Evaluation Report 2024-25

EVALUATED BY
PETER RICH, PHD

BOOTUP PD 2024-25

FULL EVALUATION

Peter J. Rich, PhD

EXECUTIVE SUMMARY

Strengths

High teacher satisfaction:

98.4% would participate in BootUp again
Net Promoter Score: 58% (strong)
Avg. rating for BootUp experience: 8.8/10

Significant growth in teacher confidence:

+51% increase in confidence (1–10 scale); large effect size (Hedge's g = 0.83)
Major gains in self-efficacy for coding (g = 0.96) and teaching coding (g = 0.94)

Teachers felt better prepared:

Over 80% met or exceeded their own expectations for teaching coding
Integration of coding into core subjects (esp. ELA, science, math)

Highly valued program features:

Hands-on learning, peer collaboration, modeled lessons, and BootUp-provided resources
Teachers cited increased confidence, new tools, and better student engagement

Diverse participation:

60% with graduate degrees; 33% of teachers were STEM/specialists
Representation from 20 states, 87% female, racially and ethnically diverse

---

Critical Gaps

Instructional coach (IC) engagement was low:

54% of teachers reported no interaction with their IC

Inconsistent expectations and support:

Only 31% said expectations to teach coding came from school leadership
Teachers felt least supported by peers and families

Inclusivity efforts were underdeveloped:

26 teachers said they did "nothing" to make coding inclusive
Common equity strategies (differentiation, collaboration) need formal support

Mixed implementation intensity:

Nearly 2/3 of teachers taught coding "a little" or less often
Time, competing demands, and scheduling were cited as barriers

Student scores:

Student attitudes remained flat or declined, as measured by the ESCAS.
Gender gaps remain between boys and girls cognitively and attitudinally

---

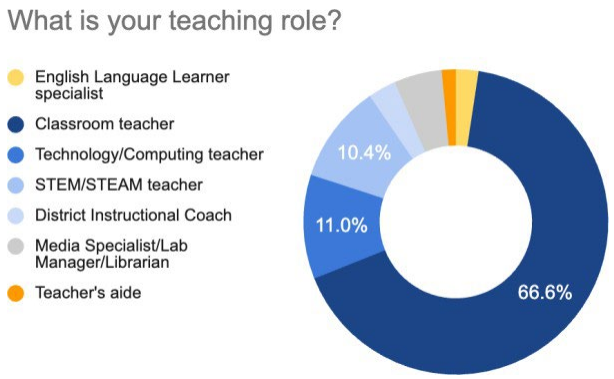
DEMOGRAPHICS

This section provides a profile of the teachers who participated in this evaluation

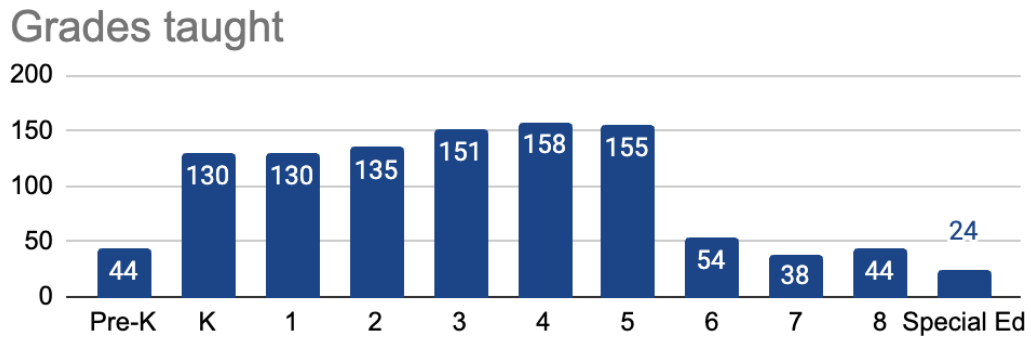
In 2024-2025, 733 teachers from 20 different states and 38 school districts completed the pre survey, while 336 teachers from 19 states and 27 school districts completed the post survey. This represents a 46% completion rate. 87% of teachers were female. Racially, just over half of the teachers were White, 1/3 were Black/African American, and 13% were Asian. 17% of teachers were of Hispanic ethnicity. These are well-educated teachers, with 60% having earned an MS degree or higher.

Race	#	%
Asian	44	13%
Black or African American	106	32%
Middle Eastern	4	1%
Native American	5	1%
Native Hawaiian or Pacific Islander	3	1%
White	174	52%

The majority of these were classroom teachers, meaning that they are generalists who teach

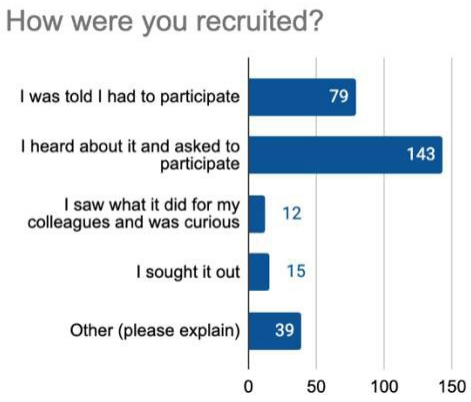


multiple subjects to the same group of students throughout the day and week. However, about 1/3 of teachers were specialists, teaching STEM, technology/computing, ELs, Library Media, or other specializations. On average, teachers had 13 years of teaching experience and about 2 years' experience having taught coding in some capacity. Primarily, these teachers work in elementary grades (PK-5th), though there are a handful who teach some middle grades, as evidenced in the chart below.



"Every time we would meet we would learn something new and feel more comfortable with the idea of coding" — 5TH/6TH GRADE CLASSROOM TEACHER

Teachers reported participating in BootUp PD for a variety of different reasons in 2024-25. Over 60% of teachers volunteered in one way or another.



Perhaps due to their willingness to participate in a coding-related PD, teachers were generally positive about the value of children learning to code. One teacher put it this way,

"Learning to code offers a significant amount of value for children, extending far beyond simply preparing them for a career in tech. [It] develops Computational Thinking, enhances problem-solving skills and boosts logical reasoning and critical thinking"

—STEM/STEAM Teacher

District Survey Completion	pre	post
District 1	33	15
District 2	2	1
District 3	10	5
District 4	15	2
District 5	2	0
District 6	1	0
District 7	14	6
District 8	1	0
District 9	38	3
District 10	25	10
District 11	24	17
District 12	5	5
District 13	10	3
District 14	18	4
District 15	1	1
District 16	69	55
District 17	49	42
District 18	36	5
District 19	10	11

Motivation

It is helpful to know what drives the teachers who choose to participate in professional development. To that end, we asked teachers what their motivation(s) was/were for participating in BootUp this year. Teachers revealed a variety of different answers, which we analyzed and grouped into common categories. The most common responses are represented in the following table.

motivation	# of teachers	Representative Quote
teaching coding	87	I wanted to be prepared to teach coding. I wanted to make sure I could get ahead of our mandatory implementation in the near future.
learn new skill	65	I wanted to stop avoiding technology and understand it. This program made me feel like a learner again, in the best way. Coding becomes another tool in the classroom, not an add-on. This is about giving my kids a seat at the table in the tech world. The best part was learning from other teachers across the district who were trying the same things as me.
help students learn	34	So I can help students to succeed
mandated	20	It was mandatory, but I did enjoy it and I am glad that I was chosen for this. I learned many new things.
collaboration	10	As a DL teacher I am always looking for opportunities to connect and collaborate with other teachers and for new ways to help support the unique needs of my students.
Other motivations	<10	resources (7), improve pedagogy (3), n/a (3), paid training (3), student opportunity (3), learn to integrate coding (2), teaching STEM (2), to gain coding experience (2), for coding resources (1), I heard they had friendly facilitators (1), high (1), low (1), intrinsic interest (1), prizes (1), respresent my school (1), student interest (1), to teach micro:bits (1), to teach science (1), to help others (1)

Teachers' primary motivations centered on building up their own skillset, whether it be specific to coding or learning something new that they saw as relevant. They were also interested in learning something that would help their students or collaborating with other teachers. For the full transcripts of teachers' answers to this question, see Appendix A.

