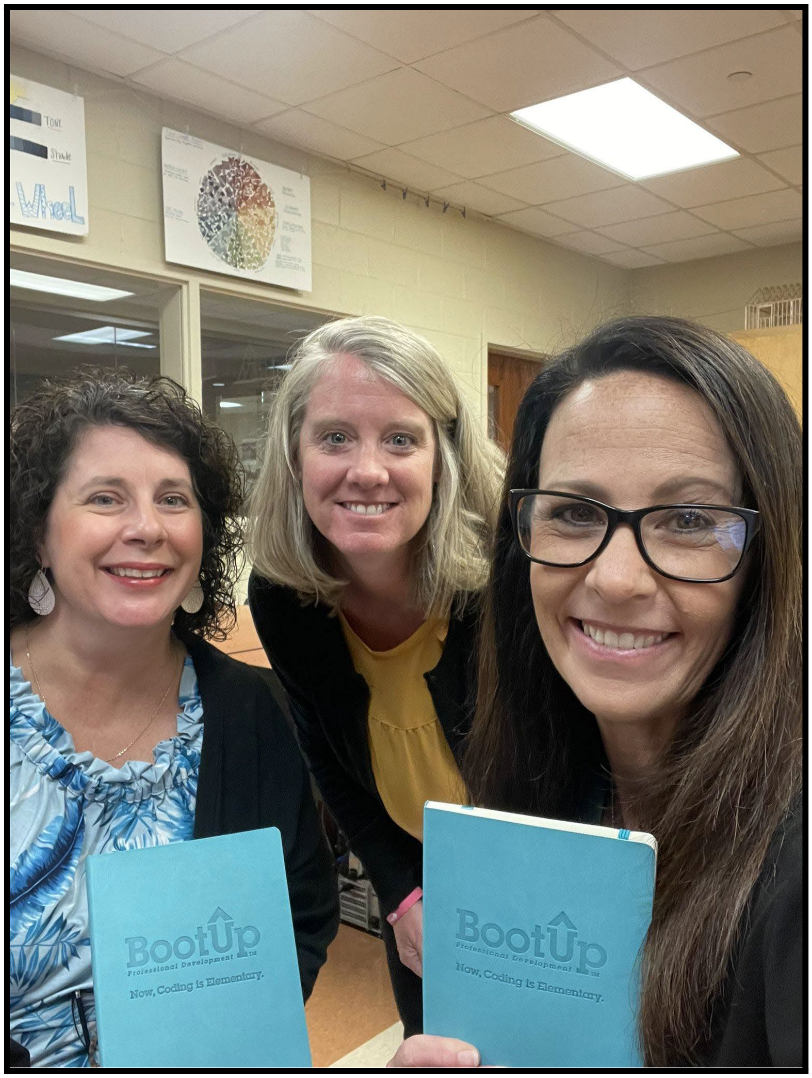
Evaluation Report 2022-23

EVALUATED BY
PETER RICH, PHD

This evaluation report includes the following:

BootUp PD 2022-2023
Teacher and PD Evaluation Report



Evaluation conducted by

Peter Jacob Rich, PhD

Table of Contents

"I teach a special education class and these lessons are so engaging for the students. Where they may struggle in reading and writing, they thrive in computer science."

Introduction

This report is an external evaluation of BootUp PD's efforts to train elementary coding teachers during the 2022-23 academic school year. Much of the funding for this work was provided by the Amazon Future Engineer (AFE) program. BootUp is a non-profit 501(c)(3) organization dedicated to empowering elementary teachers and students through computer science, especially in underrepresented and underserved communities (*cf* https://bootuppd.org/about). AFE's primary aim is likewise to increase access to computer science education for students from underserved and underrepresented communities.

Author Statement

This evaluation was conducted by Peter Rich, PhD. Dr. Rich studies and teaches elementary coding and has written over two dozen texts on this topic (*cf* https://scholar.google.com/citations?hl=en&user=JFv6vGMAAAAJ&view_op=list_works&sortby=pubdate). Dr. Rich collaborated with BootUp personnel to decide upon research questions and evaluation criteria. While Dr. Rich collaborated with BootUp on the direction of the research and evaluation, he was free to analyze and interpret the data without interference, influence, or persuasion from BootUp or any of its staff or funders (e.g., AFE). The data analyses, interpretations and recommendations in this report are therefore Dr. Rich's and may not represent BootUp's own interpretation of the data. These are provided as an external review of BootUp's efforts to train teachers to teach coding and therefore may be used for judging the effectiveness of BootUp methods on teachers and students.

Executive Summary

1. **Coding Confidence:** Teachers' confidence to teach coding increased 180% during their first year with BootUp PD.
    a. ***Most Confident***: Teachers showed the most confidence with sequences, loops, and algorithms as well as collaborating, communicating and fostering an inclusive computer culture.
    b. ***Least Confident***: Teachers were least confident with their knowledge of conditionals, abstractions, debugging, and creating computational artifacts.
2. **Excitement & Anxiety:** Teachers participating in BootUp PD increased their excitement for coding by 15% while their anxiety decreased by 30%.
3. **In the classroom:**
    a. Over 100,000 students were impacted by new BootUp-trained teachers in 2022-23
    b. On average, teachers integrated about 2 lessons/year in language arts, social science, science, and math.
    c. Most teachers taught coding 1x/week or less often.
    d. The overwhelming majority (80%) of teachers used BootUp curriculum with no or minor changes to teach coding.
    e. Teachers reported student excitement as their biggest success.
    f. Teachers reported time as their greatest challenge.
4. **BootUp Evaluation:**
    a. On average, teachers attended 4.38 BootUp PDs this year.
    b. Teachers were overwhelmingly positive in describing their experience with BootUp PD, rating the trainings as 9.2/10.
    c. Teachers' highest-rated BootUp component was the hands-on learning, with the model teaching coming in a close second. All components were rated highly (8+/10).

Recommendations

The overall findings demonstrate that BootUp's efforts to prepare elementary coding teachers were successful during the 2022-2023 school year. Teachers were overwhelmingly positive about BootUp's methods, especially appreciating the hands-on, student-centric lessons and on-site support. Notwithstanding, in the spirit of continued improvement, I make the following recommendations based on the evidence provided:

External: Recommendations for the coding Professional Development community writ large

1. *Schools should <u>require clear expectations</u> for how often teachers should teach coding*. There is great variance in how often teachers are engaging children with coding. With most teachers only teaching coding to the same children once or twice a month, they are not likely growing in their computational thinking abilities to the extent they otherwise could if it were a weekly part of the curriculum. Likewise, the more teachers engage with coding themselves, the more confident they'll become.
2. *Teachers could use a <u>second year of training</u> in order to develop greater confidence with computational concepts and practices*. While teachers grew in their confidence to teach coding in all areas, they are less sure of their abilities with conditional logic, variables, abstraction, creating and refining computational artifacts and debugging. Past evaluations demonstrated that teachers who participated in 2 years of BootUp training improved in these areas over their second year of training.

Internal: Recommendations for BootUp Professional Development specifically

3. *Encourage <u>higher survey participation/completion rates</u>*. With just over ⅓ of teachers completing the post assessment who completed the pre-assessment, it's possible that we are missing out on the experiences of a key demographic (i.e., teachers who did not finish the training).
4. *Engage teachers more in lessons that <u>make use of conditionals</u>* (if/then blocks). Conditional logic is a fundamental coding concept, but teachers report less confidence than they do with other fundamental coding concepts.

Methods

BootUp's PD model prepares teachers to teach fundamental coding concepts in pre-K though 8th grade classrooms (with a primary focus on elementary school teachers). This starts with an initial professional development session at the beginning of the school year, followed by additional PDs every other month throughout the school year. This provides teachers with the time needed to implement the learned methods in their classrooms and then bring back questions and experiences to subsequent trainings.

At the first of these and before receiving any training about how to code, a BootUp facilitator administers an initial "pre" survey. The purpose of this survey is to measure teachers' beliefs about coding. Four beliefs were measured:

1. value of learning to code,
2. teachers' self-efficacy for thinking computationally,
3. teachers' self-efficacy for coding, and
4. teachers' self-efficacy for teaching elementary coding.

These were measured on a 6-point Likert-type scale, ranging from "strongly disagree" (1) to "strongly agree" to about 30 different statements (Rich, Larsen, & Mason, 2020). This scale has been validated and used in multiple studies and contexts by scholars around the world. The survey takes about 10-15 minutes to complete. It is openly available at https://csedresearch.org/resources/evaluation-instruments/tool/?id=210.

At the final professional development session of the year, teachers took the same survey. In addition, they answered several questions about their actual experience teaching coding that year, as well as additional questions about their knowledge of computational concepts, practices, and perspectives that could not be asked before they had an understanding of what these were. The survey included Likert-type items as well as several open-ended questions.

Analysis

I used mixed methods to analyze teacher responses. Many trends are explained using descriptive statistics. Comparisons between groups were done using inferential statistics (each of which will be explained as presented). For example, to compare teacher change pre to post, I used a paired sample comparison of means. Responses to open-ended questions were analyzed using a cross-comparative open-coding methodology (Rich, 2012).

Participants

During 2022-2023 BootUp taught 988 teachers across 15 US states (see Appendix A). Some of these teachers then taught what they learned in BootUp to their colleagues, resulting in 1,746 overall teachers being impacted by BootUp's PD in 2022-23. It is estimated that this impacted 100,306 total students.

While this represents overall participation numbers in BootUp PD during 2022-203, not all teachers completed the surveys. Thus, the remaining analysis will report only on the data received through the pre and post teacher surveys. Table 1 shows data from the teachers that completed pre and post surveys for teacher training in 2022-23.

Table 1

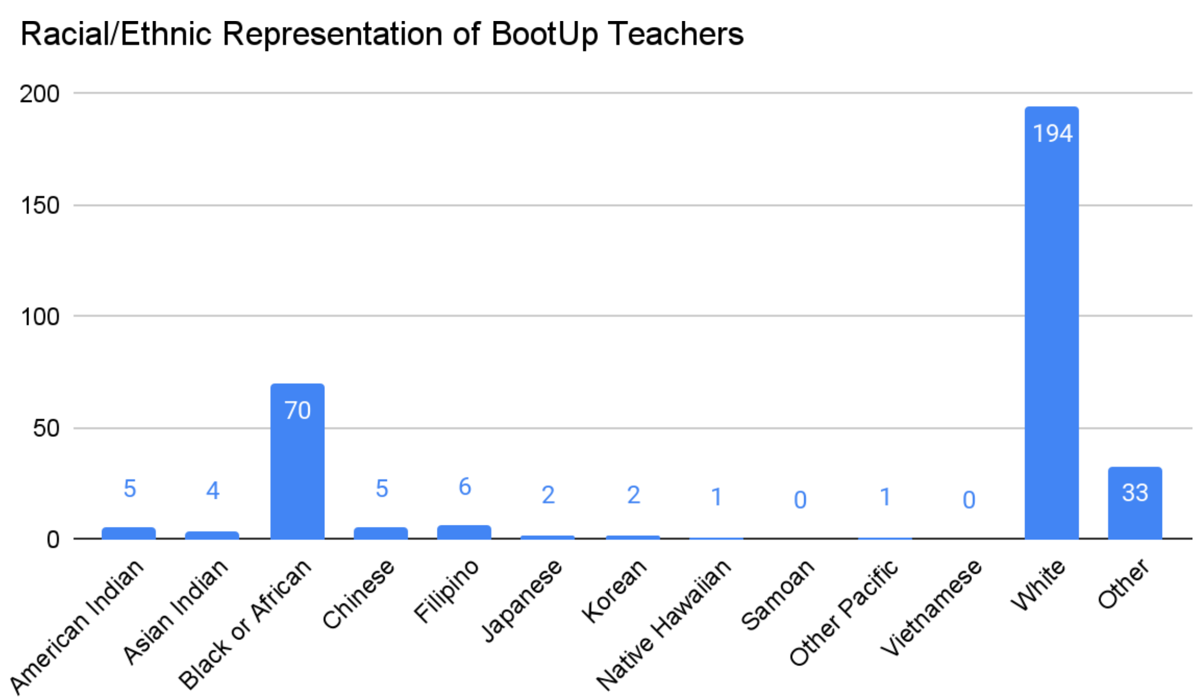
Participating Districts, # of Teacher Responses by District

District	TEACHER Pre	TEACHER Post	State	Schools
city-1	21	6	Alabama	18
suburb-2	13	8	Alabama	15
city-2	36	7	Pennsylvania	5
suburb-3	49	41	Illinois	49
town-1	14	0	Georgia	13
city-3	7	7	California	2
city-4	77	29	Texas	63
suburb-4	21	0	Alabama	14

District	TEACHER Pre	TEACHER Post	State	Schools
city-5	7	6	Georgia	5
private-1	4	2	Michigan	4
suburb-5	0	7	Iowa	1
city-6	6	0	Illinois	5
city-7	42	30	New York	15
city-8	15	13	New York	11
city-9	16	10	New York	15
mixed-1	41	20	New York	13
city-16	18	0	Georgia	6
rural-6	17	5	Texas	17
mixed-2	17	0	Iowa	3
suburb-10	20	12	Tennessee	11
city-10	22	0	California	12
town-2	14	8	Mississippi	12
rural-1	2	2	Iowa	1
city-11	14	0	Iowa	3
suburb-6	164	0	California	27
city-12	0	7	New York	7
rural-2	24	11	Tennessee	24
city-13	27	2	Iowa	4
mixed-3	23	14	Oklahoma	25
suburb-7	11	9	Arizona	8
city-14	12	12	Maryland	16
rural-3	29	0	Utah	5
suburb-8	1	0	Texas	1
suburb-9	30	24	California	3
mixed-4	8	0	Washington	3
rural-4	5	3	Georgia	2
city-15	18	16	Iowa	0
rural-5	29	16	New York	6
city-1	13	0	Iowa	1
TOTALS	**887**	**327**	**16**	**445**

Overall, the vast majority (86%) of teachers were female. The majority (60.2%) were White, while about ⅕ (21.6%) where Black or African American. Nearly a quarter (24%) identified as Hispanic or Latino. The majority (70%) were also classroom teachers, meaning that most teachers who completed the surveys this year were generalists, in charge of teaching a multitude of subjects. The remaining 30% of teachers were primarily STEM specialists, computing teachers, or media specialists. A small handful of participants were TAs, district coaches or volunteers. Most teachers had earned a Masters (57%) or Bachelors (38%) degree.

Figure 1



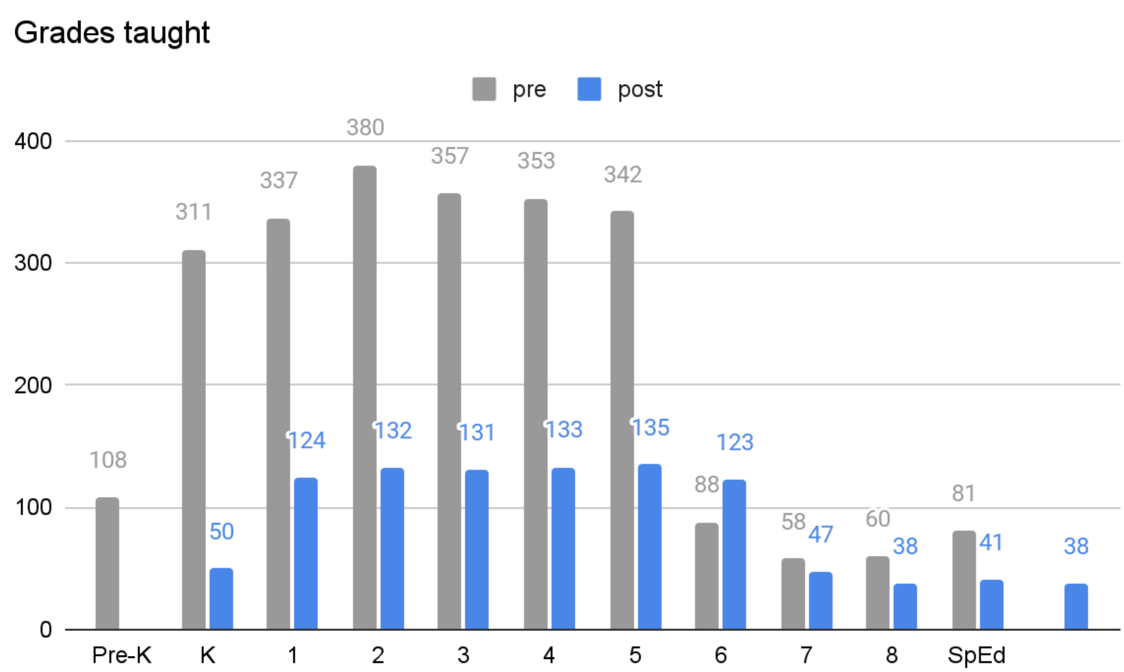
Participants averaged over 13 years of teaching experience, with over 11 of those teaching the same grade(s). Teachers reported that they had about 2.5 years of experience teaching coding. Thus, while coding was new to many teachers, on average, they had begun to teach it prior to their BootUp experience. While the bulk of participants were elementary teachers, a portion taught middle grades or special education (see Figure 2).

