.

The CAL-ScratchJr 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).

Program Intervention

Before implementation of the CAL-ScratchJr curriculum, 13 second- grade teachers in 10 CAL schools received professional development (PD) training on the implementation of the ScratchJr app in the classroom and the CAL-ScratchJr curriculum. This included attending two 2-hour PD workshops (total of 4 hours), completing their own ScratchJr project, and exploring lessons in CAL-ScratchJr curriculum.

Students were intended to receive a total of 24 CAL-ScratchJr curriculum lessons of 45 minutes each. Teachers implemented the CAL-ScratchJr curriculum lessons during regular school hours in between regular curricula. For Impact Study B, classroom teachers or supporting enrichment teachers, implemented the curriculum to the whole class in a classroom setting.

In Impact Study B, 13 second-grade teachers (84.6% female; average number of years teaching = 13.3 years) delivered the CAL-ScratchJr curriculum to 323 students; among these students, 247 consented to participate in the study. These students were enrolled in the treatment schools as of March 2022 when the implementation started. The average number of CAL-ScratchJr lessons completed by the end of implementation was 12.9. The total number of CAL-ScratchJr 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 the intervention's implementation was assessed for the intervention's three key components (CAL-ScratchJr curriculum, DevTech Research Group teacher training, and coaching). Seven indicators of implementation were measured to assess the extent to which the three key components were implemented (curriculum dissemination, group training participation, group training content, embedded onsite coaching–availability, embedded onsite coaching–satisfaction, virtual coaching–availability, and virtual coaching– satisfaction) (see table 10).

Setting

This multi-site study took place at 20 different schools within one school district in a northeastern state in the United States. The school district is public, high-poverty, and in an urban environment. The CAL-ScratchJr curriculum was deployed to child participants within the classroom context by teachers (who had been trained to implement the CAL-ScratchJr curriculum) at variable rates across the length of the intervention. Methods for randomization were standardized.

Control Condition

The schools that were assigned to the control group received no intervention of the CAL-ScratchJr curriculum during SY 2021-2022. Instead, these control schools did business-as-usual (e.g., taught literacy in the literacy block) and the teachers from the control group did not have access to training or the curriculum during SY 2021-2022. The control group schools delayed implementation of the CAl-ScratchJr implementation until SY 2022-2023, which is beyond the scope of the study.

Study Participants

Central office staff of the school district invited kindergarten, first, and second grade teachers to submit their interest to participate during the period from June 2021 to January 2022. Principals from the schools where these participating teachers worked were then invited to approve the participation. After receiving both the teachers' consents and the school principals' approval, 30 schools were identified. Due to limited resources and challenges during COVID, research effort was focused on the second grade. For this study, we excluded 10 schools that did not have 2nd grade participation. Among the remaining 20 schools with 2nd grade participation, two strata were created: a) four schools with participation from three grade levels, including kindergarten, 1st, and 2nd grade; b) 16 schools with participation from at least one 2nd grade classroom but not all the three grade levels. At the beginning of February 2022 prior to outcome testing, random assignments were done within each stratum. As a result, 10 schools were assigned to the treatment group and 10 to the control group from the two strata. Students who enrolled in the schools after random assignment are not included in the impact analysis. Figure 1.1 to 1.3 show the CONSORT flow of participants from the 20 schools by each outcome.

After agreement of participation from teachers and school principals, students' parental consent forms were sent to be collected and coordinated by a site coordinator. Parents from the treatment and control groups were required to sign and return a form if they consented for their child to participate in the study. Parental consents were received after the random assignment and before outcome measures were collected. The CAL-ScratchJr curriculum intervention was for the treatment group and the control group did business-as-usual. In the treatment schools, all students in the teacher's class would receive the curriculum, but only those who had parent consent would participate in data collection as part of the research study.