Value

Similar to motivation, it is helpful to know what value teachers find in coding. This reveals the way that they view coding and potentially what influences how they approach it with their students.

Value of Teaching Coding	# of Teachers	Representative Quote
problem solving	97	I find immense value in it. It promotes them to problem solve and be able to even decide what the block is asking or having the sprite do. It also allows them to find joy in connecting it with classroom content.
career preparation	69	I think that it is very important to teaching coding to students because I believe that many careers are utilizing and will utilize coding now and in the future.
critical thinking	50	Critical thinking through anticipating, planning, and problem solving.
creativity	48	Coding empowers students to build games, apps, or animations from scratch. It turns them into creators, not just consumers, of technology.
understanding technology	45	I think it is important for students to understand the basics of technology at a young age
intrinsic	41	There is an extreme intrinsic value in teaching my students to code!
resilience	40	They learn patience and understanding. Patience to check if the codes used are correct and if not, they need to find out which code should be used. Understanding the concept on how coding works to apply it on current project.
logical reasoning	37	Use of logic to solve problems
computational thinking	32	Coding is a value because it teaches children how to break down more complex problems into steps that are manageable.
interest	30	Alot of value. The students loved it and were very engaged anytime we completed coding lessons.
collaboration	24	Collaboration and teamwork, problem solving, critical thinking skills, creativity, artistic design, patterns, stamina
other values	< 10	form of expression (9), future (6), communication (4), good for struggling students (3), STEM (3), skill building (3), practical (2), planning (2), none (2), confidence builder (2), 21st century learning (2), transfer (1), self-regulation (1), perspective (1), organization (1), leadership (1), follow directions (1), exposure (1), empowering (1), community building (1), challenge (1)

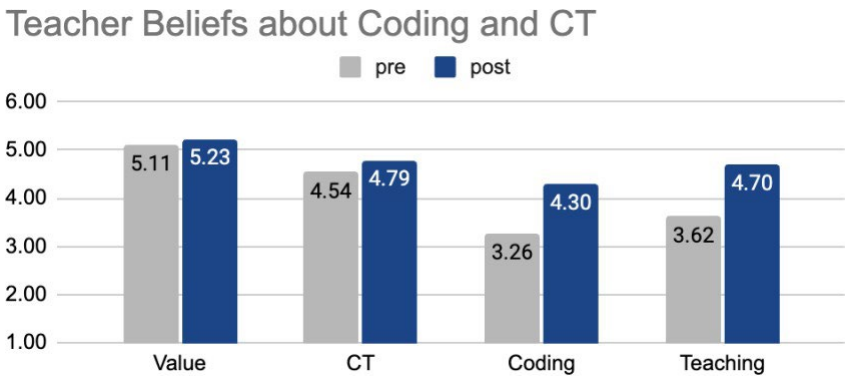
These values can be further categorized into 4 major themes. Teachers believe students should learn to code for: (1) professional/economic reasons, (2) cognitive benefits, (3) to foster desirable behaviors (e.g., resilience, creativity), or (4) simply because it is intrinsically interesting to students. Understanding the way(s) a teacher values coding may have implications for how to prepare materials that they will find useful. See Appendix B for a full transcript of teachers' comments about the value of children learning to code.

THE EFFECT OF PD

In this section, we examine the ways in which participating in BootUp PD affected teachers and their classroom practices.

The pre survey primarily asks teachers about their beliefs and their feelings about teaching coding to children. These beliefs are again measured on the post-PD scale at the end of the year. Four beliefs are measured on a 6-point Likert scale. The scale ranges from 1 (strongly disagree) to 6 (strongly agree) with statements such as: *I can explain coding/CT concepts well enough to be effective in teaching computing*

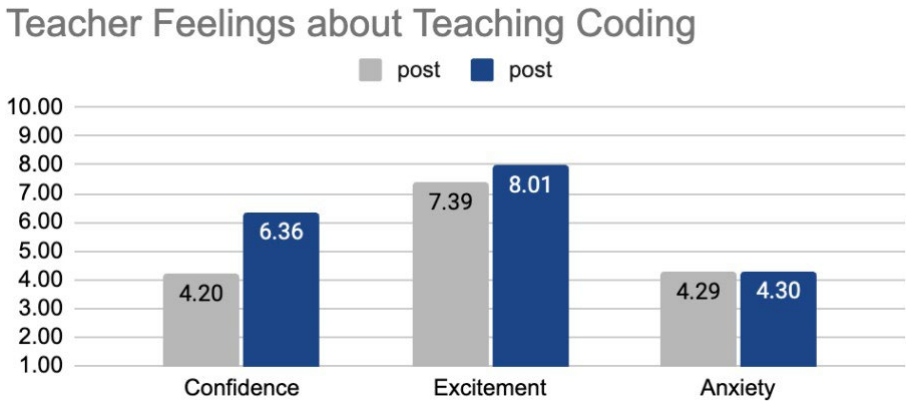
Values (10 statements): Beliefs about the importance of teaching coding to children.
Computational Thinking (4 statements): Self-efficacy statements about the teacher's confidence to identify patterns, break down large problems, and solve algorithms.
Coding (**10 statements)**: The teacher's self-efficacy for coding, involving loops, conditionals, variables and other coding-specific knowledge.
Teaching Coding (11 statements): Teachers' self-efficacy for successfully teaching coding in their classroom, including helping to debug problems and find suitable teaching resources.



These results show a few different things. First, teachers improved in all areas. Second, teachers started off agreeing that teaching children to code is valuable and should be done. They also were fairly confident in their own ability to think computationally. However, they began the year less sure about their coding ability or how well they could teach coding. Using t-tests to compare pre and post results revealed that all of these changes were statistically significant. While the practical effect for gains in value and CT beliefs were small to moderate, teachers experienced significant large effects for coding and teaching (Hedge's g = 0.96 and 0.94, respectively).

Feelings

In both the pre and the post survey, we asked teachers what their feelings were about teaching coding on a 1-10 point scale. They reported their **confidence**, **excitement** and their **anxiety** for teaching coding. These feelings are good indicators of teachers' readiness and willingness to teach coding in their classrooms.