Teacher Practice

This section details the extent to which teachers actually implemented coding in their classes and what their experience was like when doing so.

Just over half (53%) of teachers taught coding once a month or less often (see Figure 3). About one fifth of participants reported teaching once a day or more often. These tend to be specialists (STEM, computer, or library media). Classroom teachers mostly taught coding once a week or less often. This matched the pattern of how often the same teachers saw the same

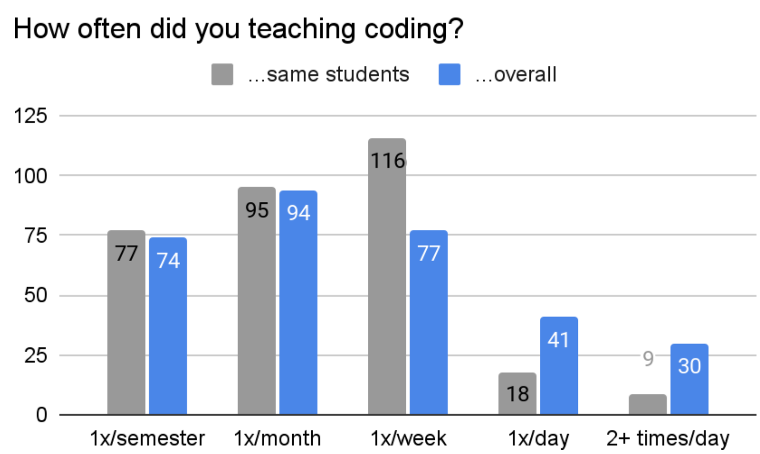
Figure 2



students. In Figure 3 below, the blue bars represent how often a single student would have experienced coding classes. Just under half (45%) of students experienced coding weekly.

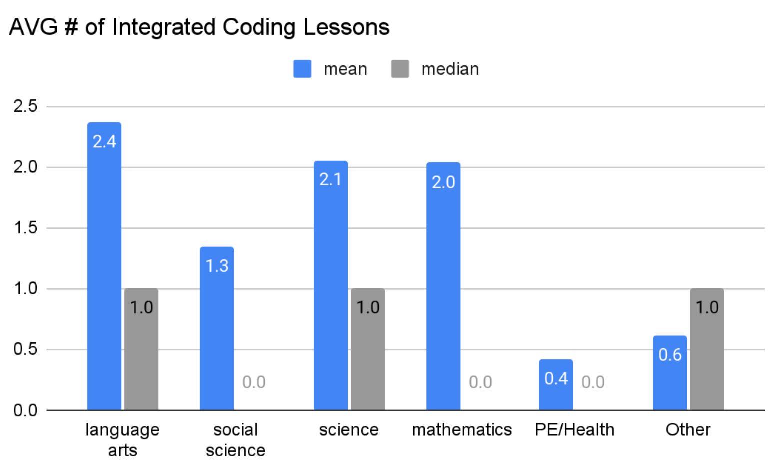
While the frequency of coding classes is important, so too is the duration of these lessons. Fortunately, the majority (56%) of lessons lasted at least half an hour. Some lessons reportedly lasted only 15 minutes. Further analysis of the data revealed that shorter lessons all appear to have been given by teachers who teach Pre-K through 3rd grade, where shorter lessons are expected due to younger children's attention spans.

Figure 3



Given that the majority of teachers were generalists, it would be reasonable to expect them to integrate coding with other subjects. Figure 4 demonstrates the average frequency of integrating coding into different subjects. The blue bars represent the mean, showing that the most common subjects with which teachers integrated coding were language arts, science and mathematics. The gray bars in Figure 4 represent the median, another way of showing the average. Some teachers are more prone to integrate many lessons, thereby pulling up the overall average.

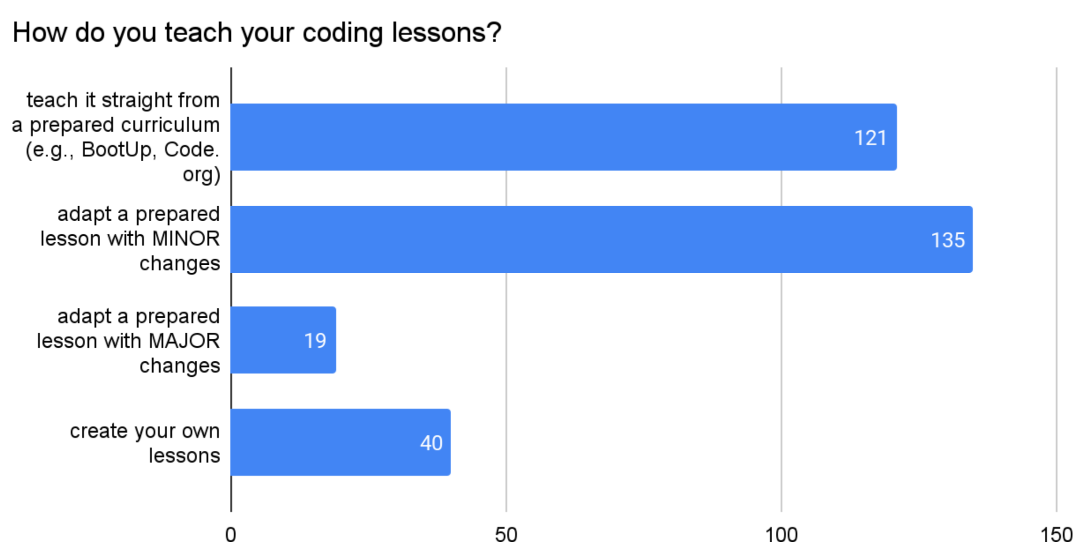
Figure 4



The median accounts for this outlier effect by weighting all the teachers' equally and showing the middle number. In this case, the median is likely more representative of the overall group of teachers. This reveals that, ***on average, teachers integrated coding into three lessons during the year, one in each of language arts, science and "other" subjects (e.g., health).***

Finally, another indicator of how teachers taught coding in the classroom is the materials they use. While the more mature coding teachers might create their own lessons from scratch, it can be difficult for teachers teaching a new subject to come up with curriculum completely on their own. The majority of teachers adopted the BootUp curriculum directly or with minor revisions. ***This demonstrates the importance not just of training teachers and providing professional development, but also of providing specific lesson materials.***

Figure 5



Challenges and Success of teaching Coding

Teachers shared their greatest successes and their biggest challenges in teaching coding in open-ended responses. To get a sense for general trends, I classified each response and then looked for patterns in teachers' responses. A single response could be coded in multiple categories. For example, the answer, "Students that have a hard time communicating verbally seem to excel at coding" was coded both as "student success" and "ability level."

Successes

While teachers gave 25 different types of answers, Table 2 shows those that were provided by at least a dozen teachers.

Table 2

Most Common Successes Teachers Experienced with Teaching Coding

Category	Definition	Representative Quote	Count
Student Interest	Comments about students' happy emotional state or desire associated with coding.	"My greatest success moment was when I realized that my students started spending their free time coding."	92
Student Engagement	Comments on how involved or focused students are when coding.	"Seeing students engaged in coding. They just don't want to stop when time is up."	64
Student Understanding	References students' cognitive state and increased comprehension for or due to coding.	"Students making connections between science concepts of pollination and coding. A recent lesson in 2nd grade was to have students code Beebots through a maze and pollinate flowers along the way."	56
Ability Level	Specific comments about students' "normal" ability level in comparison to how they unexpectedly perform while coding.	"Students who struggle academically, shine in coding."	27
Collaboration	Teacher mentions how students work with each other while coding.	"My students have been able to create amazing projects. The students have started helping each other with their coding and help debugging. "	21
Integration	References teacher efforts or student connections to incorporate coding with other subjects.	"I was able to integrate it very well in my classroom with concepts that are taught. I used letter, letter sound, shape and number recognition throughout the coding lessons to engage and reinforce the concepts. "	14

Student Success	References student achievement with or due to coding.	"We completed the entries for the Tech Fair. We had at least 10-15 kids."	14

All of the most common responses except for one focused on students. In fact, a third of teachers mentioned student interest as their greatest success, indicating that there is a lot of buy-in from children when it comes to computing. The only non-student-focused top success was one that centered on how the teacher or students were able to make connections between coding and other subjects.

These patterns were not a surprise. Since we first began collecting data on teachers' classroom experiences in teaching coding, they have put student interest, engagement, and success at the top of their list. However, one new category stuck out this year—ability level. This category highlights observations that teachers made about students succeeding that teachers would not have expected to excel at coding. While such comments have occurred in the past, I have never seen them so prevalently. Because one of BootUp's goals is to ensure that all children have equitable access to and positive experiences with coding, I share the entirety of the "ability level" comments in Table 3 below. In these comments, you can see how teachers were impressed that students in kindergarten, with autism, English language learners, or those who typically struggle in other subjects are doing well in coding. I've bolded the "ability level" reference in each quote and underlined those students' successes.

Table 3

Quotes about Students' Ability Levels in Regards to Coding

Students that have a **hard time communicating verbally** seem to excel at coding
I have a student who has **autism** and there were some skills in class he had a hard time learning, but he loves Scratch Jr. I was able to integrate the skills into Scratch Jr. lessons he could understand.
I teach a **special education** class and these lessons are so engaging for the students. Where they may **struggle in reading and writing**, they thrive in computer science.
Students have responded so well. **Students of all abilities** have been able to do some coding and they really like it.
Students who **struggle academically**, shine in coding.
I have **students that are below level in reading** that are able to successfully use coding blocks to create projects. These students are able to be leaders in the class when **usually they require the help**.

Kids as young as **Pre-K** can code! I would have never thought it! Students WANT to figure the solution out!

The children regardless of **language barriers or learning abilities** pick it up quickly and enjoy it

Coding has allowed access to all students including our **students with special needs**.

Students who are **not typically engaged** in classroom content have become interested in using coding and are more engaged and enjoy the creative aspect of it.

Students who are **academically low** have excelled when using coding. Students also like to create and explore coding on their own.

I was able to teach **kindergarteners** how to successfully code through scratch and with BeeBots.

Great success in helping our **gifted and talented** students be more innovative.

The engagement and collaboration has been a huge positive. My **youngest students** have loved coding and working with Bee-Bots and Scratch. The collaborative PLN of this cohort has been fantastic as well.

Students who are often **disengaged and unmotivated** enjoy the new experience of using Beebots in the classroom, they get excited. Students who sometimes **struggle with academics** can feel successful in their own way, from entering code one command at a time, to slowly building on to a level they feel comfortable and confident in.

My **low and diverse learners** who were **having trouble with letter and number recognition** the bee bots helped them learn their letters and numbers in a creative way.

Our **special ed students** seemed very engaged with the program. It was easy for **younger kindergarten** student to follow along

Witnessing students **(many ELLs) who are afraid to write** have fun creating projects in my **first grade classroom**.

The students are highly motivated during the coding/CT lessons, quite often the **students who lack motivation on a regular basis** are the ones that are most motivated.

Students who were shy, withdrawn, [or] speak other languages have come alive when they started coding. I have seen **students who have struggled socially, academically, [or] emotionally** find their way with coding because they found something they are good at. It's been amazing to see these students find their passion.

My **ENL and special needs Students** have learned to code in their own language. They get along and worked together to create something new. They enjoy game design and creating code. They understand how to troubleshoot and bounce ideas.

Due to the support that we get from the Boot-up team we have been successful with the implementation of coding/CT. I feel that having their support when jump starting the program and consistent feedback makes it less threatening. My students as **pre-k students** can actually do a lesson without my support because it is a part of our regular routine.

I believe children have an openness to coding! We start at the **kindergarten level** and even do physical computing with microbit. Children generally do well and feel a sense of accomplishment when finishing a course.

Most students and teachers enjoy trying to code or are at least interested in the concept. **Students you would least expect to excel in coding** thrive.

Getting kids excited to code. Watching other students take leadership roles **when they don't in other subject areas**.

Some of my **students who have no prior coding experience** have been very successful and I have seen them persevere to reach goals. Seeing their pride in themselves is amazing.

My students have been very creative and for **students who may not participate much during core academic lessons,** this provides those students with another way to participate and share their ideas with the class.

Challenges

While successes focused principally on students, challenges included time, materials, teachers and students, among others. Overall, teachers mentioned 23 different challenges. Table 4 provides a description of the challenges mentioned by at least a dozen teachers.

Table 4

Challenges in Teaching Coding

Category	Definition	Representative Quote	Count
time	Concerns about the time needed to prepare for or teach coding	"Managing the time and planning the activities."	82
teacher knowledge	Challenges with teachers' own lack of coding knowledge and how to plan lessons or answer students' questions.	"I had a hard time helping the students with some of the codes if they got stuck as I wasn't sure on next steps myself yet."	64
technical issues	Issues arising from technology, including access, supply, or whether or not it functions.	"Need more technology, better wifi, access to certain programs on DOE devices"	46
ability level	Difficulties in teaching students due to their actual or perceived abilities, including differentiating instruction for students at different levels.	"Students are at very different levels. Some students are adding extra coding to the challenges and other's ask to help identify letters on the keyboard."	23
none	Explicit statements that a teacher has not experienced any challenges in regards to teaching coding.	"Nothing really." "There weren't any challenges." "None"	16
student knowledge	Challenges with students' understanding of coding.	"The larger complexity of the curriculum go over their heads. Often times 1 lesson has to be broken down."	15
student attitude	References to students' emotional state when coding.	"Students just want the answers to code something and just want to play games rather than be the creators and try to problem solve. There seems to be a lack of motivation from students to complete their coding projects or only	

		do a limited amount of coding and say they're done or that is all that they want to do.”	
integration	Challenges arising from incorporating coding into the teaching of other subjects.	“We have very few coding elements currently implemented into our classroom curriculum. Hopefully this PD will allow us to adapt some of our assignments and projects to incorporate more into the classroom.”	13
student focus	Issues arising from students getting off-task or moving too far ahead of the lesson.	“Students struggle with focus on the lesson goal and spend time playing on scratch jr and drawing and coloring.”	13

Time is the most oft-cited challenge for teachers in many studies or efforts to incorporate something new into the curriculum (Rich, 2012). This year’s current BootUp experience was no different. Teachers’ concern for time covered several different aspects of time. In many cases, teachers indicated that they did not have enough time in an already-packed schedule. In other cases, they suggested that meeting with students once a week to teach coding was insufficient time for students to learn the material adequately. In other cases, teachers felt that they did not have enough planning time or time to learn the material on their own. One solution to this problem is for administrators to provide the structure for more regular teaching of coding. While all of these are undoubtedly important, two teachers demonstrate that the solution to time problems can extend beyond just giving more time. A specialist stated,

> “A challenge during this experience has been having teachers find time to incorporate the lessons into their instructional plans. Elementary teachers already have so much

> content to cover, being able to integrate content into coding lessons is the key to overcoming the time obstacle."

Likewise, a classroom teacher made the following observation,

> "My challenges consist of time management with needing to focus more on Science standards for testing purposes. The coding integration component will be beneficial to my learning environment in aiding me with meeting objectives/benchmarks."

In both cases, these teachers suggest that integration of coding with existing subject areas can actually help rather than hinder. This does not diminish the importance of a systemic approach to providing structured time for coding instruction. However, it does highlight that adding coding to the curriculum is not a zero-sum game but rather can actually enhance and improve teachers' existing approaches.

Finally, while there are many other challenges that teachers face in incorporating coding into their classrooms, it is noteworthy that "none" was one of the five most-common responses. This may demonstrate that, once a teacher addresses their challenges, it is possible for coding to be incorporated without major challenges.