Figure 1.1 CONSORT Flow of 2nd Grade Participants for the CSA Outcome

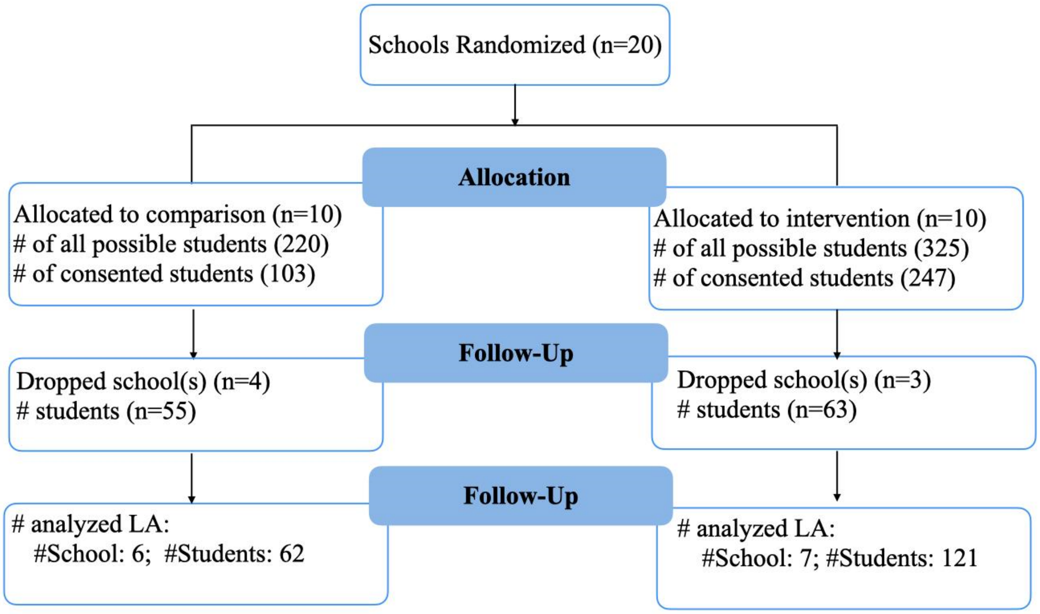


Figure 1.2 CONSORT Flow of 2nd Grade Participants for the TechCheck Outcome

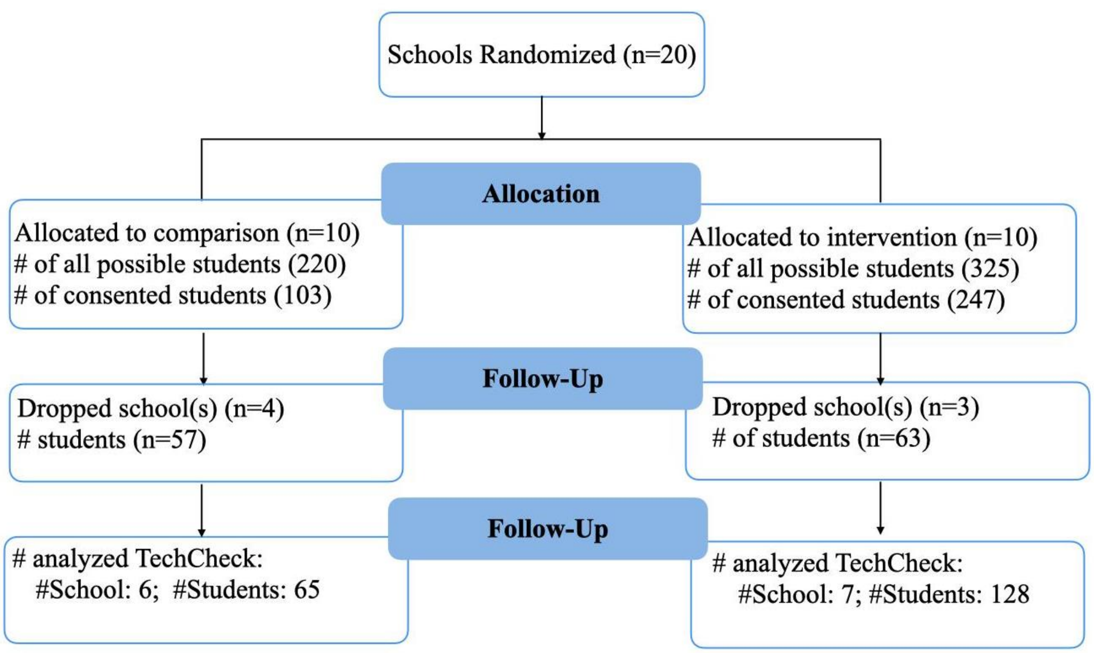
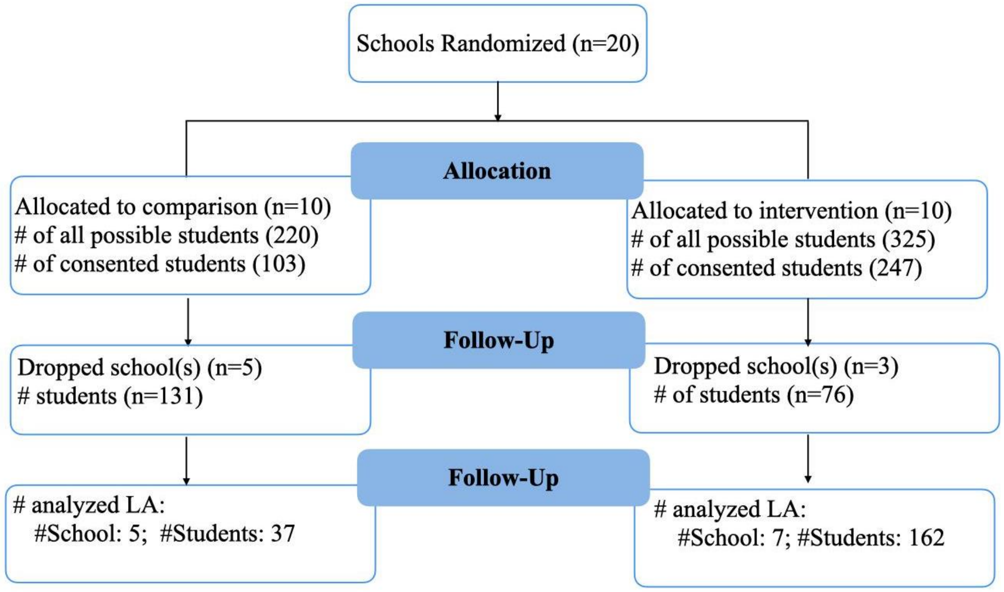


Figure 1.3 CONSORT Flow of 2nd Grade Participants for the Literacy (LA) Outcome



Sample Alignment with Those Served by the Program

The evaluation sample for the creative coding proficiency as measured by *Coding Stages Assessment* (CSA) and computational thinking as measured by *TechCheck* includes all consented second grade 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 the post-implementation literacy, as measured by MAP Reading Fluency in the Spring, only included students who took the assessment on or after May 23, 2022, which was an estimated date of post-implementation and was a starting date for post-CSA and post-*TechCheck* data collection in Study B.

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.

In order to ensure independence, a dedicated Assessment Team consisting of research assistants was employed to evaluate the two outcomes (CSA and TechCheck). These research assistants were recruited separately and independently from the DevTech Research Group. Assessment Team members were not aware of the experimental design or any other aspects of the research, including knowledge of which schools, teachers or students had been assigned to control or treatment conditions. The hiring process was overseen by the Shaffer Evaluation Group, in collaboration with school coordinators, to guarantee an unbiased selection. By employing this approach, we aimed to establish a robust and impartial assessment process.

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 B focuses on the second grade although the study design registration was updated on Friday February 24, 2023 to expand the study to include kindergarten and first grade in Study A.