Note that teachers started with low confidence, high excitement and low anxiety. Over the course of the school year, their confidence grew 51% and their excitement grew by 8%. Anxiety remained the same, though it was low to begin with. Changes in confidence and in excitement were both statistically significant. The effect for confidence was large (Hedge's g = 0.83).

Before and After

These changes are especially noticeable in the way teachers talked about their feelings for teaching coding before and after participation. Teachers provided open-ended written responses to this question on the post survey, juxtaposing how they felt about teaching coding *before* and *after* BootUp PD. We analyzed their responses and grouped them into common categories. As noted in the table below, teachers started off with a lot of anxiety about teaching coding (even though they rated it low in this above chart). They also expressed feelings of inadequacy in terms of ignorance about what coding entailed, and low confidence, with some teachers expressing excitement. The following table lists any feelings that were noted by at least 10 teachers.

Before Codes	#	Representative Quote
anxious/scared	75	Very fearful of not knowing enough to teach students especially my middle school students who may already have some background in coding. It was very intimidating.
ignorant	42	None. I didn't even know what coding was!
low confidence	41	Before the BootUp trainings this year, I was not confident in coding and it gave me a lot of anxiety considering how I would teach this to my fourth grade students.
excited	29	I was excited to teach coding, I wasn't completely sure how I would integrate teaching coding into my lessons or my schedule.
dubious	27	I was worried that majority of the students would struggle.
positive	26	I had experience so I am optimistic when it comes to coding.
neutral	12	neutral
Other feelings	< = 10	negative, overwhelmed, uncertain, confident, curious, interested, none, believing, disinterested, hesitant, comfortable, n/a, unprepared, unsure frustrated, mixed, reserved, uncomfortable, willing, ambivalent, cautious, eager, novice, overconfident, same, uncommitted, unwilling

When we compare these responses with teachers' response as to how they felt *after* participating in BootUp PD, we can see a clear shift in their attitudes.

After Codes	# of Teachers	Representative Quote
more confident	104	After the BootUp trainings, I feel a lot more confident that I can teach coding to all grade levels. For this course, I taught 3rd graders. We completed 3 lessons and the students did an amazing job. I think if I had more time to actually take a deep dive into each of the blocks and its functions, the students projects would be even more elaborate. I am a lot more comfortable teaching coding now than I was before this course.
excited	54	I am excited to teach coding! I saw how fun and engaging the lessons were for me, so I knew that my students would also be interested and would have fun. I have really enjoyed teaching my students to code and I look forward to engaging my future students with more lessons and enjoyment.
positive	45	Positive now and I feel more comfortable and confident that I can teach the content to kindergartners.
believing	22	After this course, I could see that it is possible for the lower grades. They enjoyed the challenge and were willing to try to get their coding. I think I was more apprehensive about coding than the students. They took to learning it and was engaging in the lessons.
prepared	21	I feel comfortable and confident in implementing various coding lessons into my curriculum, and if I am stuck on something I now know I have a lot of resources through BootUp to help me.

For a full list of teachers' quotes about their feelings for teaching coding before and after BootUp PD, see Appendix C.

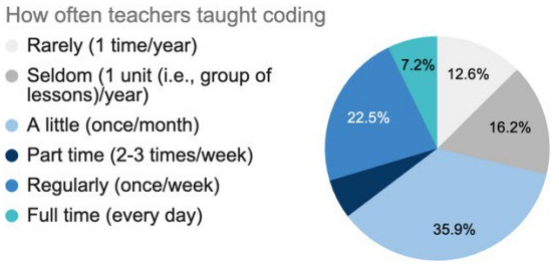
"Most students dove right into coding, but a few became stuck on details which led them to get frustrated (in not finishing their projects). During extra time, I directed them to just create a "skeleton" for the project and then add details. This led them to have greater confidence and feel they had the agency to add their creativity."

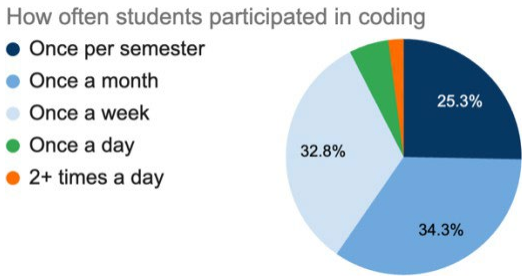
— 3RD/4TH/5TH GRADE TEACHER

CODING IN PRACTICE

In this section, we report on the ways in which teachers implemented coding in their classrooms throughout the school year.

Nearly 2/3 of teachers reported teaching coding "a little" or less often, as evidenced in the charts below:





To put this practice into perspective, we asked teachers what the expectations were for teaching coding throughout the year. We then analyzed the extent to which they met those expectations.

Expectations to teach coding		Expectations source?		expectations x actual	
There were no expectations	84	myself	220	didn't meet	57
I would teach 1-2 coding lessons over the course of the year	58	my principal	50	met	200
I would teach coding at least once a month	66	my superintendent	4	exceeded	75
I would teach coding at least once a week	44	my department chair	11		
I would teach coding daily	15	district personnel (e.g., specialist)	38		
I would integrate coding into my existing lessons	67				

Over 82% of teachers met or exceeded expectations for how often they would teach coding throughout the year. The majority reported that these expectations were self-imposed. Only 31% of teachers reported that these expectations came from administrators of any sort. What's more, about a quarter of teachers reported that there were no expectations to teach coding throughout the year. Thus, there may be value in clarifying expectations for how often teachers will implement what they learn with BootUp PD throughout the year.

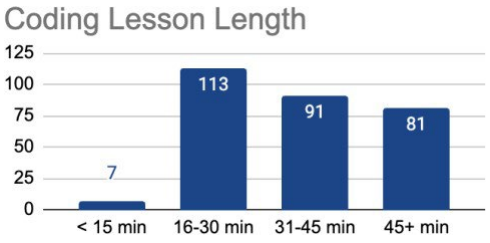
"Every time we coded, a student surprised me with their knowledge base. They would ask to come to the Smart Board to demonstrate. It was awe-inspiring."

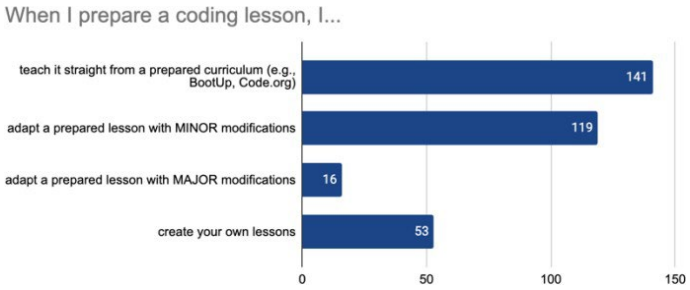
— Technology/computing teacher

Lesson Composition

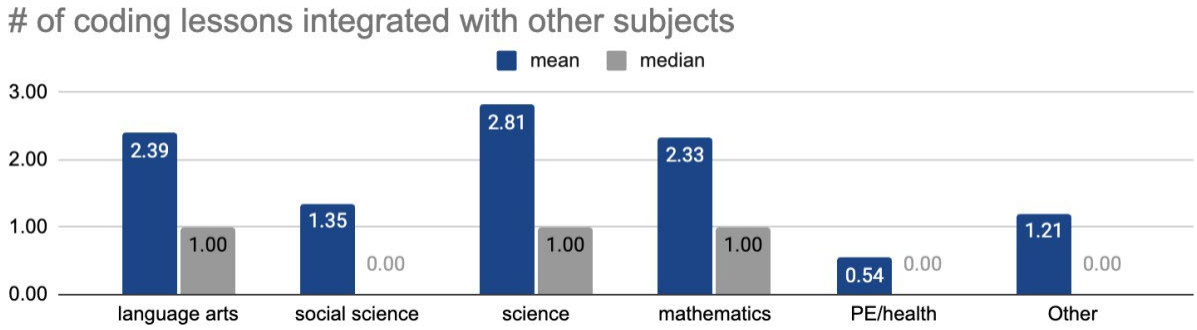
In this sub-section, we take a deeper look at the nature of teachers' coding lessons.