Teacher Confidence

Nearly all teachers who participated in the BootUp trainings and completed the post-survey indicated that their confidence had increased. The few who did not mention an increase expressed that they were already confident, but often indicated that the trainings had provided them with new ideas on how to teach coding. Table 5 provides a representative quote from teachers at every level who participated in the BootUp PDs this year. This section examines changes in different aspects of teachers' confidence to each computing and presents evidence for what might have led to those changes.

Table 5

How has your confidence to teach computing changed over the course of this year?

Grades taught	Quote
Pre-K	I've gained confidence to share coding with others due to gained knowledge throughout the cohort.
K	My confidence has increased a lot. The hands-on learning and help from the facilitator has been the biggest impact.
1st	Practice, seeing how the kids use the materials with trial and error and try do adjust on their own helped me realize that I can pretty confidently adapt instruction while teaching simple coding in the same way that I adapt instruction for other content areas.
2nd	The PD classes have been very useful and hands-on, which is empowering. Thank you
3rd	I feel a lot more confident in my ability to teach and work with coding. This workshop has positively impacted my learning and excitement to use coding.
4th	I think I went from hating to loving it. I think this PD definitely helped this change.
5th	My confidence has increased tremendously! My anxiety has decreased. This class has made the process of learning coding anxiety-free and much easier than presented.
6th-8th	My confidence has increased because I was able to complete some lessons on my own, ask questions and exchange information with other teachers and work with them.
K-5	I am much more confident teaching coding. The PD sessions showed me how to approach this challenging work with my youngest students.
K-8	My confidence has definitely gone up since the beginning of the year. I am eager to continue teaching computing.

A primary goal of the BootUp experience is to help teachers to become more confident in their competence to teach coding. To that end, the pre-survey primarily consisted of questions from the TBaCCT survey, or the Teachers' Beliefs about Coding and Computational Thinking (Rich, Larsen, & Rich, 2020). The TBaCCT measures four main beliefs: the *value* of computing, teachers' *self-efficacy for computational thinking*, teachers' *self-efficacy for coding*, and teachers' *self-efficacy for teaching coding*. The TBaCCT has been validated with 1000s of teachers and used to measure changes in teacher confidence in multiple studies by diverse researchers (Jocius, et al., 2023; Love et al., 2022; Rich, Mason, & O'Leary, 2021; McCune, 2023; Tshukudu et al., 2022; Yadav et al., 2022).

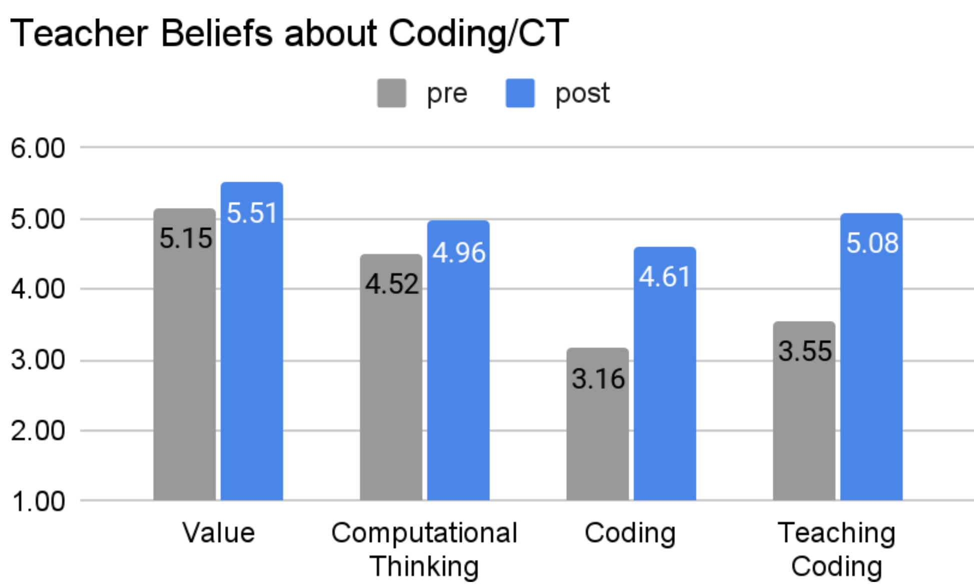
Teachers completed the TBaCCT during their first and last professional development (PD) experiences of the year (to be able to measure change). In addition to the TBaCCT questions, the post survey included questions about computational concepts, practices, and perspectives (Brennan & Resnick, 2012), a well-used framework for describing the type of knowledge, attitudes, and behaviors associated with coding and computational thinking. These questions were not asked on the post-survey because they include jargon that teachers who have not yet had any coding experience would likely have been unfamiliar with. These questions were asked on the same 6-point Likert-type scale as the TBaCCT so as to provide comparable interpretations.

Finally, the post-assessment also asked teachers about changes in their confidence to teach coding in open-ended questions, allowing teachers to express their thoughts in their own words.

Teachers' Beliefs about Coding and Computational Thinking

Overall 887 teachers completed the pre-survey, while 323 completed the post-survey. As demonstrated in Figure 6, teachers' initially began with a high valuation for the importance and utility of teaching coding (5.2/6.0). This is a positive sign, as it suggests that teachers were open to learning more about how to teach coding because of its value to their students.

Figure 6



Teachers also started BootUp trainings with positive self-efficacy for their own ability to think computationally. Questions about CT self-efficacy focused on a teacher's ability to think computationally without having to code. Teachers reported starting half-way between "somewhat agree" and "agree" on statements such as, "I am good at finding patterns," and "When confronted with a problem, I can break it down into smaller parts."

In contrast to value and CT beliefs, teachers started from a neutral or negative self-efficacy for coding and teaching coding. Questions about coding address teachers' knowledge of fundamental programming concepts, such as loops, variables, and conditionals. Questions about teaching coding, on the other hand, focus on a teacher's ability to help young children learn to code. For example, "I know where to find the resources to help students learn to code," and "I can help students debug their computer programs."

In all cases, teachers demonstrated statistically significant gains from pre to post. This means that the observed changes were not likely due to chance. I also measured effect size, which measures *practical* significance, or how much of a difference was observed.

Table 6

Changes in Teachers Beliefs about Coding and CT before and after BootUp PD

Self-efficacy (1-6 scale)	Value	CT	Coding	Teaching Coding
pre	5.20	4.53	3.23	3.63
post	5.52	4.96	4.61	5.08
t-test (statistical significance)	0.000*	0.000*	0.000*	0.000*
Hedge's g (effect size)	0.460	0.506	1.220	1.264

* results significant at a = .01

To measure effect size, I used Hedge's *g*. This is a measure that allows for comparison of two unequal-sized groups. A value between .2-.5 is considered a small effect, values between .5 and .8 are considered medium effects, and values greater than .8 are large effects. This means that teachers experienced a small practical change in their values (which makes sense, since they already started fairly high), a medium effect in their CT self-efficacy, and large effects in their coding self-efficacy and teaching-efficacy. These are notable changes that suggest that the overall training was effective in reaching its goal of increasing teacher confidence to code and teach coding.

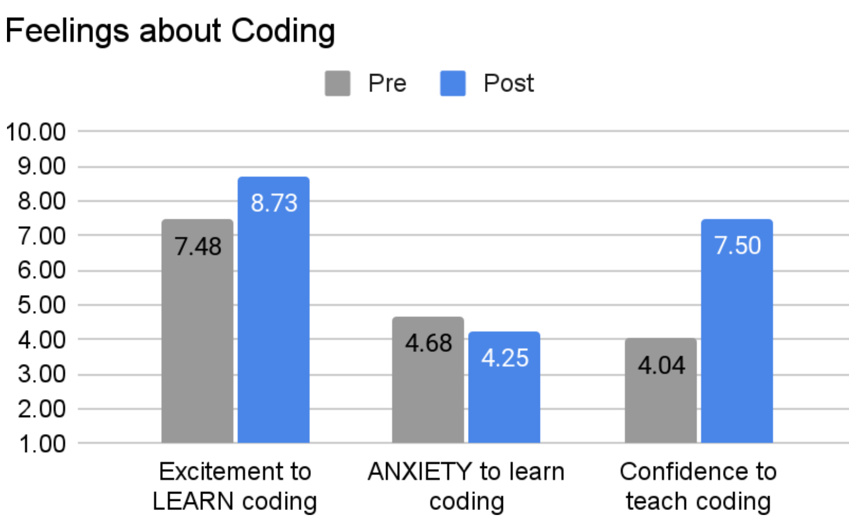
Feelings

To get a sense for how teachers feel about teaching coding, we asked three simple questions on a 10-point scale. What is their:

- *Excitement* to learn coding;
- *Anxiety* to learn coding; and
- *Confidence* to teach coding.

Figure 7 shows teachers' pre and post changes in each of these areas. As with beliefs, teachers demonstrated both statistically and practically significant change in each of these areas. The

Figure 7



change in *anxiety* was statistically significant (p = .020) but negligible (*g* = 0.148); the change in *excitement* (p < .000) showed a medium effect (*g* = 0.517); and the change in *confidence* was strong (*g* = 1.329).

Computational Concepts, Practices, and Perspectives

Brennan and Resnick's (2012) framework has been a popular way of assessing learners' knowledge, behaviors, and attitudes when engaging with coding projects. It was originally developed as a way to describe the activities teachers and researchers observed students engaging in while working on Scratch projects. It has since been used to inform a multitude of different frameworks, including the k12cs.org framework that provides the foundation for many state Computer Science standards. BootUp designed their curriculum around these concepts, practices and perspectives to ensure that they would meet such standards.

Table 7 shows teachers' confidence for each of these areas by the end of their first year of BootUp PD.

Table 7
Teachers' self-efficacy for Computational Concepts, Practices, and Perspectives

Coding Concepts	
sequence	4.73
loops	4.50
algorithms	4.53
functions	4.26
variables	4.20
conditionals	4.02
CT Knowledge	
pattern recognition	4.60
algorithms	4.45
decomposition	4.24
evaluation/analysis	4.29
abstraction	4.05
CT Practices	
collaborate around computing	4.54
communicate about computing	4.50
foster an inclusive computer culture	4.47
recognize and define computational problems	4.17
create computational artifacts	3.98
test and refine computational artifacts	3.93
develop and use abstractions	3.96
CT Perspectives	
Collaborating	5.02

Persistence	4.90
Creating	4.74
Tinkering/Remixing	4.59
Debugging	4.44

In general, teachers rated themselves in the "somewhat agree" to "agree" range on most of these measures. Teachers were most comfortable with CT perspectives, especially collaboration. They were least confident with their practices, especially creating, testing, and refining computational artifacts; and developing and using abstractions. Note that these practices are those that require creation, whereas the other practices require facilitation. This suggests that teachers need continued practice with creating. They are not likely to get to the second level of confidence (i.e., the 5.0-6.0 range) until they've had more practice creating, testing and debugging actual code projects.

Another interesting item to note in this framework is that teachers are least confident with conditionals (i.e., *if … then…*). As a programmer and teacher of coding myself, I find it curious that teachers would be less confident with conditionals than with variables or functions, both of which tend to be more abstract and difficult concepts to grasp and to teach. In contrast, Elementary teachers introduce conditional logic to children as soon as they enter school (e.g., "*If* you have a question, *then* raise your hand). Likewise, conditional logic is introduced even in some of the earliest coding lessons, such as those present in Code.org's "hour of code." This suggests to me that teachers may need more practice engaging with conditional logic early on and then throughout their BootUp training. Conditional logic is fundamental to coding and I would expect teachers to be more than "somewhat" confident with it after a year of training.

Teachers' Evaluation of The BootUp Experience

The final part of the teacher study reports on teachers' overall experience with BootUp. Regardless of whether they taught kindergarten or 5th grade or whether they were a specialist or a generalist, nearly all teachers indicated that their feelings about teaching coding changed due to BootUp. Table 8 shares a few representative quotes from teachers of different grades.

Table 8
Changes in Feelings about Teaching Coding due to BootUp

Grades taught	BEFORE the BootUp trainings	AFTER the BootUp trainings
Pre-K	I wanted to learn more about coding, but did not know how to implement it in the classroom.	I am very excited about using Beebots and students are too.
Pre-K	I did not think it would work with my blended preschool classroom.	I was totally wrong in fact the BeeBots was a great way to have my diverse learners focused. My group of students that were having some academic challenges they were able to enjoy learning their alphabets and numbers
K	The timing is really hard to make accessible.	How easily I can integrate it within my classroom.
1st	unsure, not confident	becoming more confident, especially with the two programs we explored in depth (Beebots, Scratch Jr). It's easier to envision ways to incorporate coding more into everyday activities in the classroom.
1st	disinterest, pessimistic that I could implement it in my class	Optimistic and finding ways to implement it has been easier than I thought
2nd	Fair- I have used Hour of Code for at least 5 years.	More Confident - I like the reverse engineering project.
2nd	I had no idea what I was doing, very unsure	much more comfortable with it!
3rd	Intimidating/confusing/counter-intuitive	Much more doable.
3rd	It is too difficult	It can be simplified for elementary

Table 8
Changes in Feelings about Teaching Coding due to BootUp

Grades taught	BEFORE the BootUp trainings	AFTER the BootUp trainings
4th	I did not think I would enjoy them.	I love it! I will be using this in my classroom for sure.
4th	Good and I knew how important coding was.	Amazing! Now I have more coding skills and know how to gain access to information for my students and have ideas for them to code.
5th	Questionable	Excited and Confident about the material
5th	hated it	love it
K-5	unorganized and random.	organized; I appreciate the lesson plans that provides a scope in sequence.
K-5	I was unsure about teaching coding to my students .	I am very confident about teaching coding to my students .
K-8	I teach coding, but I usually wait for the older grades. I thought it was too much for the littles.	I have lots of resources now thanks to Bootup, I just need more practice with where everything is located. Sometimes checking student work is time consuming, too.
PreK-5	Confusing	I feel empowered and ready to go!!

The remainder of this section explores teachers' experience with BootUp and their reason for rating the experience one way or the other. On average, teachers reported attending 4.4 BootUp trainings throughout the year. When comparing BootUp to other professional development experiences, teachers rated BootUp at 8.84/10.00. Overall, that's a fairly high rating. Breaking this down more reveals some interesting trends. Teachers provided a written rationale for their

Table 1.9. *Polarity Analysis*

Polarity	N	%	AVG. score
positive	216	83.72%	9.22
negative	17	6.59%	6.00
neutral	25	9.69%	7.52
TOTAL	**258**	**100.00%**	**8.84**

rating. I conducted a polarity analysis by classifying teachers' ratings as either positive, negative or neutral (see Table 9).

Positive comments referenced the benefit or quality of trainings. The vast majority of comments were quite positive, scoring BootUp over 9.2 points, on average. Over a third of the positive comments (n = 76) referenced the hands-on nature of BootUp trainings. Another 43 comments (20%) credited their rating to how engaging the PDs were, while another 20% indicated that the trainings' usefulness was their reason for giving it a positive rating. Comments about usefulness expressed appreciation for the fact that teachers could readily implement BootUp curricula in their classrooms. Comments from those who rated the trainings positive also focused on the resources available at the trainings, the ability to collaborate with their peers, and knowledgeable or helpful facilitators.

Those who provided negative rationales for their ratings rated the trainings over 3 points lower. To be clear, this was only a small portion of the overall population of teachers. However, it's helpful to understand what might have led to their displeasure in order to address such issues for future PD and work toward continual improvement. There were about a dozen different reasons teachers gave for rating BootUp PD lower. No single reason dominated all answers. Reasons repeated by multiple teachers included pacing, usefulness, technical problems, and modality (see Table 10). Other concerns involved the timing of when training was offered, resources, being confused or overwhelmed, and the general pedagogy.

Table 10
Teachers' reasons for rating BootUp PDs negatively

Reason	Definition	Sample Quote	Count
Usefulness	Concerns about the ability of the teacher to apply BootUp material directly to their classroom situation.	"I feel like we were not given things we could implement tomorrow. There was a lot of sharing with people in the cohort, but I didn't want to share, I wanted to get more ideas."	3
Pacing	Issues arising from how quickly the information was taught.	"I feel like we blew through a lot of the coding for the third session and then it was assumed we know it. I'm a monthly person so I'm not sure if we	3

Table 10
Teachers' reasons for rating BootUp PDs negatively

Reason	Definition	Sample Quote	Count
		were there for the actual training or not. The coder resources will help fill in the gaps I hope."	
Modality	Problems caused by the online delivery of BootUp training	"It's difficult to follow along on Zoom site while having multiple windows open on our computers. Navigating and keeping up with the the pace is hard. It's not possible to navigate multiple tabs without getting lost or missing something when you have only one screen. I do better on my own watching videos and building my projects on my own."	2
Technical Issues	Impediments caused by the functioning of or access to technology	"I was quite frustrated that scratch was heavily used and we are not able to use that program in CPS with our students."	2

Which BootUp Practices were most Effective?