Design

During Impact Study B, 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 (CAL-ScratchJr group) and the control group. Student outcomes were measured before and after the implementation of the curriculum/business-as-usual. CAL-ScratchJr group students were compared using two-level hierarchical linear modeling (HLM), controlling for covariates at the student 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 (CAL-ScratchJr and Control groups)	Score before start of curriculum (winter)	Raw score
Creative Coding Knowledge	Coding Stages Assessment	n/a	Right after completion of curriculum (CAL-ScratchJr and Control groups)	Score before start of curriculum (winter)	Raw score
Literacy	MAP Reading Fluency	Adaptive Oral Reading subtest	Included students who took the assessment on or after May 23, 2022 (spring MAP Reading Fluency)	Status of passing or fail on the Oral Reading Fluency Assessment before start of curriculum (winter)	Status of passing or fail on the Oral Reading Fluency Assessment

TechCheck

TechCheck assesses students' computational thinking (Relkin et al., 2020). This measure was developed by the intervention developer and validated with 769 five-to-nine-year-old students. In the current study, reliability was $\alpha = 0.60$ in the baseline and $\alpha = 0.62$ in the post assessment. TechCheck was correlated with TACTIC-KIBO ($r = .53$). This measure consists of 15 questions that are in a multiple-choice format, resulting in a*TechCheck* score that ranges from 0 to 15. Data for the current study was collected March 2022 - June 2022. Example items include: "What comes next" after showing a series of shapes. Please see more details about this instrument on the DevTech Research Group website. To ensure the early readers can understand the question, narrators read the questions to the student synchronously via Zoom by sharing a window with the TechCheck questions.

Coding Stages Assessment (CSA)

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 (de Ruiter & Bers, 2021). This assessment has 30 items presented in five stages. According to de Ruiter and Bers (2021), the total CSA score, which ranged from 0 to 39, was calculated as a weighted score of the five stages, with higher weights assigned to more advanced stages. Data for the current study was collected March 2022 - June 2022. Example items include: "Can you show me where you tap on the screen to make the program go?." Please see more details about this instrument on the DevTech Research Group website. In the current study, the internal consistency was λ_6 = .77 for the baseline coding assessment and λ_6 = .68 for the post coding assessment.

Trained research assistants administered the assessment via Zoom synchronously by sharing a window with visual prompts for each question in the assessment and narrating each question to the participating students. Participating students then coded in the ScratchJr app on their own device and presented their ScratchJr coding via Zoom with verbal explanations (when applicable). The research assistants rated students' responses to each question either as "Satisfactory" or "Unsatisfactory." Because the students have support using ScratchJr, all students in both treatment and control groups were able to start the ScratchJr app and continue the coding assessment.

All research assistants completed a systematic CSA training; after passing the CSA training test at the end of the training, qualified research assistants then took reliability checks for CSA. The research assistants demonstrated a great inter-rater reliability; specifically, a minimum Cohen's Kappa of 0.82 with an experienced rater was set as the standard to proceed for administering the CSA to the research participants in the study.

Literacy

The MAP Reading Fluency measures students' oral reading fluency, foundational skills, and literal comprehension using an adaptive benchmark. The participating school district has been using this assessment to measure their students' literacy achievement. For this study, the Adaptive Oral Reading subtest of MAP Reading Fluency was used since it is a common assessment type assigned to the majority of students. Students can be assessed in the Fall, Winter, and Spring. The Winter and Spring data were utilized as the pre and post curriculum implementation data for the current study.

Sample Sizes and Attrition

Study B sample size information is reported in Table 2, both for clusters and individuals, by condition and for each outcome. Outcome for coding skill is labeled as post-CSA, computational thinking as post-*TechCheck*, and standardized MAP Reading Fluency literacy performance as post-LA. The number randomized is the number of students and schools at enrollment, respectively. The number of schools and students in the analytic sample are the number of non-missing data at both the pre and the post stage for each outcome.

Reasons for school attrition are summarized below. For the CSA and TechCheck in the treatment condition, there were three cases of school attrition: one resulted from informed withdrawal from a teacher who was the only teacher in the school before collecting parental consents and two were due to incomplete data collection at the post stage. In the control condition, for the CSA and *TechCheck* outcomes, there were four cases of school attrition: two resulted from informed withdrawal prior to collecting parental consents, one was due to a non-responsive teacher participant who was the only teacher in the school, and one was due to incomplete data collection at pre- and/or post-stages. For the literacy outcome, in the treatment group, there were three cases of school attrition and, in the control group, there were five cases of school attrition. Except for the three informed withdrawals mentioned above, most of the school attrition for the literacy outcome was due to the non-mandatory nature of the assessment.