The most common length for a coding lesson was 16-30 minutes. However, nearly 60% of teachers taught longer lessons. The vast majority of teachers relied heavily on BootUp materials to plan and prepare these lessons, demonstrating the importance of providing ready-made lessons and materials for teacher success.





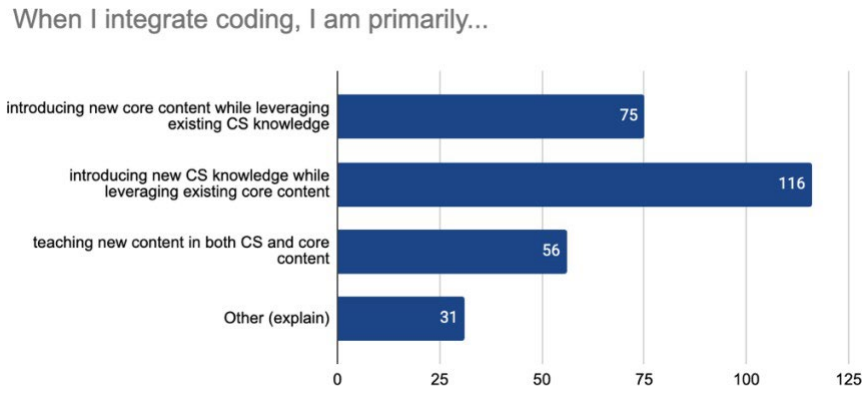
Considering that the majority of these are classroom teachers, BootUp placed a heavy emphasis on integration in their 2024-25 workshops. Integration refers to lessons in which coding is combined with and utilized in other subject areas. For example, children creating a simulation to demonstrate a scientific model in Scratch, using ScratchJr to create an interactive story for language arts, or making arithmetic games for math. On average, teachers reported teaching between 5 and 9 integrated lessons throughout the year. While these took place across the curriculum, they were most often taught in language arts, science, and mathematics.



"I think that learning code can be an engaging way for students to learn about any subject. I also think that the BootUp curriculum teaches perseverance and problem-solving skills that can be leveraged in future learning."

— 3RD/4TH GRADE TEACHER

Digging into integration further, we see that teachers most often leveraged existing subject-specific content while introducing new coding concepts.



Inclusivity

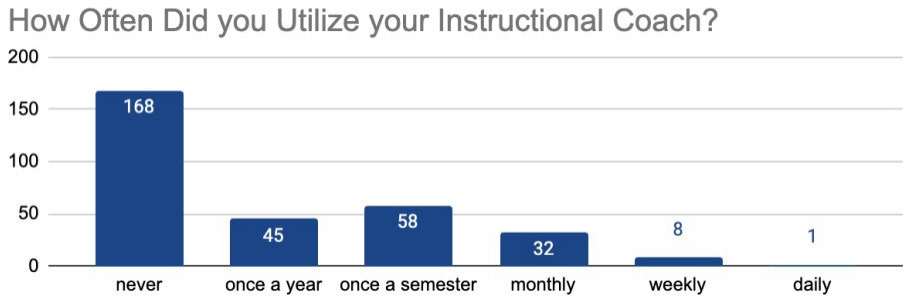
We also asked teachers what they did to make coding **inclusive for every child**. Teachers utilized a bevy of different strategies, showing creativity and variety in the way they try to involve all students. The most common of these are shown in the table below. Note that three of the most common strategies simply involve teaching everyone or not doing anything. This could point to an opportunity to prepare teachers on how to reach all students through coding.

Strategy	# of Teachers	Representative Quote
teaching all students	51	I am making sure all students are able to participate when activities are taught whether it is with laptop or IPad.
differentiation	34	Differentiated the lessons so that all students can adapt and learn how to code.
nothing	26	I have not done anything particular.
collaboration	25	The best way to make coding inclusive for students is to collaborate! Open the opportunity of collaboration with others. Another is to break up large tasks into smaller, manageable tasks.
integration	19	I have integrated coding with other subject matter such as ELA and math
n/a	10	N/A
visual aids	10	I have started implementing more visual aids, hands on modeling, and guided steps to accommodate diverse learning needs and ensure equitable access to coding.
Other strategies	< 10	Scaffolding (9), modeling (9), native language support (9), student interests (8), exposure (7), make it fun (7), creative freedom (6), time (6), after school club (5), resources (5), robots (5), teacher knowledge (5), teaching specific students (5), 1-on-1 (4), encouragement (4), accessibility tools (4), Friday fun day (3), make it easy (3), pacing (3), teaching resilience (3), translations (3), unplugged (3), coding last (2), decomposition (2), examples (2), implementation (2), micro:bits (2), personalization (2), teaching aides (2), universal design for learning (2), broad practice (1), choice (1), coding in free time (1), cognitive adaptations (1), discovery (1), grading (1), learning styles (1), motivation (1), projects (1), make it relevant (1), simplify it (1), small groups

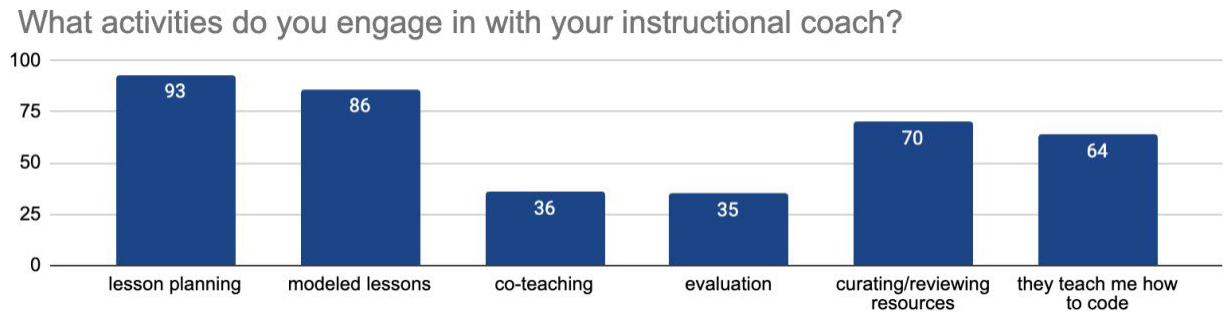
Supporting Teachers

In this sub-section, we look at efforts to support teachers and their effectiveness in enabling them to teach coding.