BootUp Professional Development training consists of many different components. Each of these has been intentionally chosen and designed to improve the teacher learning experience in order to prepare them to be able to teach coding in their own classrooms without the aid of a "more knowledgeable other" (Cole, 1985). BootUp uses a project-based, scaffolded approach. Project-based means that all of the learning is focused on completing specific coding projects rather than smaller individual tasks or assessments. Scaffolded means that the support begins with lots of help. As teachers become more "stable" in an area, the support is gradually removed (and sometimes set up in a different, new area where they can construct their knowledge). While the goal is to cultivate teachers who can work independently, learning is a social endeavor. Teachers learn from the BootUp facilitators as well as from each other. BootUp facilitators are "more knowledgeable others" (MKOs) who help to maintain teachers in what is called the "Zone of Proximal Development," or that zone in which they can accomplish something with the aid of an MKO that which they couldn't on their own. Ideally, MKOs shift their instruction to accommodate the learner's current capabilities.

One more important element of BootUp' project-based model is a rhizomatic, interest-based approach. While all teachers work on the same project (e.g., introduce yourself to your students), the projects are designed to appeal to and allow teachers (and their students) to express their own interests. Thus, every project is like a fingerprint; it identifies the individual it came from. The rhizomatic aspect of this learning means that, like rhizomes, their interest and projects may spread out and make new connections. Each teacher's experience may ultimately be dynamic and different.

In order to support this social, project-based, rhizomatic approach to learning to code and to teach coding, BootUp professional development is comprised of a variety of different components. Teachers were asked to rate these on a scale of 1-10 (see Table 11).

Table 11

Teacher ratings of the different BootUp Components

BootUp Component	Definition	Rating
Hands-On Learning	Practice completing projects during PD	9.20
Model Teaching	BootUp Facilitators visit teachers in their classroom and demonstrate teaching a lesson to students.	9.15
Sharing Projects/Ideas	Time given during PD to discuss and share ideas with peers.	8.94
Coder Resource	Student materials, including video walk-throughs and debugging slides	8.90
Videos	Walk-throughs of completed projects and how to work out each section of a coding project.	8.86
On-Site Coaching	BootUp facilitators visit teacher at their school and offer formative feedback/support	8.79

Over the years, BootUp has refined its approach and methods to accomplish their teacher learning goals. As noted in Table 11, teachers rated nearly every component in the high range (8-10). Teachers were very happy with the hands-on learning approach, which confirms the analysis of their qualitative comments. They rated the model teaching nearly as high. Model teaching is when the BootUp facilitators and teachers role-played as teacher and students in their classrooms. This modeling enables teachers to visualize themselves teaching the same curriculum to their students.

Teachers were likewise pleased with the ability to share ideas and collaborate with their peers. This was commonly mentioned as a strength of BootUp in teachers' open-ended comments. This was exemplified by several teachers' comments, shared below:

> "I really appreciate the opportunities to review different lessons, to ask questions, to practice, and to discuss with fellow teachers."

> "Most PD experiences I have had so far have felt like a lecture. BootUp PD has allowed us hands-on learning experience and tons of collaboration which is how I learn best."

> "Loved working with Lauri and peers sharing. I took a lot of ideas back to my classroom."

> "Just like students, we learn best from hands-on collaborative learning. I was able to understand CS concepts and Scratch by doing the projects myself and having access to direct feedback."

> "It was hands on and I loved how we got to play with it and share experiences."

> "I was more invested since I had the materials already at hand. I loved all of the collaboration from all of the other teachers."

> "I loved the breakout rooms with my colleagues. Most of the time we didn't do the assignment, we instead shared ideas!!!"

> "I enjoy the hands on and the feedback and input not only from the facilitators but the participants as well."

To be clear, not all participants appreciated the collaborative nature of the PDs. One teacher expressed her frustration in the following way:

> "I feel like we were not given things we could implement tomorrow. There was a lot of sharing with people in the cohort, but I didn't want to share, I wanted to get more ideas."

Notwithstanding this teacher's statement, a large contingent of teachers expressed gratitude for being able to collaborate with their peers and gather ideas on how they might teach coding in their own classrooms.

Physical Computing

As part of their efforts to continually improve BootUp's approach and to ensure that they are teaching up-to-date tools, BootUp makes pedagogical changes each year. One of the changes they made in 2022-23 was to introduce a "physical computing" component using BeeBots or micro:bits. This took place with the teachers in Chicago, who had already participated in other non-BootUp PD over the past several years.

To identify mentions of the physical computing program, I looked at teachers' open-ended responses. This included questions about integration, successes, challenges, rationale for their BootUp rating and their feelings about teaching coding. In all, there were 39 responses from teachers in Chicago Public Schools. Of these, 16 different teachers mentioned physical computing. The majority of these referenced their experience with Beebots (other devices mentioned were micro:bits, Finches, and Hummingbirds).

Fifteen of the 16 teachers who mentioned physical computing did so in a positive manner. The one negative response was in regards to classroom management and the difficulty of assessing learning by using the Beebots.

Table 12
Physical Computing Mentions

Context	Sample Quote	Count
Integration	"It has been amazing!!!! I didn't think my preschoolers would be able to understand how to program the Bee Bots. The students after a couple of sessions were confident enough to help and teach other. Can't wait to use the Bots again next year."	4
Successes	"Students making connections between science concepts of pollination and coding. A recent lesson in 2nd grade was to have students code Beebots through a maze and pollinate flowers along the way."	9
Challenges	"physical computing - class management, assessment"	2*
BootUp rating Rationale	"It's been very effective to meet throughout the year with other teachers who are implementing coding and bee-bots with their students. Great discussion, sharing lessons, working through challenges, etc."	1
Before BootUp	N/A	0
After BootUp	"I was totally wrong in fact the bee bots was a great way to have my diverse learners focused. My group of students that were having some academic challenges they were able to enjoy learning their alphabets and numbers."	6

*the other reference to Beebots in challenges was actually used to counter the teacher's other challenge and was therefore painted as a positive success rather than as a challenge.

Curiously, though several teachers mentioned physical computing in comments about integration, none of these identified specific subjects in which they integrated the robots/micro:bits. However, teachers did identify several ways that they integrated physical computing throughout their other comments. Teachers demonstrated that they used physical computing in science, Spanish, ELA, mathematics, and fine arts. All teachers, including the one who mentioned the challenge of classroom management, indicated that students were highly engaged and motivated to work with robots. In several cases, teachers were surprised that their students were capable of working with robots due to their perceived ability levels (e.g., grade level, typical achievement). As one teacher stated,

> "I was totally wrong. In fact the BeeBots was a great way to have my diverse learners focused. My group of students that were having some academic challenges they were able to enjoy learning their alphabets and numbers"

In conclusion, it appears that the introduction of physical computing was highly successful. It enabled teachers to challenge and change their preconceived notions of what students are capable of; teachers were able to integrate physical computing across a wide range of subjects; students were highly engaged; and teachers were grateful for the fact that the devices were provided for classroom use and not just available during PD.

Despite the overall positive evaluation of this aspect of the program, there are two key limitations to this analysis. First, I'm relying only on the comments of the teachers who willingly mentioned physical computing. It is possible that those who didn't mention it did not have similarly positive experiences. Second, the evidence provided is merely anecdotal. Teachers did not provide measurable ratings of this component of BootUp PD as they did with other aspects of the program. Future evaluations will need to account for physical computing pedagogy.

References

Cole, M. (1985). Zone of proximal development: where culture and cognition create each other. In J.V. Wertsch (Ed), *Culture, Communication, and Cognition: Vygotskian Perspectives*, pp146-161. Cambridge University Press, New York.

Jocius, R., Albert, J., O'Byrne, W. I., Joshi, D., Robinson, R., & Blanton, M. (2023). Computational thinking infusion as transformative teaching: investigating content area teacher perspectives and practices. *Computer Science Education*, 1-30. https://www.tandfonline.com/doi/pdf/10.1080/08993408.2023.2210458

Love, T. S., Bartholomew, S. R., & Yauney, J. (2022). Examining changes in teachers' beliefs toward integrating computational thinking to teach literacy and math concepts in grades K-2. *Journal for STEM Education Research*, *5*(3), 380-401. https://link.springer.com/article/10.1007/s41979-022-00077-3

McCune, J. C. (2023). *Computational Concepts, Practices, and Perspectives in K-8 Computer Science Teaching Readiness*. Grand Canyon University]. https://search.proquest.com/openview/85eba3a9e9e12c645db1f0a9eafdedb2/1?pq-origsite=gscholar&cbl=18750&diss=y.

Rich, P. (2012). Inside the black box: revealing the process in applying a grounded theory analysis. *Qualitative Report*, *17*, 49. https://eric.ed.gov/?id=EJ981472

Rich, P. J., Larsen, R. A., & Mason, S. L. (2020). Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education*, 1-21. https://doi.org/10.1080/15391523.2020.1771232

Rich, P. J., Mason, S. L., & O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education*, *168*, 104196. https://www.sciencedirect.com/science/article/pii/S0360131521000737

Tshukudu, E., Waite, J., Rizvi, S., & Sentance, S. (2022). *Teachers' Motivations to Learn about ML and AI*. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 2* (pp. 609-609). https://dl.acm.org/doi/pdf/10.1145/3502717.3532148

Yadav, A., Israel, M., Bouck, E., Cobo, A., & Samuels, J. (2022). *Achieving CSforAll: Preparing Special Education Pre-service Teachers to Bring Computing to Students with Disabilities*. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1* (pp. 196-201). https://dl.acm.org/doi/pdf/10.1145/3478431.3499333

Appendix A. Overall Demographics

Table 13. *Overall Demographics for Participating Districts 2022-23*

District	Locale:	Total Schools	Number of Schools with school-wide Title I	Free and Reduced Lunch (%)	Male Students (%)	Female Students (%)	Black or African American Students (%)	Hispanic or Latino Students (%)	Students of color (%)	Teachers Participating	Additional Teachers Impacted	Total # of participating impacted teachers	Students Impacted	# Students Impacted from 2020-21 Teachers (estimate)	# Students Impacted from 2021-22 Teachers (estimate)	TOTAL NEW Students Impacted in 2022-23 (estimate)
city-2	City: Large	62	-	-	51%	49%	86%	11%	98%	159	11	170	15994	351	232	16577
city-3	18 City: Large; 1 Rural: Fringe	19	19	90%	51%	49%	28%	66%	95%	11	55	66	2157	750	1673	4580
city-4	City: Small	12	6	69%	52%	48%	28%	32%	67%	24	0	24	1944	599	378	2921
suburb-4	Suburb:Large	5	3	47%	52%	48%	38%	22%	71%	10	55	65	1683	0	567	2250
mixed-1	5 Suburb: Large 5 Rural: Fringe	10	2	58%	51%	49%	46%	21%	74%	12	0	12	840	0	1147	1987
suburb-5	Suburb:Large	3	-	-	50%	50%	77%	15%	98%	4	0	4	332	0	168	500
mixed-2	7 City: Mid-size; 2 Rural: Fringe; 1 Suburb: Large	10	7	-	51%	49%	36%	16%	60%	21	0	21	2636	718	0	3354
suburb-6	4 Suburb: Large; 1 Rural: Fringe	5	5	77%	50%	50%	16%	31%	57%	9	62	71	1246	0	1480	2726
city-11	11 Suburb: Large; 3 City: Large	14	-	87%	51%	49%	77%	6%	88%	210	0	210	8275	0	969	9244

Table 13. *Overall Demographics for Participating Districts 2022-23*

District	Locale:	Total Schools	Number of Schools with school-wide Title I	Free and Reduced Lunch (%)	Male Students (%)	Female Students (%)	Black or African American Students (%)	Hispanic or Latino Students (%)	Students of color (%)	Teachers Participating	Additional Teachers Impacted	Total # of participating impacted teachers	Students Impacted	# Students Impacted from 2020-21 Teachers (estimate)	# Students Impacted from 2021-22 Teachers (estimate)	TOTAL NEW Students Impacted in 2022-23 (estimate)
city-7	City: Large	4	-	47%	51%	49%	6%	16%	79%	17	98	115	2133	567	126	2826
mixed-3	3 City: Large; 2 Suburb: Mid-size; 1 Suburb: Larg	6	6	24% (DC)	51%	49%	8%	73%	89%	11	13	24	700	0	587	1287
suburb-7	1 Rural: Fringe; 31 Suburb: Large	32	25	74%	51%	49%	44%	47%	97%	42	117	159	3410	137	189	3736
city-12	City: Large	10	10	56% (DC)	50%	50%	80%	13%	98%	10	0	10	4026	156	223	4405
suburb-9	Suburb: Large	3	3	75%	51%	49%	21%	31%	88%	8	0	8	960	0	530	1490
mixed-4	1 Suburb: Large; 2 Rural; Fringe	3	3	89%	54%	46%	8%	4%	20%	5	120	125	1388	0	205	1593
city-15	City: Small	5	5	40%	51%	49%	11%	56%	77%	8	0	8	1021	0	1343	2364
mixed-5	2 Rural: Fringe; 1 Town: Fringe; 1 City: Midsize	4	-	91%	52%	48%	0.2%	92%	94%	0	0	0	581	0	194	775
suburb-3	Suburb-Large	10	10	37% (DC)	51%	49%	70%	21%	99%	11	0	11	5465	0	0	5465

Table 13. *Overall Demographics for Participating Districts 2022-23*

District	Locale:	Total Schools	Number of Schools with school-wide Title I	Free and Reduced Lunch (%)	Male Students (%)	Female Students (%)	Black or African American Students (%)	Hispanic or Latino Students (%)	Students of color (%)	Teachers Participating	Additional Teachers Impacted	Total # of participating impacted teachers	Students Impacted	# Students Impacted from 2020-21 Teachers (estimate)	# Students Impacted from 2021-22 Teachers (estimate)	TOTAL NEW Students Impacted in 2022-23 (estimate)
suburb-1	Suburb-Large	5	5	81%	50%	50%	77%	17%	97%	11	0	11	1720	0	0	1720
city-1	City-Midsize	27	27	81%	61%	39%	86%	11%	98%	27	0	27	2190	0	0	2190
city-5	City:Large	7	7	80%	50%	50%	66%	19%	85%	8	0	8	1535	0	0	1535
city-16	2 Rural: Fringe 16 City: Mid-size	18	18	60%	51%	49%	4%	74%	79%	24	83	107	5110	0	0	5110
city-10	City: Mid-size	16	16	100%	51%	49%	93%	3%	97%	146	0	146	5993	0	0	5993
city-13	3 Suburb: Large; 29 City: Large	32	32	92%	52%	48%	19%	57%	88%	32	0	32	13851	0	0	13851
suburb-2	Suburb: Large	4	3	99%	51%	49%	87%	8%	98%	72	0	72	2518	0	0	2518
city-6	City: Large	13	13	81%	51%	49%	2%	73%	91%	44	0	44	7606	0	0	7606
city-7	City: Large	12	12	75%	52%	48%	10%	44%	89%	12	35	47	2505	0	0	2505
suburb-8	Suburb: Large	4	-	87%	50%	50	6%	89%	99%	33	102	135	2351	0	0	2351
town-1	Town: Distant	2	-	51%	51%	49%	1%	63%	72%	7	7	14	136	0	0	136

Table 13. *Overall Demographics for Participating Districts 2022-23*

District	Locale:	Total Schools	Number of Schools with school-wide Title I	Free and Reduced Lunch (%)	Male Students (%)	Female Students (%)	Black or African American Students (%)	Hispanic or Latino Students (%)	Students of color (%)	Teachers Participating	Additional Teachers Impacted	Total # of participating impacted teachers	Students Impacted	# Students Impacted from 2020-21 Teachers (estimate)	# Students Impacted from 2021-22 Teachers (estimate)	TOTAL NEW Students Impacted in 2022-23 (estimate)
OVERALL Districts (29)	161 City: Large; 68 City: Midsize; 17 City: Small; 92 Suburb: Large; 16 Rural: Fringe; 1 Town: Fringe; 2 Town: Distant	357	237	52%	52%	48%	32%	47%	88%	988	758	1746	100306	3278	10010	113594

BootUp PD 2022-2023

Student Knowledge and Perceptions of Coding

Evaluation conducted by

Peter Jacob Rich, PhD

Table of Contents

Executive Summary

Over 100,000 students were affected by BootUp's efforts in 2022-23. This study reports on the efforts of a handful of schools that collected student data to create a baseline for understanding elementary students' cognitive abilities and attitudes toward coding in BootUp classrooms. A high-level summary is provided in the following summary:

1. **Participation**
    a. Three districts administered the Computational Thinking test (CTt), while four districts gave the Elementary Student Coding Attitude Survey (ESCAS).
    b. Overall, 737 students completed the CTt (341 girls, 381 boys, 15 other). Most responses came from 3rd and 4th graders. Nearly two thirds (65.9%) of students identified as Black or African American. Another 17% identified as Hispanic, with only 8.8% of students identifying as White. Thus, the population for the ESCAS study is one made up in large part of students who are traditionally under-represented in computing.
    c. In total, 589 students completed the ESCAS (273 boys, 311 girls). Responses came from 237 3rd graders, 185 4th graders, and 164 5th graders.
    d. The majority of students have less than a year of coding experience, though just under a third (29.2%) reported 1-2 years' experience with coding.
2. **Performance**
    a. Students completed the CTt in under 30 minutes, with 4th graders finishing 5 minutes faster than 3rd graders. It took students about 12 minutes to complete the ESCAS.
    b. On average, 3rd grade students scored 9/28 points on the CTt. The average score tended to increase by 1 point per grade.
    c. On average, boys scored 1 point higher than girls in the same grade on the CTt. There was no noticeable difference between genders on the ESCAS.

d. There is a negative, but weak, correlation (r ~ -.232) between grade and attitude. This means that the older students are less positive toward coding than the younger students.