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 may be defined as high at the cluster level and the student level. The attrition was due primarily to the parent consent rate (76.3%), not being able to collect data because of challenges during the COVID-19 pandemic and limited resources, and the district's policy that the literacy assessment was not mandatory.

Data Analysis and Findings

Baseline Equivalence

The baseline mean difference between the treatment and control groups was calculated using a multi-level model that adjusts for clustering by having students at level 1 and schools at level 2 and included the baseline measures as the outcome variable. The analytic sample for each outcome differs slightly due to differences in response rates across outcomes.

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 as measured by passing or fail in the Adaptive Oral Reading subtest of MAP Reading Fluency in Impact Study B. The standardized difference (Hedges' g) between control and treatment groups on *TechCheck* is lower than 0.05 and therefore satisfies the WWC standard for baseline equivalence (What Works Clearinghouse, 2022). The standardized difference (Hedges' g) between control and treatment groups on CSA is 0.07, which requires statistical adjustment to satisfy the baseline equivalence standard. The standardized difference (Cox index) between control and treatment groups on the literacy assessment is 0.20, which requires statistical adjustment to satisfy the baseline equivalence standard. Nevertheless, all three outcomes included baseline measures respectively in examining the treatment effect on the outcomes.

Background characteristics beyond those 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 disadvantaged social-economic status (Yes or No).

Program Effects

Approach to estimating program effects:

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**

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

- **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 control.
- $\gamma_{02.q}$= 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^2_{00}.
- 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.

All covariates in the model were included. One exception is the gender variable and the school-level averaged baseline CSA score when predicting the CSA outcome. Due to the limited variability and the small sample size, these two variables were not included as covariates in the impact model that predicts the CSA outcome.

The impact model for the dichotomous literacy outcome used a linear probability model with the adjustment of all the above-mentioned student level and school level covariates.

No participants or units were excluded from the analysis except for the ones that are missing data (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-ScratchJr curriculum on second-grade student participants' computational thinking, the treatment group showed no notable difference compared to the business-as-usual control group (Hedge's g = -0.09). 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-ScratchJr curriculum on second-grade student participants' coding skills, the treatment group showed significantly higher coding skills compared to the business-as-usual control group (Hedge's g = 0.39). The Hedge's g effect size of 0.39 indicated a medium effect size 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-ScratchJr curriculum on 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 control group (Cox index = -0.19). Table 8 presents sample sizes, means, and the beta coefficients for the grouping variable (treatment or control) by condition for the dichotomous literacy outcome. Consistent with the WWC standards (2022), Cox index was computed as the effect size measure for this dichotomous outcome. Table 8 presents the mean (average probability of passing) in the control group with the adjustment of school cluster effect and the mean of the treatment group by adding the regression coefficient in the full model with adjustment of covariates and cluster effect from school.

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

The CAL-ScratchJr curriculum showed a positive impact on students' coding performance in the CAL group compared to the control group. Unlike the coding performance, the treatment group did not show a notable difference in students' computational thinking when compared to the control group although both groups showed significantly higher computational thinking scores compared to baseline. 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 Oluk's study to other studies. Additionally, code.org activities include the overly aligned and abstract unplugged computational thinking practices, which made the curricula not comparable with other curricula that do not explicitly teach computational thinking such as codeSpark Academy or the CAL-ScratchJr 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.

No significant difference was found regarding literacy performance when the CAL group was compared to the control group. One should note that because literacy was not a mandatory assessment, the sample size is small, especially in the control group. While there are uncertainties associated with findings from a small sample size, the results did not show a negative impact on student's literacy associated with CAL-ScratchJr curriculum implementation.

In summary, the CAL-ScratchJr showed a positive impact on students' coding performance. While no significant impact was found on students' computational thinking compared to the control group, 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-ScratchJr 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-ScratchJr 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-ScratchJr 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-ScratchJr curriculum.