A key component of BootUp's PD model is the establishment and utilization of **Instructional Coaches** (ICs). In the BootUp context, ICs are experienced teachers who specialize in technology integration and coding. Each partner district identifies and designates ICs that can support teachers through a variety of means. Curiously, 54% of teachers reported that they never utilized their IC during the 2024-25 school year.



When teachers did use their IC, they engaged in a variety of different activities. In fact, teachers who worked with an IC often reported participating in at least three of the following activities.

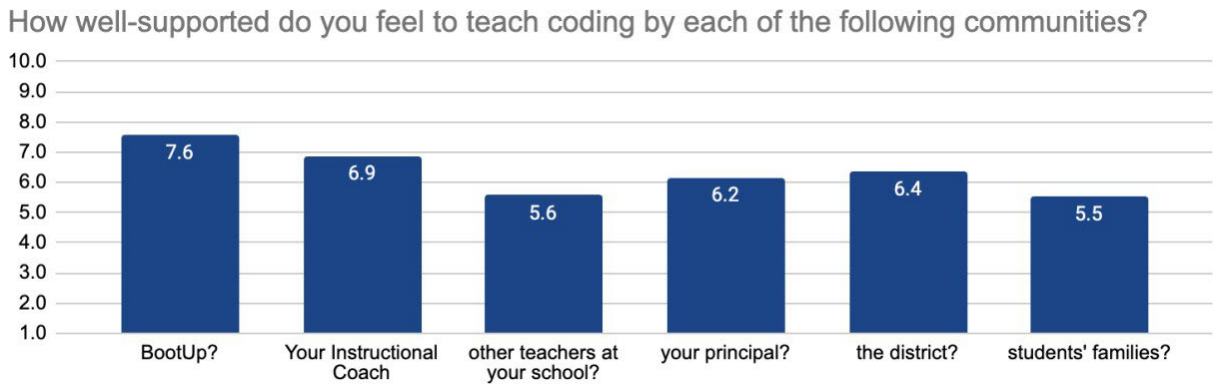


The most common activities were to plan and model lessons. A model lesson involved the IC teaching the lesson in the teacher's classroom. This enabled teachers to get a better idea of how to actually implement what they had learned and to respond to student reactions. The least-common activity with an IC involved evaluation, demonstrating that the role of the IC is less about judging a teacher's performance and more about supporting it.

"I have been trying to get all students to explore with different coding sources like scratch and code.org ....they had a great time creating their own music during an activity with code.org. the students had gallery walk where they could got to each persons laptop and play the music their classmates created."

— 3RD GRADE TEACHER

In looking more closely at teachers feelings, we asked them to indicate, on a scale of 1-10, how well-supported they felt by different colleagues and stakeholders in their teaching community.



Teachers felt the most well-supported to teach coding by BootUp, though the rating is in the "acceptable" rather than "good" or "great" range. They next felt best supported by their ICs. But in general, teachers reported not feeling super well-supported by many in their teaching community. Surprisingly, they felt least supported by other teachers and students' families. This may speak to other issues of isolation that classroom teachers might otherwise be feeling. It should be looked into in order to better support teachers moving forward.

"I had just received a new student and he had some behavioral issues- hard time focusing and completing projects. We were in our robot/hands on tech unit using Makey Makey. Our project entailed creating a scene/story using scratch and their Makey Makey coding blocks. This student was instantly engaged and went above and beyond the project requirements. He incorporated advanced coding blocks and as the weeks went on he continued to request to work in scratch - even instead of participating in our Fun Friday free time period! This was such a wonderful inspiration and makes teaching coding such a great experience. You never know when it can touch a student in such a positive way."

— STEM/STEAM TEACHER

Implementation Challenges

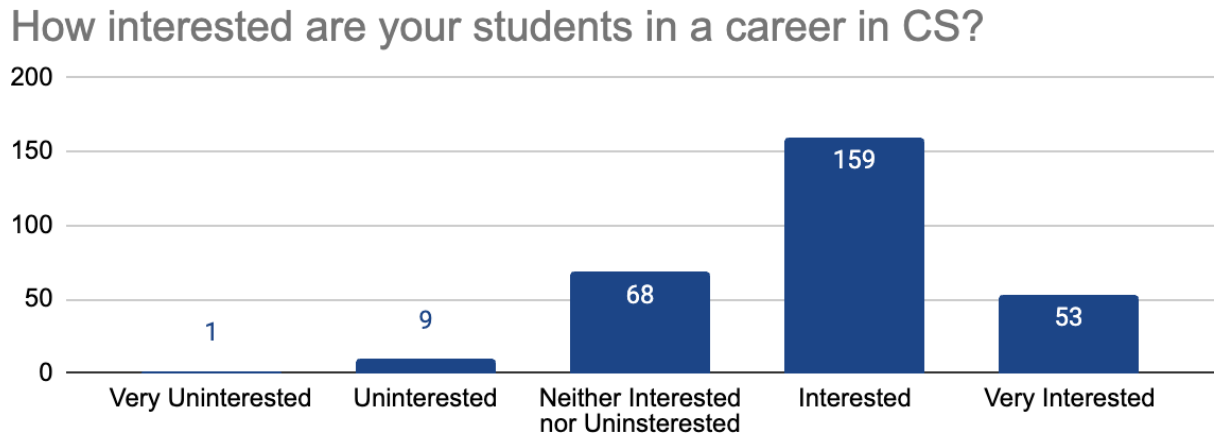
Teachers likewise reported on the various challenges they faced when trying to implement coding in their own classrooms. As has happened in prior studies, ***time*** remains teachers' primary barrier, outpacing the next-largest barrier by nearly 2.5x. Other primary barriers include resources, teachers' own knowledge and students' knowledge. It should be noted that two of the top responses to this question were that teachers did not face any barriers, accounting for about 10% of teachers. The full transcript of teachers' barriers is located in Appendix J.

Barriers	#	Representative Quote
time	102	The main barriers I experienced in implementing coding in the classroom were limited time and a packed schedule.
resources	42	It is difficult to get scratch jr. on the computers. we have a limited number of devices in the classroom.
none	22	No real barriers. Most of my students are excited about coding and love learning it.
teacher knowledge	22	Because I am not knowledgeable in blocks such as functioning, etc. teaching new content in Scratch became challenging.
student knowledge	17	Our scholars are emerging learners so reviewing content while coding was somewhat challenging. There were varying ability levels and independence that were a bit challenging but it wasn't anything would couldn't overcome after reflecting and adjusting instruction with guidance and support.
n/a	12	N/A

STUDENT RESPONSE

This section provides a few insights into how teachers' students responded to learning to code

While the purpose of this report is to show how teachers changed due to their BootUp training, teachers provided some insight into how their students dealt with learning to program. When asked how satisfied teachers thought the majority of their students were with the BootUp curriculum, they scored 4.2 out of 5. What's more, according to their teachers, most students are quite interested in pursuing a career in computer science or in which they would use coding. This reinforces the value that many teachers saw learning to code as professional/economic.



"I think coding is awesome! It teaches creative problem solving, teamwork, algorithmic thinking, and so much more. So often, school subjects are taught as rote memorization or having set answers/steps. Coding opens up space for different experiences and knowledge to be leveraged, and often, students who struggle with "academic" subjects are quite successful in programming."