Recommendations

Inasmuch as the purpose of this study was to establish a baseline for student knowledge of and attitudes for coding/CT, the most important recommendation is to ensure that students at these schools:

(a) continue to participate in classes that integrate coding, and

(b) complete these same post assessments at the end of the 2024 school year. That data will enable us to see their growth.

Methods

Prior to the 2022 school year, no BootUp-aligned school districts were collecting either cognitive or affective data from students. In order to measure a longer-term effect, we first needed to establish a baseline for knowing where students are starting from. Thus, this year's data does not reveal the results of prolonged exposure to coding education, but rather provides the first point from which students will later be measured to indicate change. In this section, I first describe the measures used, the analysis methods, and the baseline findings.

Measures

We used the Computational Thinking test (CTt) and the Elementary Student Coding Attitude Survey (ESCAS) to measure students' cognitive abilities and affective leanings for coding. I describe each of these measures below so that the reader might have a better understanding of the types of questions students were asked.

The Computational Thinking test

The CTt was originally developed in Spain with the purpose of creating a test that can measure students' CT ability from elementary to high school (Román-González, et al., 2016). It was developed and tested on 5th-10th grade children and found to have good reliability. It scales well across grades, meaning that as students grow, they get progressively better at the test. The CTt has demonstrated convergent validity with other well-established problem-solving instruments, such as the Primary Mental Abilities (PMA) battery and the RP30 problem-solving test. While demonstrating convergence with other problem-solving tests, the CTt validation procedures also revealed that solving computational problems is different enough from general problem-solving that it is a related, but separate, type of thinking.

According to the authors, the CTt is highly focused on "computational concepts." Computational concepts tested include loops, conditionals and functions. These are tested in seven tetrads, totalling 28 questions overall. Each tetrad presents the student with a series of four related tasks, moving from easier to more difficult tasks. Computational tasks involve either sequencing, debugging, or completion. *Sequencing* tasks pose problems of ordering computational instructions. *Debugging* tasks focus on finding and resolving errors in a set of codes. And *Completion* tasks focus on adding in missing code. CTt problems do not require students to know a specific coding language in order to solve and can thus be completed even by students who have never coded previously. On average, it takes students about 30 minutes to complete the full 28-item test.

Figure 1

Sample item from the CTt

M. Román-González et al. / Computers in Human Behavior xxx (2016) 1–14

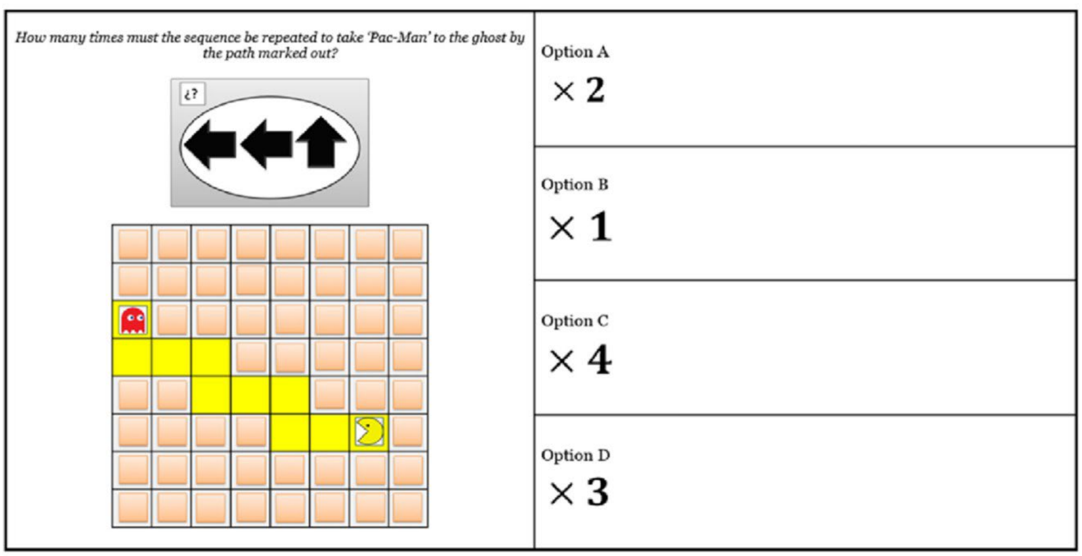


Fig. 1. CTt, item 6: loops—repeat times; 'The Maze'; visual arrows; no-nesting; completion.

The Elementary Student Coding Attitude Survey

While it's important to measure cognitive abilities affected by learning to code, it is equally important to measure affect, or attitudes. If students learn to like coding while young, they will be more likely to engage with coding as they grow older. Unfortunately, most coding attitude measures were either directed at older students (e.g., high school, college) or were too generic to be useful (e.g., "I like STEM"). Thus, Mason and Rich (2020) developed the ESCAS as a measure that could be used specifically with elementary-aged children. The ESCAS is written in simple language that elementary students can understand. It takes roughly 8-15 minutes to complete, on average.

The ESCAS consists of 23 Likert-type questions. Likert-type questions provide a statement and ask respondents the degree to which they agree or disagree with that statement. For example, for the statement, "Coding is Interesting" students can strongly disagree, disagree, somewhat disagree, somewhat agree, agree, or strongly agree. The ESCAS measures six key attitudes that have been shown to affect a student's likelihood to pursue coding. Namely: confidence, interest, usefulness, math confidence, perceptions of coders, and social value. *Confidence* measures their own confidence with coding. *Interest* measures how much they might like or be curious about coding. *Usefulness* gets at the practical aspects of coding, such as its value for employment, school or solving problems. *Math confidence* is a student's confidence with their math ability, which correlates with success in coding. *Perceptions of Coders* seeks to reveal a student's biases about what they believe a "person who codes" is like. Finally, "social value" reveals how a student believes their peers and parents view coding.

Procedures and Analysis Methods

In order to administer these surveys, we first sought Institutional Research Board (IRB) approval by each district. BootUp facilitators then worked with school teachers in districts that granted approval to distribute and collect parental permission forms. Only after a sufficient number of forms were collected were teachers allowed to administer the surveys. Thus, while the intent was that these would establish a baseline, different districts collected student data at different points of the year, depending on when they were able to secure all the appropriate permissions. Students did not provide their name or other personally-identifying information on these measures, but instead were asked to provide their school/lunch ID #. We will use this number in future studies to pair students and measure individual change over time.

Both the CTt and the ESCAS provide only quantitative data. To analyze this data, we used both descriptive and inferential statistics. Descriptive statistics provide a picture of the "shape" and nature of the data. It includes total participation counts, max/min scores, score ranges, etc. Much of the story of how students performed can be revealed with descriptive statistics. Inferential statistics, on the other hand, typically engage in comparisons. For example, we might use inferential statistics to determine the extent to which boys and girls or students from different grades performed on the test and whether or not these differences were due to natural variation or if there is likely a significant difference between groups.

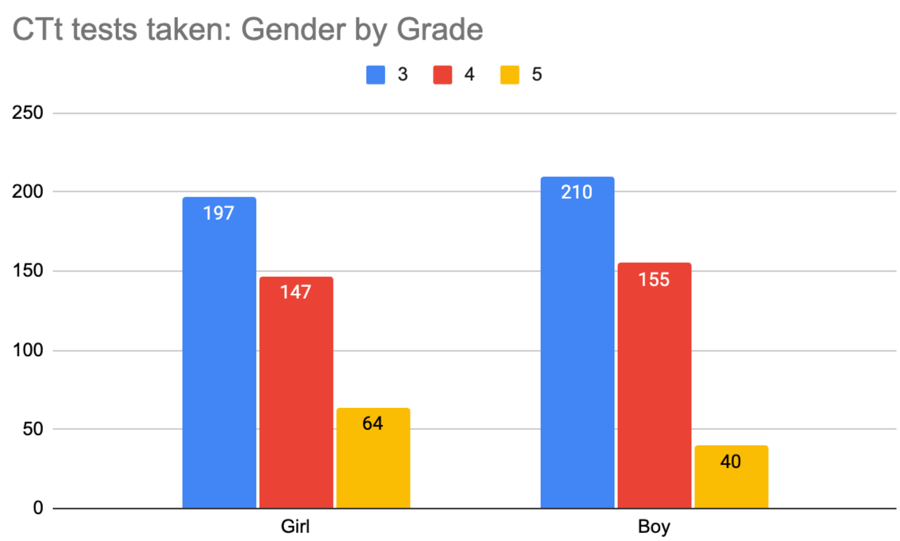
Findings

Three school districts, representing Georgia, Mississippi, and Oklahoma, completed the CTt. The same three districts administered the ESCAS, as well as a district from Alabama. I first present the findings from the CTt and then the ESCAS. This is followed by an analysis of correlations between these two measures.

CTt Findings

Overall, 933 students started the CTt. However, many of these did not attempt any questions or completed the exam fairly quickly. I contacted Dr. Román-González to ask what he thought the minimal amount of time needed to reasonably complete the CTt would be. He recommended not to consider any exams completed in fewer than 10 minutes, as those responses would likely only have been completed by guessing or marking answers at random. After eliminating such attempts, there were a total of 830 CTt exams, with students from 13 different schools (7 in Oklahoma, 5 in Georgia, and 1 in Mississippi). In total 410 usable responses came from girls, 406 came from boys and 14 students identified as 'other' (see Figure 2.1). There were 414 usable tests from 3rd graders, 314 from 4th graders, and 110 from 5th graders.

Figure 1



On average, students completed the CTt in about 28 minutes. There was a clear difference between grades, with 3rd and 5th graders taking an average of about 28 minutes and 4th graders each completing the test in 24 minutes. A little over 83% of students completed the exam.

CTt Performance Outcomes

Overall, scores were distributed normally, with a slight skew to the right (see Figure 2.2). Mean scores on the exam (see Table 1) showed about a .36-point increase by grade level. Third graders scored a 9.54, fourth graders scored 9.90 and fifth graders averaged 10.17 points. These scores are a few points lower than what has been observed by other studies that have administered the CTt to 3rd-5th grade students. The most common score across all grades and genders was 10/28 points. A t-test comparison of means revealed a statistically significant difference between overall boys' (10.148) and girls' (9.290) scores ($p < .002$). Research on the

CTt has consistently shown that boys tend to outscore girls by about 1 point. However, in this study there was one notable difference; 4th-grade girls performed on-par with 4th-grade boys. This may be more reflective of 4th grade boys under-performing, as 3rd-grade boys also outperformed both 4th-grade girls and boys. That is, one would normally expect 4th-grade boys to score in the 10-11 range. However, in this case, 4th grade boys averaged scores below 10 points.

Figure 2

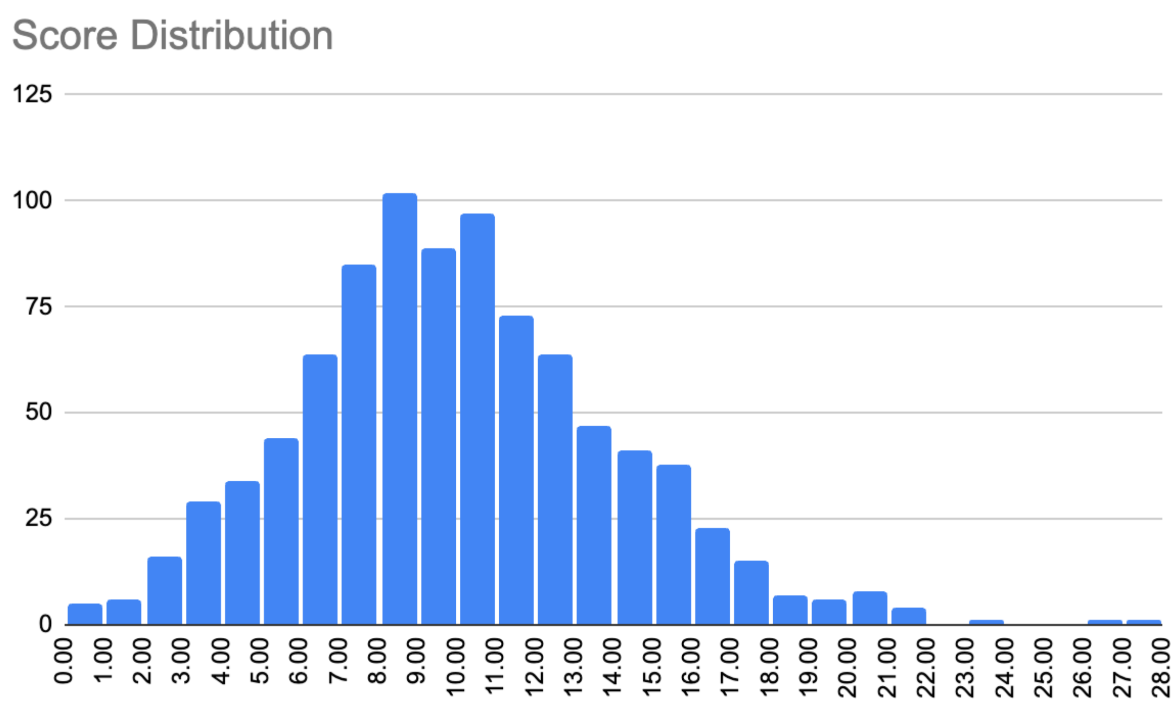


Table 1

CTt performance by grade and gender

	Overall	3rd		4th		5th	
Avg completion time	28.03 min	28.33 min		23.99 min		27.32 min	
Mean score	09.75	9.54		9.90		10.17	
Median score	09.00	9.00		10.00		9.00	
finished %	83.59	87.63		83.78		70.95	
		girl	boy	girl	boy	girl	boy
Mean score		8.84	10.13	9.82	9.89	9.50	11.18
Median score		9	10	10	10	9	10

CTt Performance Self-Assessment

In addition to coding problems, the CTt asks students to guess how many questions they thought they got right out of 28 as well as their facility with computers (on a 1-10 point scale). Estimation of performance has been shown to be an important indicator of students' ability to self-regulate their learning (Rebello, 2012).