The DevTech Research Group plans to expand the program to more schools, including engaging additional schools in CAL-ScratchJr professional development training and providing support for implementation of the CAL-ScratchJr 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-ScratchJr curriculum implementation was assessed by examining three key components: (1) revised CAL-ScratchJr curriculum, (2) DevTech Research Group teacher training, and (3) teacher coaching. In Impact Study B, the revised CAL-ScratchJr curriculum component encompassed one indicator: curriculum dissemination (measured by completing lesson log entry). The DevTech training component encompassed two indicators: group training participation (measured by an attendance sheet) and group training content (measured by a topic list checklist). The coaching component encompassed four indicators: onsite coaching availability, onsite coaching satisfaction, virtual coaching availability, and virtual coaching satisfaction (all measured by teacher self-report survey). The scoring model for assessing fidelity of implementation is presented in Table 10.

Fidelity Findings

Overall, Impact Study B achieved fidelity for some of the three components. In key component one, curriculum dissemination, 10 teachers (out of 13 total) accessed the curriculum, or completed a lesson log entry, and therefore earned a score of ‘1’ at the teacher-level. At the school-level, five schools had 90% or more teachers earn a score of ‘1’, while there was one school 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 71.4% 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, 12 teachers (out of 13 total) participated in the provided group training. This included attending training in synchronous or asynchronous format. The non-participating teacher was due to withdrawal from the study right after the random assignment. 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 ‘6’ at the program level indicating adequate implementation.

For coaching, the third key component, five teachers out of the 13 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 2.5 (on a rating scale of 1 = needs a lot of improvement - 5 = couldn't be better) with virtual coaching. Of the three teachers who requested an onsite coach, all three received a response to their request. There was an average satisfaction of 4.0 with onsite coaching. Overall, this component earned a sum score of '7' indicating adequate implementation.

Overall, two out of three components were implemented with fidelity. 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. (Ed.). (2021). *Teaching Computational Thinking and Coding to Young Children*. IGI Global. http://doi:10.4018/978-1-7998-7308-2

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.

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. 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).

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

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

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.

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.

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.

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.

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

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 (within analytic cluster)	# Analytic Sample	# Randomized	# Analytic Sample	# Randomized (within analytic cluster)	# Analytic Sample
Post-CSA	10	6	220 (165)	62	10	7	325 (262)	121
Post-TechCheck	10	6	220 (163)	65	10	7	325 (262)	128
Post-LA	10	5	220 (89)	37	10	7	325 (249)	162

[a] Reported the number of students in non-attrited clusters only, for cluster-assignment evaluations.

Table 3. Attrition Assessment for Impact Study B									
	Clusters			Students			Student		
Outcome Measure	Overall Attrition	Differential Attrition	Optimistic Threshold	Overall Attrition	Differential Attrition	Optimistic Threshold	Overall Attrition Within Analytic Cluster	Differential Attrition Within Analytic Cluster	Optimistic Threshold Within Analytic Cluster
Post-CSA	35%	10%	high	66.2%	3.8%	high	56.9%	9.0%	high
Post-TechCheck	35%	10%	high	64.4%	4.1%	high	54.4%	9.4%	high
Post-LA	40%	20%	high	63.5%	13.3%	high	41.1%	23.5%	high

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	62	6.07	10.00	121	6.79	9.46	0.72	0.07 (Hedge'g)
Pre-TechCheck	65	7.06	6.35	128	7.27	6.04	0.21	0.03 (Hedge's g)
Pre-LA	37	0.52	-	162	0.60	-	0.08	0.20 (Cox Index)

Table 5. Background characteristics		
Background Characteristics	% in Control group (n=103)	% in Treatment group (n=247)
Female	53.4%	53.8%
Limited English Proficiency	26.2%	23.1%
Individualized Education Plan	16.5%	14.6%
Free/reduced-price lunch	61.2%	53.4%

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	6	220	62	71.8%	7	325	121	62.8%
Post-TechCheck	6	220	65	74.1%	7	325	128	66.2%
Post-LA	5	220	37	83.2%	7	325	162	50.1%