— 4TH GRADE TEACHER

Surprise! Suprise!

We asked teachers to tell us of a time when their students surprised them as they were learning to code. Teacher responses were revealing of the student experience (in a good way), sharing heartwarming accounts. The most common of these responses are included in the following table.

Student Surprises	#	Representative Quote
student knowledge	110	I found that most of my students were able to debug on their own and that surprised me the most. Another time was when a low-achieving student was telling the class how to debug in a class discussion about Scratch Jr.
struggling/unexpected student succeeds	59	"I had an EL student who hardly talked in the classroom. When I called her name in front of the class, the rest of the students said, ""She doesn't talk."" I told the class to give her more chances, and later, I found out she was a good coder, better than most of her classmates. A 1st grader, Able often struggled with focus and behavior. But during a ScratchJr coding lesson, he quietly created an animation where a lion knocks over a monkey's blocks, said sorry, and helps rebuild. Jayden shared, "I made this because I knocked over Jacob's blocks yesterday." Through coding, Able found a way to express empathy and say sorry—something he hadn't been able to do with words alone."
drive/initiative	40	After we completed a couple of projects, a few have been able to code and create very detailed projects. The blocks they added to extend the coding we started with were great. It encouraged others to add more and see what else they could create.
student interest/engagement	30	I love seeing each student saying "WOW" that is sooooo cool! That is always my highlight.
collaboration	19	Students surprised me with how engaged they were in ALL math practices while working on a coding problem. Collaboration and critical thinking were off the charts.
none	19	Nothing I can remember
student creativity	15	Students are creative. They come up with different surprises as they find ways around coding
other surprises	< 10	debugging (9), students teaching me (6), resilience (5), family support (2), abstraction (1), coding is intuitive to students (1), hands-on coding (1), problem solving (1), unsure (1)

Teachers were most surprised by students' ability to understand and grow in their knowledge of coding. They likewise shared dozens of stories of students who they otherwise did not expect to succeed or who struggled in other subjects excel in coding. A full transcript of these responses is available in Appendix E.

Student Study

In this subsection, we share the results from students' cognitive and affective assessment scores

Teachers in District 17 and in several other schools, participated in a study to examine the measurable cognitive and affective effects that learning to code was having on their students. In the first half of the year, teachers administered the following validated assessments to their students.

- **Computational Thinking test (CTt)**: A 28-item test that measures students' ability to think computationally by solving a series of 4 questions around 7 sets of different problems, focusing on sequencing, conditional logic, and functions. This is the most used test worldwide to measure elementary school students' computational thinking abilities. Takes ~30 minutes to complete.

- **Elementary Student Coding Attitude Survey (ESCAS)**: A 32-item survey with 6-point Likert scale statements that measure students' attitudes for coding. Namely: confidence, interest, utility value, perceptions of coders, math self-efficacy, and social persuasion. Has been validated with 1000s of prior BootUp students and used worldwide in a variety of contexts.

To analyze students' scores, we combined all tests for all students into a single file. We also included demographics and potential explainer variables to be able to identify trends. In all, we have results from 844 students across 34 different teachers.

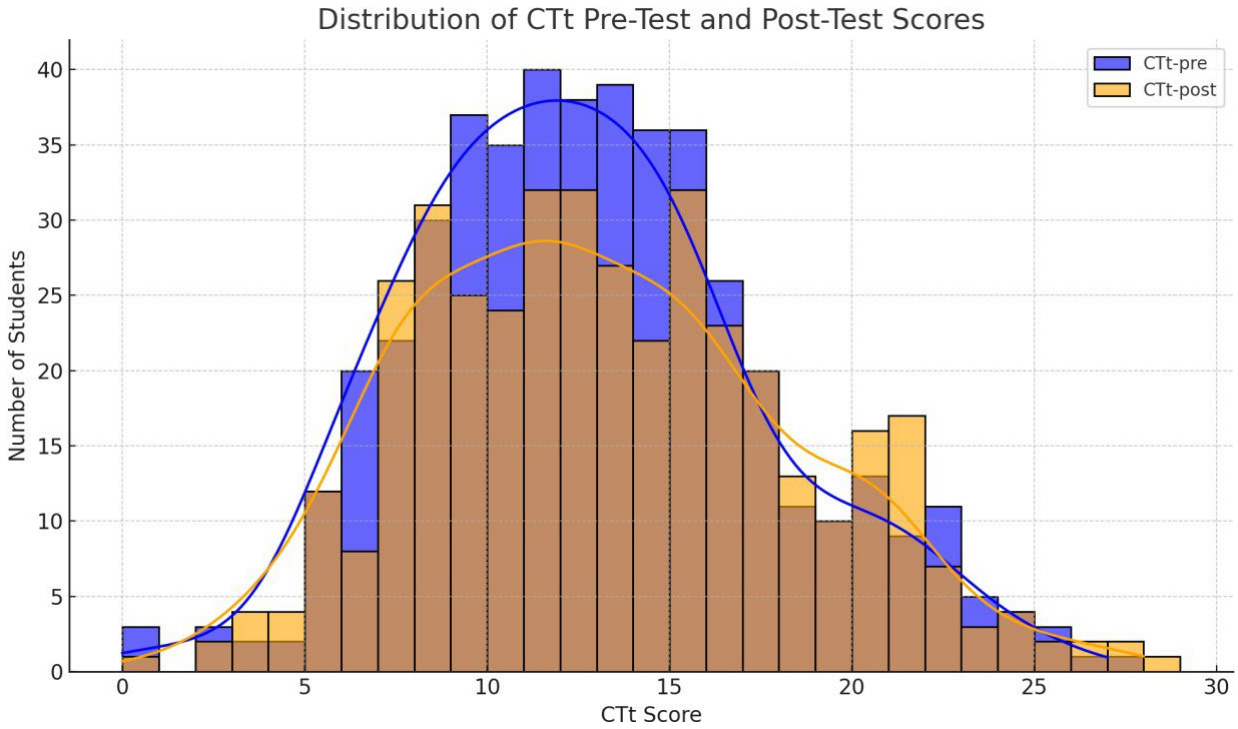
CTt Results

The following table breaks down students' CTt scores across grade by grade. 5th grade teachers were the most likely to administer the CTt. A total of 204 students completed both the CTt-pre and CTt-post exams.

Test	grade	N_students	Avg_score	Std_dev	N_girls	N_boys
CTt-pre	3	64	11.7	3.9	30	31
	4	109	11.4	4.7	50	57
	5	253	13.2	4.9	114	126
	6	43	13	5.4	19	24
CTt-post	3	56	12.3	4.4	29	24
	4	144	12.5	5	79	58
	5	201	13.5	5.4	93	98

A recent analysis of the CTt as it has been implemented worldwide gives context to these scores. In that study, the average CTt-pre scores were 11.4, 12.4, and 12.8 for 3rd-5th grades, respectively. The average post scores were shown to be 12.9, 14.2, and 14.9, respectively. Consequently, students' pre-test scores appear to be in-line with international expectations. Post test scores do not appear to have increased as much as with international comparisons. However, the standard deviation puts these scores well within range of international trends for CTt-post ranges.