Table 2

Students' self-estimations of score and comfort with computers

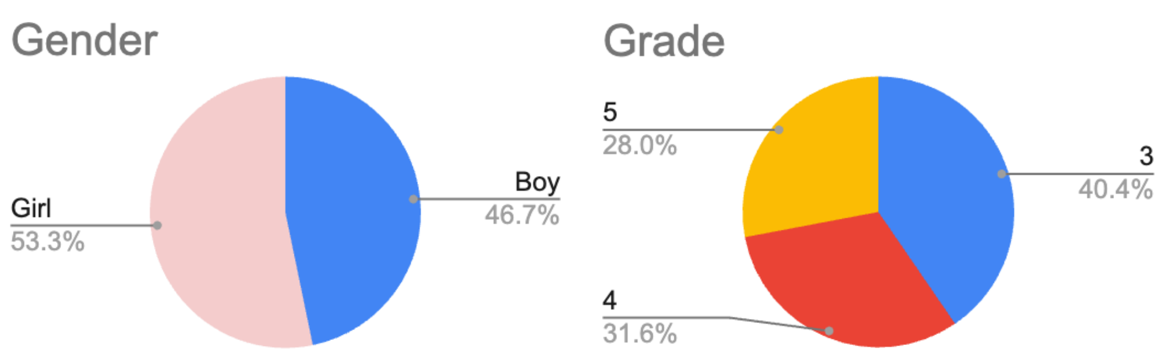
	Overall	3rd		4th		5th	
Estimated Correct # of 28	17.66	18.00		17.10		17.99	
Estimation vs. actual	4.85	6.01		4.27		2.16	
facility w/ computers	7.72	8.11		7.30		7.26	
		girl	boy	girl	boy	girl	boy
Estimated Correct # of 28		16.45	19.39	16.65	17.64	16.69	19.03
Estimation vs. actual		5.10	6.96	3.55	5.02	-0.11	5.48
facility w/ computers		7.81	8.42	7.03	7.49	6.75	7.71

On average, students overestimated their scores by nearly 5 points. Students improved their estimations by about 2 points for every grade they matured, meaning that older students made better estimates of their performance than younger students. In every grade, girls' estimations were closer to their actual scores than boys' estimations. The starkest difference occurred with 5th graders, where the difference between boys' and girls' estimations jumped to 5.5 points. In fact, the 64 5th grade girls who spent at least 10 minutes completing the CTt were surprisingly accurate on their estimations of their scores, actually underestimating slightly.

ESCAS Findings

Students in four districts (suburb-1, suburb-3, city-10, city-13) completed the ESCAS. Because this is a faster survey, we used completion as the criteria for inclusion in the analysis. While 649 students started the ESCAS, 589 completed it (311 girls, 273 boys). The survey was completed by 237 3rd graders, 185 4th grades, and 164 5th graders. On average, students completed the ESCAS in 12.22 minutes. Curiously, boys finished the survey 4 minutes faster than girls (10 min vs. 14 min). Older students finished the survey 1-1.5 minutes faster than those in the preceding grade.

Figure 3

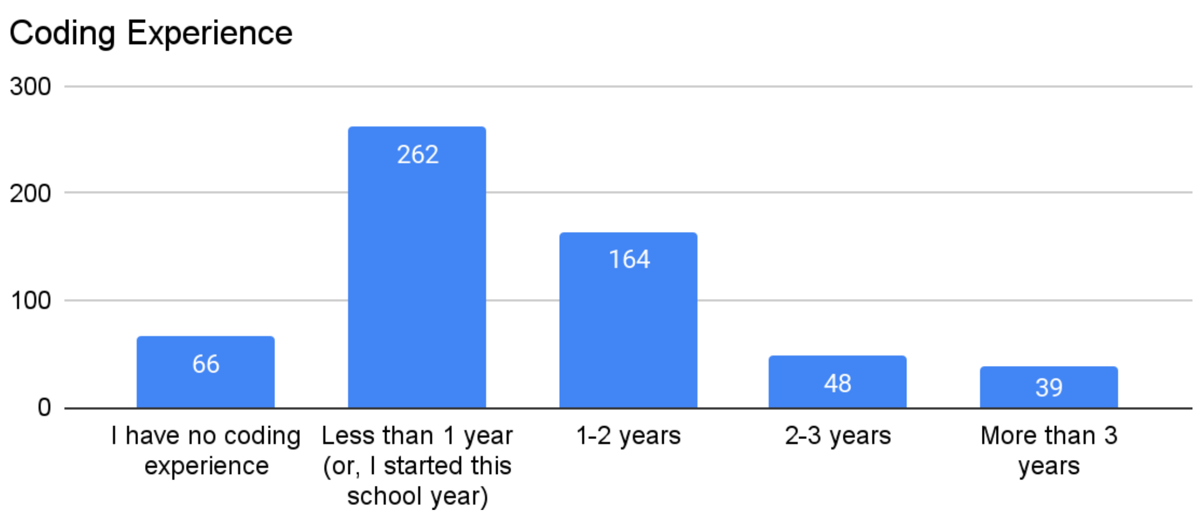


Students were asked to identify their race/ethnicity from a list of seven choices. They could select multiple races/ethnicities. Nearly two thirds (65.9%) of students identified as Black or African American. Another 17% identified as Hispanic, with only 8.8% of students identifying as White. Thus, the population for the ESCAS study is one made up in large part of students who are traditionally under-represented in computing.

Elementary Student Coding Experience

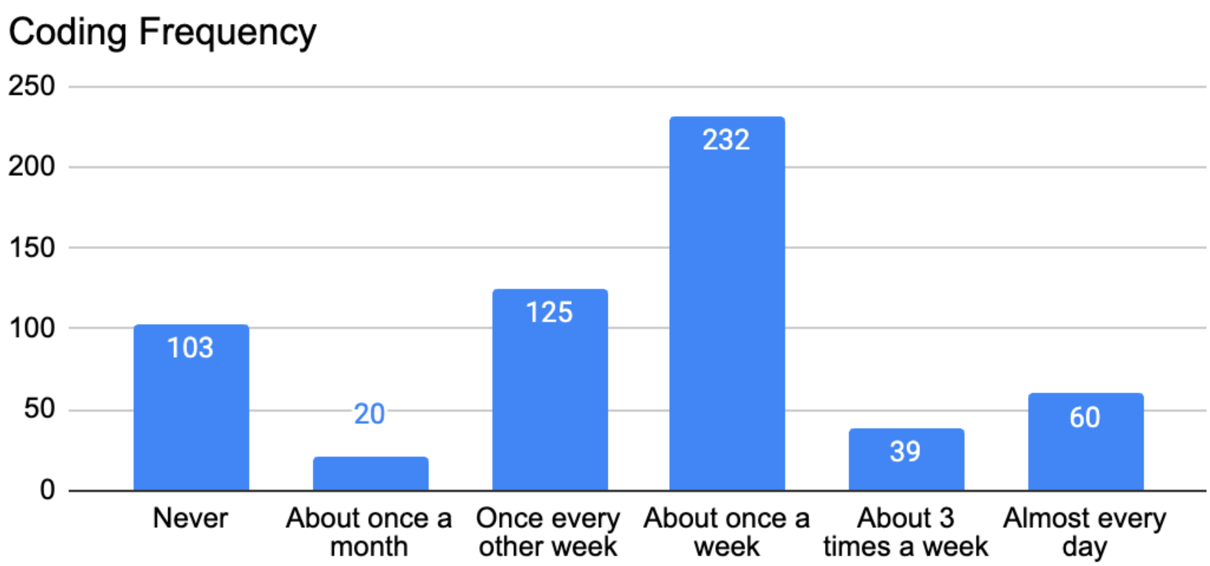
We asked students to indicate how much coding experience they had (see Figure 4), as well as how often they coded (see Figure 5). The majority of students (55.9%) indicated that they had less than a year of coding experience. Only 6.6% of students had more than 3 years' experience.

Figure 4



While students had little coding experience, the majority (60.7%) indicated that they were coding regularly. This means that, when they took the ESCAS, students were likely regularly engaging in coding lessons at school. About ⅕ (19.4%) of students were coding once a month or not at all.

Figure 5



Elementary Students' Attitudes for Coding

Students demonstrated slightly positive attitudes toward coding across most of the six measured attitudes. Their strongest attitude was actually not toward coding, but rather their confidence in their math abilities, which was rated ¾ of the way between "somewhat agree" and "agree" (see Table 3)

Table 3

Baseline Attitudes for Coding

ATTITUDE	score	count	boys	girls	3rd	4th	5th
Confidence	4.38	588	4.39	4.37	4.53	4.39	4.15
Interest	4.37	589	4.42	4.35	4.59	4.47	3.94
Utility	4.34	587	4.32	4.38	4.48	4.45	4.02
Perceptions	4.14	586	4.24	4.07	4.27	4.11	3.97
Math	4.75	585	4.82	4.71	4.91	4.70	4.56
Social Value	3.98	587	3.93	4.04	4.21	4.03	3.58

I used multiple regression to examine the predictive effect of demographics and experience on students' overall attitude score (i.e., the average of all their sub-scores). The model included grade, gender, coding experience, and coding frequency as predictors of students' attitudes. The model revealed a weak, but significant negative correlation between grade and attitude ($r = -.213$), but no significant effect of gender. In layman's terms, that means that older students in this population demonstrated slightly less confident attitudes toward coding than younger students (decreasing about .2 points per grade level) but that gender was not a significant factor between boys' and girls' attitudes. It is important to note that this grade-level effect is minimal, accounting for only 4% of the observed variance between grades.

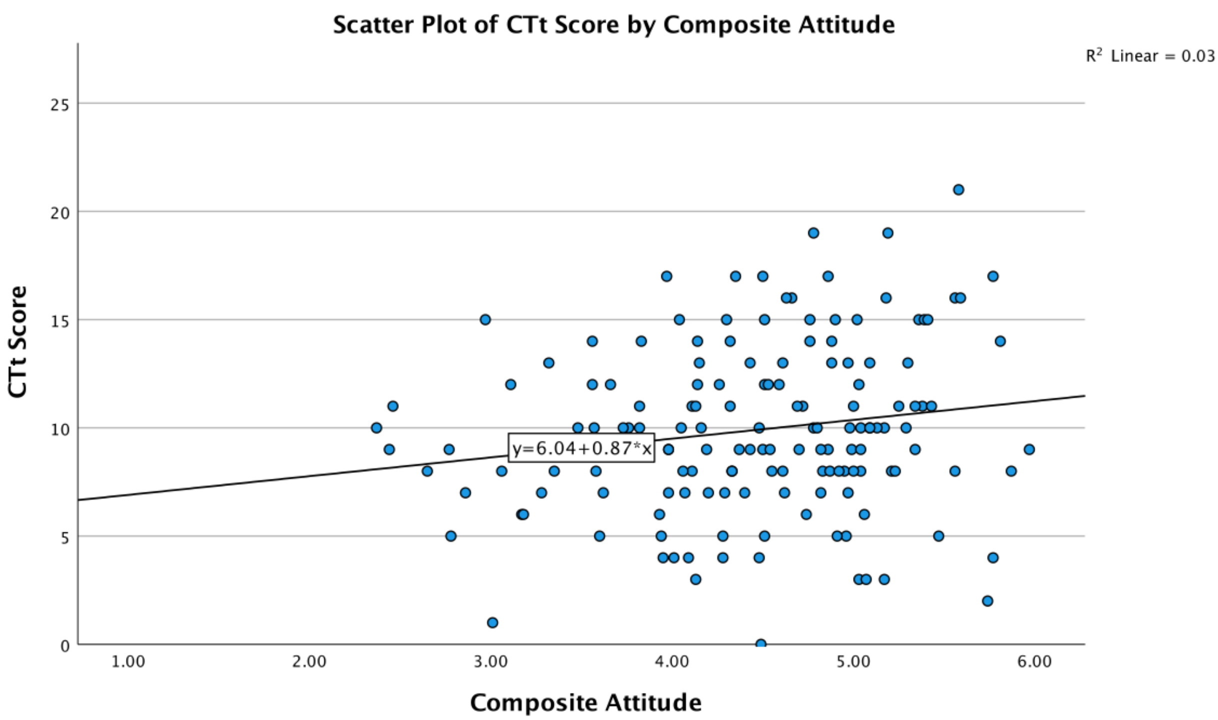
The model also revealed a weak, but significant, effect of coding frequency ($r = .207$, $p < .000$) on overall coding attitude. This means that, perhaps unsurprisingly, students who code more frequently have slightly more positive attitudes toward coding. That being said, this also only accounts for about 4% of the observed variance amongst students, meaning there are many other factors to consider. Taking into account all four predictor variables (i.e., grade, gender, coding experience and coding frequency) accounts for about 8% of the total variation in predicting a student's overall attitude toward coding.

It will be important to track these same students across years in future studies to see the extent to which their attitudes change. For example, will this same group of 3rd grade students carry over their more positive attitudes toward coding into their fourth grade year? If so, I would expect to see the observed effect of grade to decrease and disappear over two to three years. Furthermore, will that attitude become more positive with time as students more regularly engage with coding? We will be able to answer this question only if teachers in these same districts make sure to administer the ESCAS again at the end of the coming school years. Thus, it is vitally important that, at a minimum, these four districts continue to administer the CTt and ESCAS at the beginning and ending of the 2023-2024 school years.

Correlations between Cognitive and Affective Indicators

While the CTt and the ESCAS are separate measures, it might be helpful to know the extent to which they're correlated. In other words, do students who have more positive attitudes toward coding score higher on a cognitive test of computational thinking? Or vice versa? To test this, I first identified students who completed both measures. I did this by using students' self-provided school IDs. This resulted in 156 unique students, mostly representing Georgia (n = 113) and Oklahoma (n = 35), though there were a few students from Mississippi and Alabama, as well. A scatterplot shows students' attitude scores against their CTt scores (see Figure 6). This revealed a significant (p = .029) positive, but weak (r = .176) association between the two, accounting for only about 3% of the variance between scores.

Figure 6



Breaking this down by individual attitude (see Table 4), we can see that there were generally weak positive correlations across all attitudes. While both Interest and Utility were statistically significant, Interest was the only attitude to correlate above the .2 level, which is generally considered the level at which a correlation moves from nothing meaningful to weak. Two things are important to note here. First, correlation does not imply causation. Second, this may support the importance of collecting both cognitive and affective data for students, inasmuch as these two measures are clearly measuring something different.

Table 4

Correlations between students' CTt scores and measured attitudes

		Statistic			
Variable	Variable2	Correlation	Count	Lower C.I.	Upper C.I.
CTt	Confidence	.146	156	-.011	.297
	Interest	.257	156	.104	.398
	Utility	.157	156	.000	.307
	Perceptions	.091	156	-.067	.244
	Math	.088	155	-.070	.243
	Social	.041	156	-.117	.197

References

Mason, S. L., & Rich, P. J. (2020). Development and analysis of the Elementary Student Coding Attitudes Survey. *Computers & Education*, 103898. https://doi.org/10.1016/j.compedu.2020.103898

Rebello, N.S. (2012). How accurately can students estimate their performance on an exam and how does this relate to their actual performance on the exam. In AIP Cnference Proceedings, 1413, 315-318. https://doi.org/10.1063/1.3680058

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2016). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking test. *Computers in Human Behavior*. https://doi.org/http://dx.doi.org/10.1016/j.chb.2016.08.047

BootUp PD 2022-2023
Instructional Coach Report

Evaluation conducted by

Peter Jacob Rich, PhD

Table of Contents

Purpose

A unique aspect of BootUp's Professional Development model is their coordination with Instructional Coaches (ICs). ICs are district or administrative-level teachers who support other teachers in their efforts to teach coding. Each district that works with BootUp must commit to ensuring that an IC will work with the district's teachers. This helps to address issues of attrition (i.e., teachers leaving the district), onboarding, and on-going training.

This year, we were interested in the experience of ICs. Specifically, we wanted to know what their job duties entail, how much of their time they spend supporting the teaching of coding in the classroom, how many teachers they support, as well as the successes and challenges that ICs face. Knowing this information may help BootUp to more clearly communicate IC expectations with partner districts. Additionally, this information may help BootUp to identify and address possible barriers to successful implementation of coding in partner schools.