Table 7. Descriptive Statistics and Sample Sizes of the Baseline and Outcome Variables by Condition in Study B

Measures	Control			Treatment		
	n	Mean	SD	n	Mean	SD
Pre-CSA	62	6.07	4.66	121	6.84	4.59
Post-CSA	62	8.97	5.70	121	13.69	7.29
Pre-TechCheck	65	7.23	2.48	128	7.69	2.76
Post-TechCheck	65	8.35	2.31	128	8.36	2.95
Pre-LA	37	0.57	-	162	0.70	-
Post-LA	37	0.73	-	162	0.81	-

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	6	62	8.90	8.90	7	121	12.66	9.91	3.77	0.39 (Hedge's g)	0.001
Post-TechCheck	5	65	8.31	4.27	7	128	7.87	5.24	-0.44	-0.09 (Hedge's g)	0.38
Post-LA	5	37	0.68	-	7	162	0.61	-	-0.07	–0.19 (Cox Index)	0.51

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	# Individuals	Mean of Baseline Measure	Mean of Outcome Measure
Analytic sample (same as Tables 6 or 7)	62	6.07	8.97	121	6.84	13.69
Subsample of individuals with non-missing values for post-CSA and pre-CSA measures	62	6.07	8.97	121	6.84	13.69
Subsample of individuals with non-missing post-CSA measure and missing pre-CSA measure	4	Not applicable	21.08	8	Not applicable	13.23
Subsample of individuals with non-missing pre-CSA measures and missing Post-CSA measure	5	3.30	Not applicable	89	5.70	Not applicable
Correlation between the baseline and outcome measures (calculated using only non-imputed data): _0.59____						

Table 9.2. Additional Information for Post-TechCheck and Pre-TechCheck 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	65	7.23	8.35	128	7.69	8.36
Subsample of individuals with non-missing values for post-TechCheck and pre-TechCheck measures	65	7.23	8.35	128	7.69	8.36
Subsample of individuals with non-missing post-TechCheck measure and missing pre-TechCheck measure	8	Not applicable	9.88	11	Not applicable	7.55
Subsample of individuals with non-missing pre-TechCheck measures and missing Post-TechCheck measure	5	6.60	Not applicable	78	6.62	Not applicable
Correlation between the baseline and outcome measures (calculated using only non-imputed data): _0.64_____						

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	37	0.57	0.73	162	0.70	0.81
Subsample of individuals with non-missing values for post-LA and pre-LA measures	37	0.57	0.73	162	0.70	0.81
Subsample of individuals with non-missing post-LA measure and missing pre-LA measure	2	Not applicable	1.0	19	Not applicable	0.73
Subsample of individuals with non-missing pre-LA measures and missing Post-LA measure	56	0.71	Not applicable	48	0.88	Not applicable
Correlation between the baseline and outcome measures (calculated using only non-imputed data): __0.69____						

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-ScratchJr 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 2 (medium) = attended 51% to 75% of training 3 (high-medium) = attended 76% to 90% of training 4 (high) > 90% of training	School-level: 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	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
(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 DevTech Training		**Adequate implementation at teacher level = score of "3"**	**School-level: 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**	**Sample level: Range: 0-8 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**	**School-level: 0 = < 25% teachers with score of "3" (one coaching type**	**Sample level: 0 =< 25% schools with score of "3"**

		the coachings is N/A) or 6). If no training is accessed, the threshold is N/A.	**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 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**	**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**

Table 11. Findings on Fidelity of Implementation by Component for Participating Schools						
Key Components, Number of Indicators, Units, and Threshold				**Year 1 Results** *(2021-22 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-ScratchJr curriculum	1	Teacher	Adequate implementation at teacher level = score of "1"	1 program 13 teachers 10 schools	1 program 13 teachers 10 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 13 teachers	1 program 13 teachers	Score is 6 ***Program fidelity = Yes***
3.Coaching	4	Teacher	Adequate implementation at teacher level = score of "3"	13 teachers	7 teachers	Score is 7 ***Program fidelity = Yes***