The following chart provides a visualization of CTt-pre and CTt-post test scores overlaid.



A Man-Whitney U test to compare means of pre and post-test scores show no statistically significant difference (U = 19,225; p = 0.404).

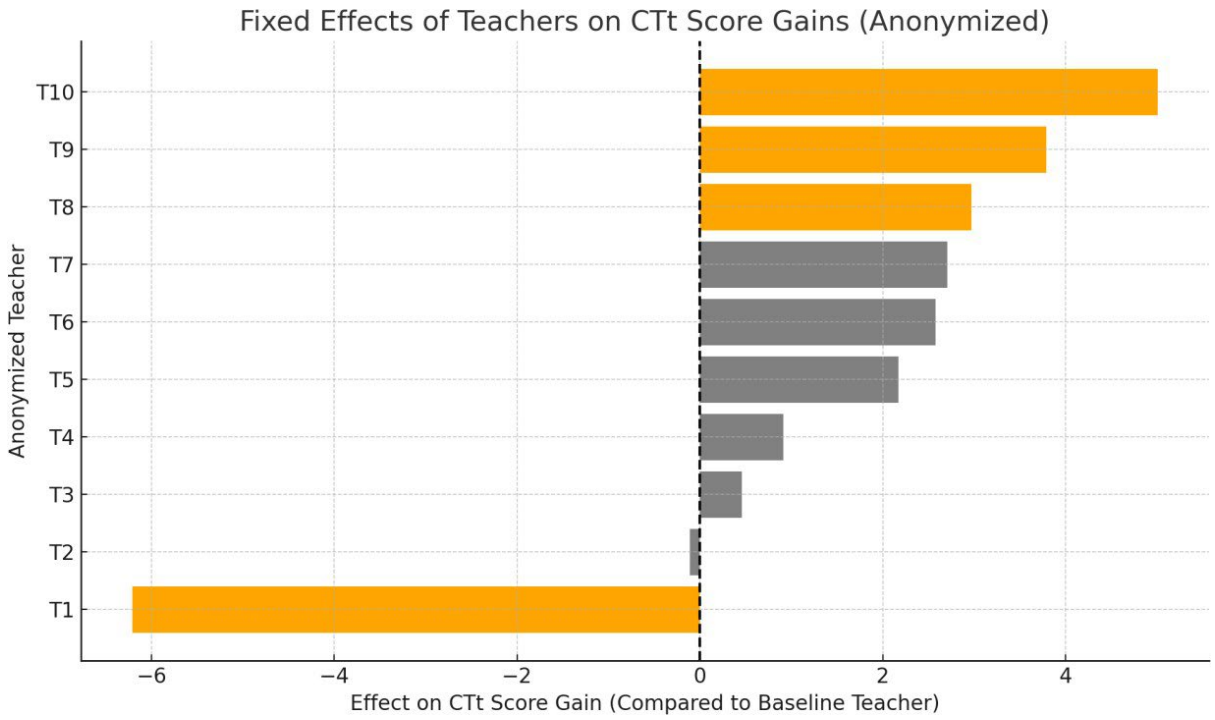
Gender Comparisons

The following table presents student scores for those who completed both the pre and the post test by gender. A Wilcoxon Signed-Rank test revealed that boys made significant gains (W = 1256, p = .0014) while girls did not (W = 1535, p = 0.1985).

gender	N_students	CTt_pre_mean	CTt_post_mean	CTt_pre_std	CTt_post_std
Boy	99	13	14.4	5.1	5.4
Girl	101	12.2	11.9	5	4.9
Other	2	10	15	1.4	1.4
Prefer not to answer	2	9.5	17.5	3.5	2.1

Predictors

To determine if there were latent factors affecting these changes, we ran a multiple regression model was to predict gains in students' computational thinking scores (CTt-diff) based on prior ability, demographics, coding background, and teacher assignment. Teachers were modeled as fixed effects to allow for direct comparisons between individual teachers and to test whether specific teachers were associated with significantly higher or lower student gains. Teachers with significant gains (positive or negative) are indicated in orange in the following chart.



The full statistical model is included below

$$
\begin{aligned}
\text{CTt-diff} = 4.004 \\
&- 1.328 \cdot \text{Gender}_{\text{Girl}} \\
&+ 5.854 \cdot \text{Gender}_{\text{Prefer not to answer}} \\
&- 0.888 \cdot \text{Grade}_4 + 1.921 \cdot \text{Grade}_5 \\
&- 0.565 \cdot \text{Experience}_{\text{2–3 yrs}} + 0.020 \cdot \text{Experience}_{\text{None}} + 1.201 \cdot \text{Experience}_{\text{<1 yr}} + 1.056 \cdot \text{Experience}_{\text{>3 yrs}} \\
&- 1.024 \cdot \text{Freq}_{\text{Once/month}} - 1.632 \cdot \text{Freq}_{\text{Once/week}} + 0.014 \cdot \text{Freq}_{\text{Daily}} \\
&- 0.826 \cdot \text{Freq}_{\text{Never}} - 2.663 \cdot \text{Freq}_{\text{Every other week}} \\
&+ 5.010 \cdot \text{Teacher}_{\text{Dolin}} + 3.788 \cdot \text{Teacher}_{\text{Einstein}} + \cdots - 6.211 \cdot \text{Teacher}_{\text{Swartzman}} \\
&- 0.433 \cdot \text{CTt-pre}
\end{aligned}
$$

The model explained 48.3% of the variance in CTt gains, meaning it accounted for nearly half of the differences in how much students improved from pre- to post-test**. In practical terms, this suggests the included predictors—especially prior ability and teacher identity—were strong indicators of which students made more progress.** Lower CTt-pre scores predicted greater gains, likely reflecting room for growth. Gender differences were not statistically significant after excluding students who identified as "Other" (due to small sample size). Neither students' prior coding experience, coding frequency, nor self-reported exposure to coding instruction significantly predicted CTt gains. These findings highlight the central role of teachers and students' starting points in driving improvement in computational thinking.

Summary table of key predictors.

Predictor	Coefficient (β)	p-value	95% CI	Interpretation
Intercept	4.00	0.010	[0.98, 7.03]	Baseline gain (reference group)
CTt-pre	-0.43	<0.001	[-0.58, -0.28]	Higher pre-score → smaller gain
Gender (Girl)	-1.33	0.072	[-2.77, 0.12]	Not significant; trend toward lower gains
Teacher (various)	See figure	<0.05*		Some teachers had significantly different gains
R^2	0.483			Model explains ~48% of variance in gains

ESCAS results

In total, 527 students completed the ESCAS-pre and 527 students completed the ESCAS-post (not necessarily all the same students). Of these, 269 completed both administrations. As with the CTt, we first compared students who took only one administration with those who completed both. Because there were no statistically significant differences between their scores, we include all data when making pre/post comparisons.

The following table reports on students' average ratings for each of the ESCAS attitudes.