Executive Summary

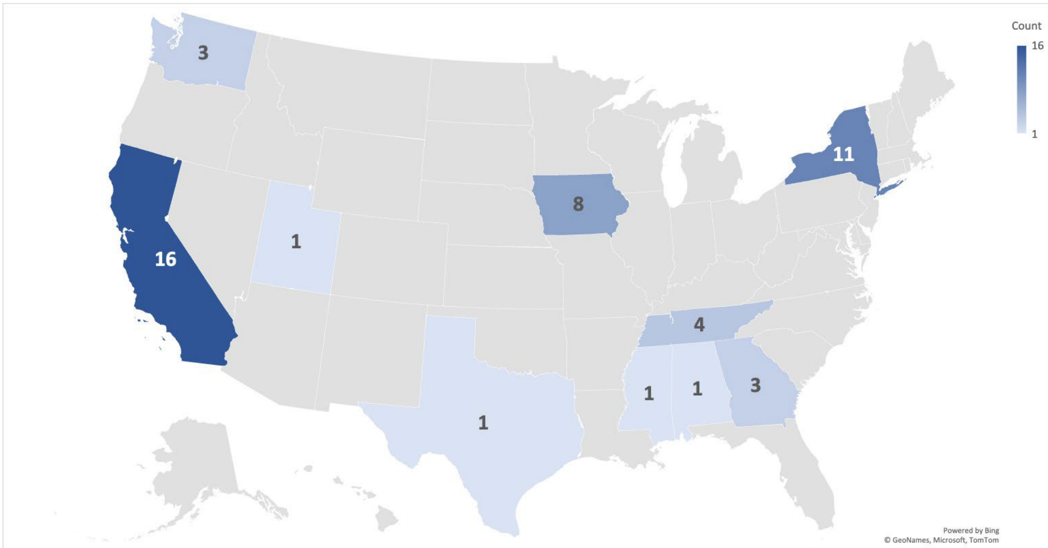
1. **Demographics**
    a. 94% of ICs (46/49) were trained to teach coding through BootUp PD.
    b. ICs are experienced teachers, with over 19 years of teaching experience, on average. However, they only have 3.5 years of experience as ICs.
    c. Twenty (40.82%) coaches had no coding experience prior to becoming an IC.
2. **Classroom Teaching**
    a. ICs report teaching at least 3 subjects, on average.
    b. ICs teach about 5 different grade levels, on average.
3. **Supporting Other Teachers**
    a. The average IC supports 27 teachers; most say this is a reasonable load.
    b. ICs spend nearly 45% of their time fulfilling IC responsibilities. Other duties include teaching, administration, PD, curriculum development, modeling, working with students and managing technology.
    c. About 42% of ICs make site visits weekly or more often. Site visits entail: providing resources, coaching, planning, model teaching, and co-teaching.
    d. ICs use BootUp resources about 1x/month to support other teachers.
    e. Expectations for how coding will be taught varies by district. The most common expectations are that coding will be taught weekly.
4. **Being Supported as an IC**
    a. Most ICs collaborate with BootUp facilitators 1x/semester.
    b. IC expectations may be set by a variety of players, but “district personnel” and “principals” are the most common.
    c. ICs feel well-supported by BootUp. They feel slightly less (but still) supported by their district, principals, teachers they coach, and the community.

Recommendations

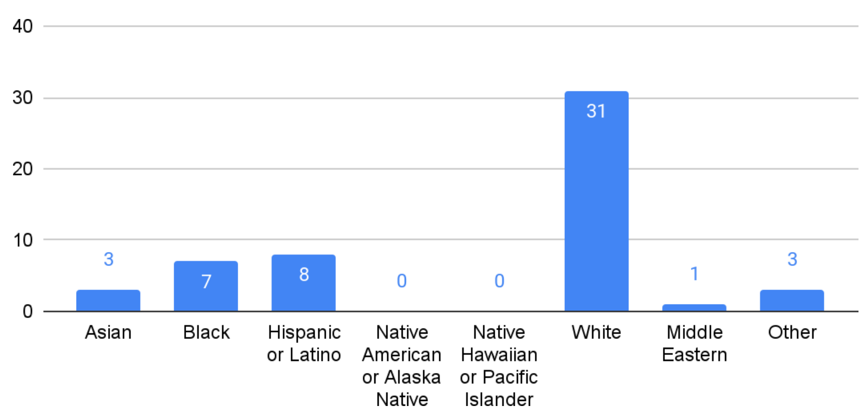
First, it’s important to note that most ICs expressed that they really enjoy their role as ICs. They find joy in seeing other teachers. Based on the findings, there are a few recommendations that may help to further ensure that ICs are well-supported in their roles to support elementary teachers to integrate coding in the classroom.

1. Establish clear expectations for **time**.
    a. Teacher Time: Since principals and district personnel, not ICs, establish expectations for how much a teacher should actually implement coding in the classroom, this needs to be clearly communicated all the way down the line. This will enable ICs to point to a clear standard when working with teachers. It will also help teachers to know how much they should be addressing coding in the classroom.
    b. IC time: ICs have many responsibilities, with only about half of their time being spent on supporting other teachers to learn to code. Given the possibility of competing priorities, establish clear expectations for what portion of time should be dedicated to supporting other teachers’ coding will provide clear guidance for ICs to know how to dedicate their limited resources.
2. ICs may need **continued training & opportunities to teach**
    a. ICs are enthusiastic about their role in supporting other teachers. They are also fairly new to coding themselves, expressing the lowest confidence with their own coding abilities. They operate largely on their own, only tapping BootUp resources once a month or less often. Bi-monthly training, opportunities to teach coding themselves, and opportunities to network with other ICs may help them to continue to build confidence and a repertoire of ideas for teaching coding.

Demographics



This survey was sent to all instructional coaches that have participated in BootUp professional development (PD) over the past 3 years (79 coaches representing 39 districts). Forty-nine Instructional Coaches (ICs) representing 22 different school districts across 10 states completed this survey. Fifteen districts were each represented by a single district. ICs reported having over 19 years of teaching experience, with about 3 ½ years of experience as an IC.

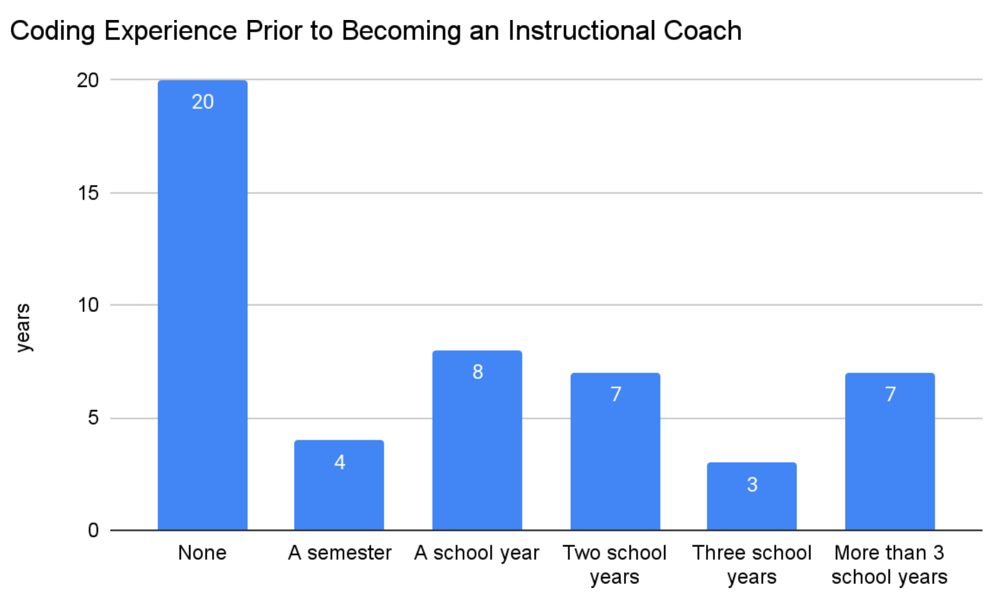


The racial and ethnic makeup of these teachers was primarily White (58.49%), with 13.21% identifying as Black. Just over 15% of respondents were Hispanic or Latino. Asian (5.66%), Other (5.66%) and Middle Eastern (1.89%) made up the remaining population.

Training

Nearly half of our ICs (49%) had less than one school year of coding experience prior to becoming an IC (see Figure 3.1). For many, BootUp was a primary experience in preparing them to teach coding. About 93% of ICs participated in BootUp's two main PD experiences— ScratchJr (for early elementary) and Scratch (for upper elementary). A handful of ICs participated in the newer trainings, which involve physical computing, and a course specifically designed for ICs to learn how to coach. Coaches rated BootUp as 8.59/10.00 when asked how well BootUp prepared them to be an IC, suggesting they found it to be an effective training.

Figure 1



While BootUp PD was effective and important for these ICs, it was not the first experience with learning to code for many. About 45% of ICs had participated in other coding training prior to the BootUp experience.

A Day in the Life

This section attempts to paint a picture of what an IC's experience looks like. These are general trends observed across our entire population. Individual experiences will undoubtedly differ based on district and school circumstances.

BootUp

Inasmuch as these ICs are part of the BootUP PD series, we were curious to know how it affected their day-to-day experience. On the one hand, ICs report using BootUp PD resources fairly frequently, with 77% using them monthly or more often. Despite this, ICs did not collaborate with BootUp professionals as frequently. Nearly 96% of ICs reported working with a BootUp professional monthly or less often. This indicates that, once trained, ICs largely operate on their own during a regular week, though they may have several spaced interactions with BootUp throughout the school year (about ⅓ reported monthly interactions).

Table 1

Other Subjects Taught by ICs

Subjects taught	count
General Elementary (K-6)	29
STEM	23
Computers	19
Language Arts	18
Mathematics	14
Science	14
Social Studies	12
Other	8
Art	5
Physical and Health Education	3
Music	2

Teaching

ICs teach much more than just computing. On average, ICs reported spending between 4 and 8 hours/week coaching, indicating that there are many other duties that occupy their time. The majority (59%) teach general elementary courses, with 47% teaching STEM and 39% teaching

Computers. As demonstrated by Table 3.1, it is apparent that ICs teach across the curriculum. In fact, ICs reported teaching at least three subjects on average.

In addition to teaching multiple subjects, ICs teach multiple grades. On average, each IC reported teaching five different grade levels. The most commonly taught grades were elementary (K-5), though between 11 and 16 coaches reported teaching secondary grades. Thus, while teaching multiple subjects may indicate that ICs are generalists, the fact that they teach multiple grades suggests that they're also specialists. This puts them in a unique position amongst their peers.

Self-Efficacy

ICs are looked at as the "coding experts" by their peers. Consequently, we were interested in better understanding ICs confidence to teach coding. When we asked them to evaluate their coding skills on a 10-point scale, ICs reported what might be considered a lower level of confidence (6.3/10), especially for an expert. That being said,their confidence to *teach* coding (7.8) and to teach the BootUp curriculum (7.7) both were in the moderately-high range, demonstrating that while ICs may continue to harbor some doubt about their own coding skills, they are more confident in their ability to teach coding to children and to other teachers.

I used an open-coding scheme to characterize teachers' rationale for their self-efficacy. For example, I characterized the following rationale as 'training,'

> "I feel prepared to support teachers to use Scratch and Scratch JR based on the BootUp professional learning that I participated in."

Coding teachers' rationales in this way enabled me to see if there were some reasons that lead to higher or lower ratings. For example, in Table 3.2, we can see that 13 teachers indicated that the BootUp curriculum itself was their rationale. What's more, these teachers all provided higher ratings across the board than for any other rationale that was provided by more than a single teacher. One teacher, who indicated that the curriculum was the reason for their rating, explained it in this way,

> "The curriculum has been skillfully crafted, and its comprehensibility allows for seamless navigation. Furthermore, it effectively intertwines with fundamental subject matters."

Likewise, the teachers who rated their self-efficacy the lowest were those who indicated that they still lacked adequate knowledge of coding or the curriculum. One teacher put it simply as, "We have just started with BootUp in February, so I have a lot to learn yet," while another stated, "I'm still learning, but I have a lot more to learn." Comparing these rationales for their ratings may help BootUp to better understand how to increase ICs' self-efficacy for teaching coding. Basically, those who are able to recognize the foundation that the BootUp curriculum and training provide are more likely to rate their own confidence to teach coding higher.

Table 2

Teachers' rationale for their self-efficacy ratings

codes	count	coding abilities	elementary coding	BootUp curriculum
curriculum	13	7.00	8.69	8.77
experience	15	6.20	7.80	7.20
foundation	4	6.25	8.25	7.75
knowledge	9	6.11	6.56	6.89
lack of knowledge	7	6.29	6.29	6.71
teachers	1	9.00	10.00	6.00
technologically inept	1	3.00	5.00	7.00
training	9	5.89	7.00	7.78

Support

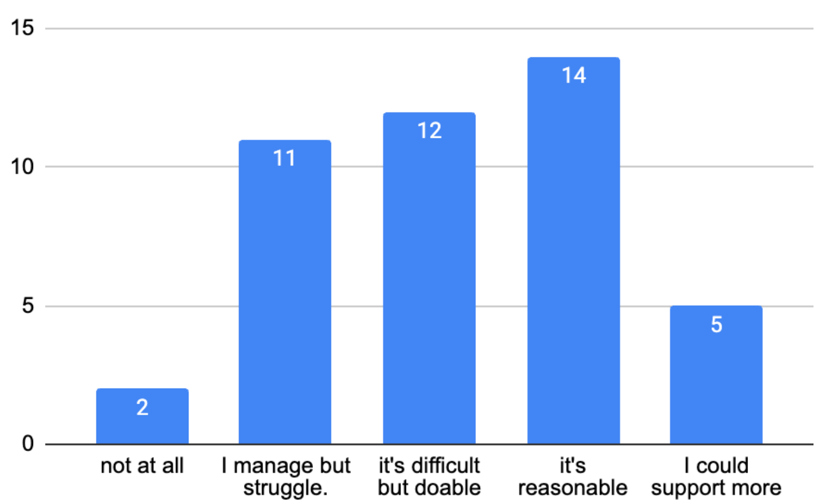
In this section, we look at several different ways in which ICs support and are supported in their role. This includes policy, practice, parents, and peers.

How do ICs Support Teachers?

First, we address the ways in which ICs support teachers. On average, IC reports that they support 27 teachers. I used the median to counter an extreme outlier that reported supporting 1000 teachers, which was 800 more than the next closest IC. When asked how sustainable they thought supporting this number of teachers was (See Figure 3.2), only two indicated that their load was unsustainable (and the person with 1000 teachers was not one of them!).

Figure 2

How sustainable is supporting this number of teachers?



In addition to teachers, it's helpful to understand how many schools an IC supports. The median IC supports two schools, which most report as being sustainable. Curiously, of the six ICs that reported their load was not sustainable, four coached teachers at two or fewer schools (with two supporting 15 schools). Thus, the sustainability of an IC's load does not appear to correlate with the number of schools they're asked to support.

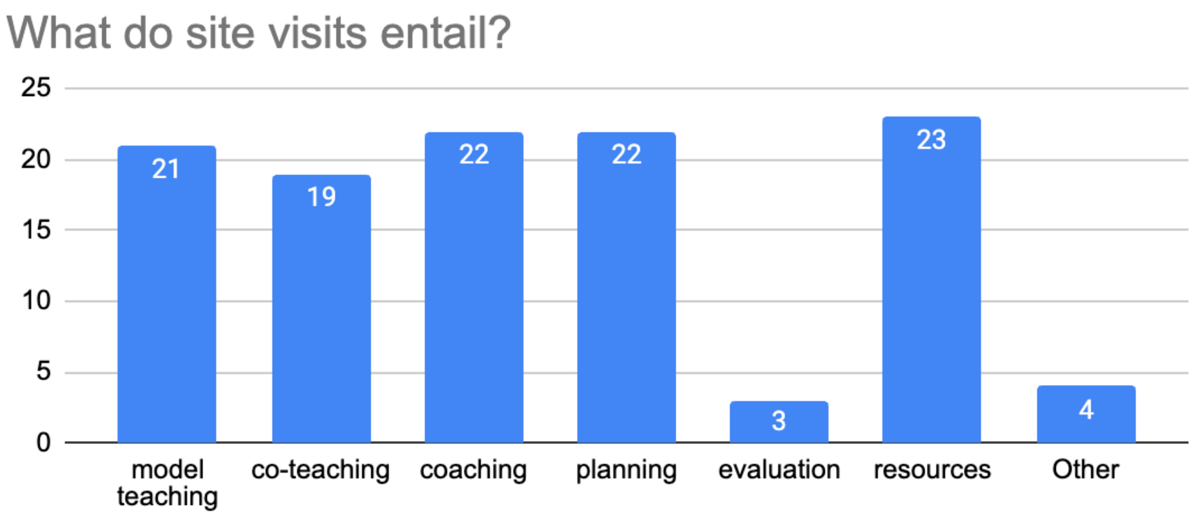
Table 3

Frequency of IC Site Visits

Frequency	# of ICs	Percentage of ICs
Never	8	18.18%
1x/semester	13	29.55%
twice a month	1	2.27%
monthly	3	6.82%
weekly	7	15.91%
2-3 times a week	3	6.82%
daily	9	20.45%

ICs report making site visits to teachers at varying rates (see Table 3.3). While 47% only make a site visit once a semester or not at all, 43% make site visits at least weekly. When an IC visits a school or a teacher's classroom, they engage in multiple activities (though not necessarily all in the same visit) (see Figure 3.3). Most ICs reported doing 3 different activities when visiting other teachers. While the most common was to provide resources, other activities included coaching, planning, model teaching and co-teaching, which were all almost equally popular. Thus, those preparing to become ICs should be comfortable with all of these activities. It also seems meaningful that ICs rarely conduct evaluations on site visits. This highlights their role in supporting teachers rather than judging them.

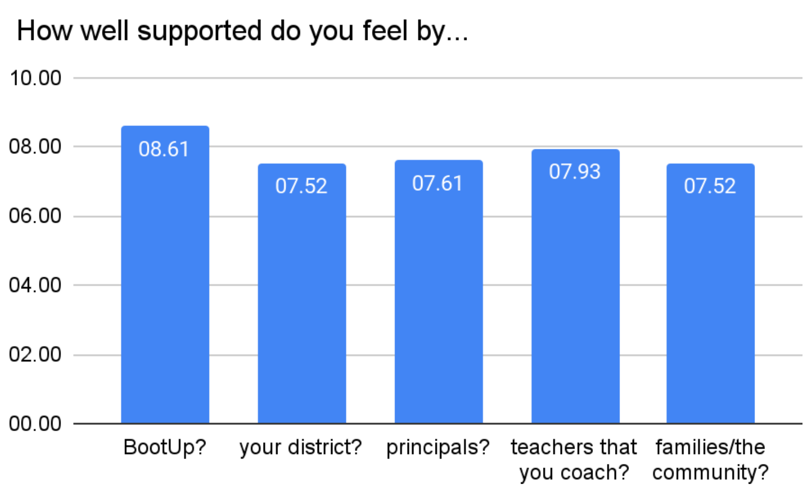
Figure 3



How Well are ICs Supported by Others?

In addition to supporting teachers, it's also helpful to know how well ICs feel supported by others. Some of this will be further explored in the "Expectations" section below. When asked to rate how well-supported they feel by different groups, ICs indicated they felt a moderate level of support (7-8 points out of 10) from their district, principals, other teachers and the community (See Figure 3.4). They felt a strong level of support from BootUp.

Figure 4

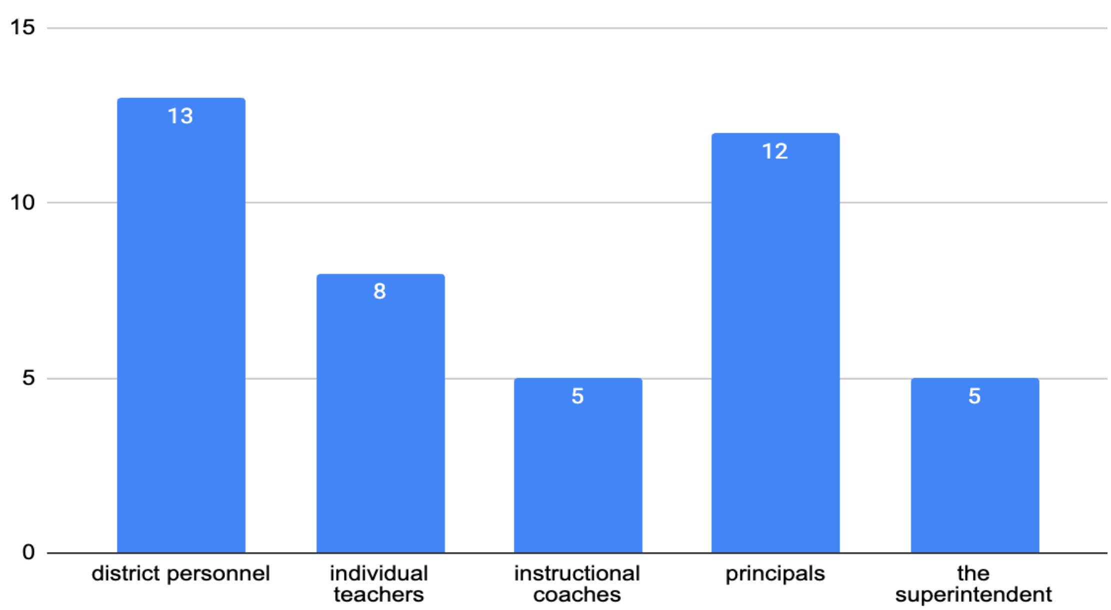


Expectations

When asked what the work expectations were for their role, 33 of 49 ICs provided a percentage for different activities. On average ICs reported spending 44.27% of their time on coaching. The range was wide, varying from 5% in one case, to 100% in several other cases. The most common answer was that ICs spend half of their time coaching. Other expected duties included teaching their own classes, administrative work, teaching PD or developing curriculum, model teaching, managing technology, working with students, and other, unspecified, duties.

Figure 5

Who is the person responsible for establishing expectations for teaching coding?



We also asked what the expectations were for teachers in their districts in regards to teaching computing. The most common expectation (about 30% of the time) was that teachers would engage students in coding weekly. One district expects about 20 hours a year, which may be similar. Surprisingly, about 10% of ICs reported that there were no specific expectations that teachers engage students in coding. ICs indicated that "district personnel" or "principals" were most often those responsible for establishing these expectations. Only 10% of ICs reported that they set teacher expectations (see Figure 3.5). This highlights the important role that district and school administrators play in ensuring that coding is being taught by their teachers in the classroom. Despite providing support, training and guidance to other teachers, ICs do not set the policy on how often coding must be taught by these same teachers. Thus, ICs need to work closely with principals and district personnel to ensure that teachers' time is structured appropriately to be able to meet the common expectation of teaching children to code once/week.

Highs and Lows

This section seeks to hear from ICs what the most rewarding and most frustrating parts of their IC jobs are. ICs were asked to provide an open-ended answer to this question, meaning that they were free to answer the question in any way they saw fit. I used an open-ended coding scheme to classify their answers and then looked for trends across those answers.

Most Rewarding

> "[The most rewarding part of being an IC is] being able to share and teach educators and students how to use computational thinking in context. The use of the Amazon BootUp program facilitated entry points of access to advance coding. Teachers and I enjoyed the multiple sources of resources and tutorials for each project."

ICs reasons for job satisfaction were many and varied, but the above quote provides a way of seeing how these all fit together. ICs provided 18 different explanations of what is most rewarding about their role. These fit into three main categories: teachers, students, and personal reasons.

Teacher reasons focus on supporting helping other teachers and seeing them change their practice. ICs rejoiced in seeing teacher growth, teacher confidence for coding, teachers' excitement for coding, and the "lightbulb" moments when something finally clicked. Teacher reasons for a rewarding IC experience is exemplified by the following quotes,

> "The most rewarding thing about being an ITF is changing teachers traditional way of teaching. Introducing teachers to new applications that can help with student success."

> "Teachers excitement and understanding has been the most rewarding part of this process. I have seen elementary teachers grow exponentially as a result of the BootUp training sessions."

> "Having teachers be successful in trying new strategies and curriculum."

> "The opportunity to positively impact more students by providing support to their teachers and speaking to stakeholders about equitable opportunities for enriching learning experiences for ALL students."

Student reasons emphasized student excitement, success and understanding. These comments demonstrated IC classroom interaction with students directly. ICs mentioned student excitement the most out of all student reasons. For example, the quote, "seeing student eyes light up when they figure out how to make the sprites do what they want," shows that some ICs work alongside students rather than just with teachers. Comments about student excitement also showed that ICs have noted when students take what they learn in their coding lessons and apply it to other aspects of their education, as exemplified by the following statement,

> “Watching the students be inspired. Seeing them apply their new knowledge to other projects that don't require on it. Having the students show me their projects that they worked on at home.”

Other teachers made similar comments about students applying their coding knowledge beyond their coding projects. Consider the following quotes:

> “Seeing the students love learning and grow in their academic and social skills.”

> “The students getting it and being excited about getting on different types of technology.”

Personal reasons was the least common of the three main categories. These included the satisfaction of learning new things, establishing relationships or the ability to share their knowledge with others.

Successes

When talking about their successes, ICs also gave answers focused mostly on teachers and students. The overwhelming majority of successes were framed in terms of teachers. The two most common successes were comments about teacher confidence to teach coding, and teachers’ implementation of coding in the classroom, including integration of coding with other subjects.

Table 4

IC's main Teacher Successes

Teacher Confidence	Teacher Implementation
I am successful in…increasing teacher confidence	Seeing one of my own teachers implement CS lessons in her class.
Festivals and playground to help teachers and students get comfortable!	Teachers are now designing integrated coding lessons.
Being able to take tech shy teachers and students and provide them with training and tools to soar.	Teachers using a new tool that I introduced them to use solemnly.
Connecting with other teachers and building confidence.	The teachers have been very willing to try using Scratch and Scratch Jr. I have seen some outstanding lessons so far.
Teachers feeling more comfortable with it in their classrooms.	When a teacher texts or emails me a strategy that we worked on together that worked for his/her class
I see teachers growing in confidence and are more willing to try	Getting teachers to transform the way they teach and integrate technology and computer science into their curriculum
Teacher willing to try new things.	Increasing teacher participation in educational technology
This is my first year and I have been working hard to change the view of coaches in the building. I've been successful in getting more buy-in from teachers and having them use the coaches.	Beginning a cohort of teachers who implement EduProtocols in their classrooms
The main success I have had as a coach is watching the teachers transition into coding experts who had doubts about teaching it.	Helping teachers implement a new literacy curriculum
The aha moments for both teachers and students. Also, seeing teachers being willing to allow students to explore and teach them and each other.	More teachers using coding as a tool for learning

The most common student-centered successes ICs noted in students was actually exposure. As one IC stated, "I've had success in implementing computer science/coding in 40 elementary schools." Others mentioned coding being taught in multiple schools without providing a specific number. Student success with coding and student understanding were also mentioned by multiple ICs (though much less often than the teacher-centered successes).

Most Challenging

As with ICs' successes and their most rewarding comments, IC feedback on what is most frustrating about their role resulted in many different contributors. However, there were two sources of frustration that stood head and shoulders above the rest: teacher attitudes and lack of time. The followings sections presents evidence for each of these individually.

Teacher Attitudes

The main source of frustration for an IC is dealing with teachers and administrators who do not buy into or understand the teaching of coding. Fully 40% of teachers mentioned something about teacher attitudes when asked about their frustrations. When discussing attitudes, ICs often suggested that resistant teachers were unwilling to move from their traditional ways of teaching. In some cases, a teacher's attitude meant assuring them that the need for a coach did not indicate that the teacher themself was poor at their job. In other cases, it involved ICs convincing teachers that teaching coding was not just "one more thing" to add to their plate, but that it could be integrated in a way that complemented, rather than competed, with teachers' existing responsibilities. And still in others, it involves acknowledging that not all teachers are prepared to discuss issues of equity and access the CS Education. Table 3.5 provides all the comments ICs made about their frustrations with teacher attitudes.

Table 5

IC Frustrations about Teacher Attitudes

IC Quote
● Some teachers do not try to implement the skills or strategies for coding in their own classrooms.
● The resistance or a lack of commitment from teachers can be the most frustrating aspect of being an instructional coach. This is especially true when it comes to introducing new teaching methods or approaches, which may be met with reluctance if teachers have grown accustomed to their current practices. Juggling competing demands and priorities from school administrators, teachers, and students can also be a significant source of frustration. Additionally, managing the workload of supporting multiple teachers and catering to their individual needs can be challenging, potentially leading to feelings of being overwhelmed and burnt out.
● When teachers won't get on board and therefore their kids miss out on opportunities.
● Most of the time teachers want us to push in and deliver the lessons rather than use us as actual coaches.
● The most frustrating is not having buy-in or having teachers who don't want any support and are used to traditional ways.
● Convincing other teachers to utilize it more often and that it doesn't require anything extra from them.
● Not everyone likes change, so it may take longer to implement new curriculum.
● Teacher Push Back
● Close minded people and people who complain about the coaching position.
● Teachers unwilling to add motivation to learning.
● Getting teachers to understand the importance of teaching computer science
● Struggling with educators afraid of technology or afraid of not being the experts in the room.
● When teachers don't attempt to figure out aspects of technology programs on their own
● As I mentioned previously, it is getting teachers to understand that this isn't an extra thing added to an already full plate of responsibilities.
● Resistance to change and working with people that have a fixed mindset.
● Working with teachers/admin that do not see the "CS vision" and how impactful students learning CS can be for all academic learning.

IC Quote
● The most frustrating part about being an instructional coach is trying to get everyone to realize that working with a coach doesn't mean you aren't a great teacher. It's trying to get teachers to be more reflective and realize we all have room to grow.
● Knowing that not every teacher at my schools is ready to have conversations around equity and access to CS ED and other crucial content.

Lack of Time

Lack of time is probably one of the most-commonly cited frustrations that teachers express. However, research has shown that consistent training can overcome time barriers, especially when it comes to technology-related integration (Rich, 2012). Lack of time can also be overcome at the administrative level by restructuring a teacher's schedule to account for and communicate the important of coding time during regularly weekly instruction.

Regardless of the solution, it's important to recognize that teachers feeling that they do not have the time to implement a new program are likely to not put it into action. ICs characterized lack of time in two primary ways. First, most discussed their own lack of time and competing priorities. Second, a few ICs spoke about the teachers they support and their lack of time to either work with the IC or implement what they'd learned in their own classes. Table 3.6 shares both of these types of "lack of time" frustrations.

Table 6

IC Comments about Frustrations with the Lack of Time

Quote	Person Affected
It would be nice to have more dedicated time to support others in their classrooms.	IC
Time to continue learning and practicing.	IC
Perhaps the frustrating part is trying to find time in the school's PD calendar.	IC
Time or lack of time is the most frustrating part. Teachers are spread so thin. I know they would like to commit more time to learning coding and designing integrated curriculum but unfortunately they simply do not have the extra time.	Teachers
Not enough time in the day to teach all the fun things I could.	IC
Not enough time to get to all of the teachers in my building.	IC
Not having enough time to support all of the teachers	IC
At a secondary level, getting time and space to work directly with teachers, who are often overwhelmed by other responsibilities/initiatives.	IC, Teachers
Not enough time.	IC
I am frustrated with the lack of time to support teachers.	IC
TIME and Making sure that all technology works to include internet. Having the right equipment.	IC
Fitting all of the job requirements into my daily schedule at times.	IC

What more do you wish you could do?

Rounding out this section of teaching challenges is a question about what more ICs wish they could do. In contrast to their frustrations, there was no single clear thing that ICs wish they could do more of. Their answers were almost as varied as the number of ICs. One of the more common answers was more time; this was more time to plan with teachers, to co-teach, to train, to polish their own skills, or to collaborate around computing in some way or other. Curiously, the second-most common response was “none” or “NA,” indicating that there is a group of ICs that are satisfied with their current roles and the way they’re able to carry them out.

Conclusion

The goal of this study was to better understand the experiences of instructional coaches who support elementary coding through the BootUp program. Coaches tended to be experienced, mid-career teachers with about 3.5 years of experience in teaching coding. They support about 27 teachers at a few different schools, though this number can vary greatly. On average, they spend about 50% of their time coaching. They teach multiple subjects (3, on average) and are thus dynamic teachers that both specialize (e.g., educational technology) and generalize. Coaches find the most satisfaction in their role in helping other teachers to feel confident and catch the vision of implementing coding in the classroom. This perhaps explains why one of their greatest frustrations is the flip-side of that coin—teachers’ general resistance to change and an unwillingness to experiment with new methods, technologies, and approaches. Coaches’ time for actual coaching is most often decided by principals and district personnel, which highlights the important role that systemic structure plays on coaches’ experience. As with many teacher studies, lack of time was a common factor in their frustration. Despite these frustrations, though, coaches find joy in their role supporting other teachers to learn to teach coding. Changes in teacher practice and student excitement highlighted their reasons for this joy. Finally, instructional coaches felt most confident in their ability to help others when they felt more confident in their own ability to code. This is curious, since the coaches in this study only rated their confidence in the 6/10 range. Thus, it is important for coaches to continue to develop their knowledge of coding beyond their initial training.