Attitude	Pre Mean	Pre SD	Post Mean	Post SD	U Statistic	p-value	Effect size (*r*)
Confidence	4.233	0.88	4.23	0.927	138788	0.988	0.001
Interest	4.451	1.189	4.075	1.373	160554.5	< .000*	-0.156
Utility	4.161	1.042	3.97	1.09	153320.5	0.002*	-0.108
Perceptions	3.977	0.87	3.868	0.85	149848	0.019*	-0.083
Math	4.515	1.345	4.519	1.291	139437.5	0.822	-0.008
Social	4.17	1.006	4.119	1.038	141102	0.501	-0.024

The first thing to note is that, ***these attitudes are all positive*** according to the way attitudes are measured on the ESCAS. Students are presented with statements and asked the degree to which they agree with those statements (1 = strongly disagree, 6 = strongly agree). Anything above 3.5 indicates positive agreement (4 = somewhat agree). The following five statements make up the "Interest" sub-scale:

I1. I like coding, or I think I would like coding.
I2. I would like to learn more about coding.
I3. Solving coding problems seems fun.
I4. Coding is interesting.
I5. I would like to study coding in the future.

Despite starting and ending with positive attitudes, students' attitudes for interest, utility value, and perceptions all declined significantly. However, the practical effect (measured by effect size) are negligible (< .1) to small (.1 < *r* < .3). Furthermore, when we examine only paired scores (i.e., scores of students who took both tests), only interest remains statistically significant ($p < .001$).

Gender Differences

To determine if gender played a role, we now examine whether boys and girls differed on their attitudes for coding. The following table compare students' pre-test attitudes, post-test attitudes, and changes (i.e. *diff*) by gender. We found no difference by other demographics (i.e., race and ethnicity).

In looking at the table below, we see that boys and girls differed significantly on every sub-scale on both the pre and the post-tests. Boys generally had more favorable attitudes by .2-.3 points on each scale. The effect sizes (r) are mostly small (except interest, which is on the small-medium threshold). However, there were no statistically significant differences between boys and girls on any of the changes they experienced. Thus, we might conclude that the gender differences were pre-existing and that the degree of that distance remained the same throughout the school year.

Attitude	Boy Mean	Girl Mean	U Statistic	p-value	Effect size (*r*)
Confidence (Pre)	4.307	4.156	35727	0.0218*	-0.118
Confidence (Post)	4.378	4.047	37749	0.0001*	-0.203
Confidence (Diff)	0.099	-0.027	9188	0.1736	-0.098
Interest (Pre)	4.631	4.255	38937.5	< 0.001*	-0.218
Interest (Post)	4.402	3.699	40994.5	< 0.001*	-0.307
Interest (Diff)	-0.226	-0.451	9529	0.0538	-0.138
Utility (Pre)	4.285	4.05	36718.5	0.0028*	-0.153
Utility (Post)	4.117	3.8	36933	0.0004*	-0.182
Utility (Diff)	-0.059	-0.155	8817	0.4569	-0.053
Perceptions (Pre)	4.071	3.891	35926	0.0158*	-0.124
Perceptions (Post)	3.983	3.742	36113.5	0.0019*	-0.16
Perceptions (Diff)	-0.139	-0.041	8268	0.8658	0.012
Math (Pre)	4.644	4.389	35878	0.0099*	-0.131
Math (Post)	4.674	4.345	36527	0.0013*	-0.164
Math (Diff)	-0.085	0.101	7606.5	0.1896	0.091
Social (Pre)	4.305	4.025	36526	0.0031*	-0.152
Social (Post)	4.303	3.886	38744.5	< 0.001*	-0.245
Social (Diff)	-0.021	-0.047	8638.5	0.6547	-0.032

Relationships between Coding Attitudes and Computational Thinking

Finally, we examine the relationship between students' cognitive and affective states. The following table compares the relationship using a Pearson correlation on the pre, post and differences in attitudes.

Attitude	Pre		Post		Diff (change)	
	Correlation (r)	p-value	Correlation (r)	p-value	Correlation (r)	p-value
Confidence	0.247	< .001	0.247	< .001	-0.056	0.4614
Interest	0.267	< .001	0.267	< .001	-0.074	0.3338
Utility	0.178	0.0009	0.178	0.0009	-0.112	0.1428
Perceptions	0.052	0.3343	0.052	0.3343	-0.117	0.1257
Math	0.187	0.0005	0.187	0.0005	0.072	0.3434
Social	0.207	0.0001	0.207	0.0001	-0.025	0.742

Curiously, there was a statistically significant relationship between each attitude on both the pre and post surveys except for with perceptions. Thus, children's perceptions of coders (i.e., their biases) do not correlate with their own CT ability. That being said, these correlations tend to be small to medium. There were no correlations in the change in attitudes and students' CTt change scores.

As with the CTt, we sought to determine if there were other, latent, factors that might explain these differences. To that end, we ran a fixed effects regression model predicting CTt-post scores using students and teacher characteristics (gender, coding experience, coding frequency. The model showed strong predictability (R2 = 0.577, Adj. R2 = 0.519), revealing the following key results.

Predictor	Coefficient	p-value	Interpretation
CTt-pre	+0.550	< .001	Strongest predictor: higher pre-score → higher post-score
Gender (Girl)	–1.451	0.020	Girls scored significantly lower on CTt-post
Social-post	+0.870	0.014	Higher perceptions of what peers and parents think about coding was associated with higher CTt-post scores.
Other attitude variables	Not significant	> 0.23	Confidence, interest, utility, perceptions, math had no clear effect

The strongest predictor of change in attitude was, once again, students' CTt-pre scores.

Student Struggles

This sub-section discusses teachers' strategies that they employ with students who struggle with coding

We asked teachers to tell us of times when (a) student(s) struggled with coding and how they addressed it. Teachers provided a slew of different strategies for dealing with times when students had difficulties with coding. There was possibly more variety in the response to this question than in almost any other open-ended question, demonstrating that teachers employ a wide variety of strategies for dealing with struggling students. The most common strategies are in the table below.

Strategy	#	Representative Quote
collaboration	54	Some of the students had difficult with deleting some of their coding or spirt and some of their coding did not work. We would discuss it as a class and some of their classmates jump to the opportunity to help them. I noticed that more students were willing to help each other and ask each other how to do some of the coding.
resilience	25	I purposely let my students struggle for a bit while they are working on any programming project. That productive struggle is an essential learning tool. Students must utilize all possible hints provided by program, ask a should partner or student coding leader first before they can ask for my help. That is a process that is a foundation in my class learning environment because I want my students to become researchers to find solutions. It is not something I learned from BootUp but I have found it is implied in the lessons but It is also something I have learned over my 26 years of teaching.
debugging	20	A student got stuck during their project based on my training I was able to help them identify the error and correct the code
modeling	20	My students struggled with using the circle tool to draw part of the fidget spinner It was difficult to hold their trackpad with one hand and then to drag their finger on the other hand to create the circles. I modeled how to use the trackpad with two hands to draw their circles and after some frustration they worked their way through it.
teaching aides	20	The biggest struggle was with confusion with how turns work. To help my students I created visuals of the arrows and helped them practice planning their robot's routes along the mat routes or on a pocket chart. This way students can see each other's code and help each other debug.
decomposition	16	A student struggled with coding because they couldn't understand the concept of it (you tell the object what to do). So, we broke it down step-by-step and had just on sprite do one action, instead of focusing on the whole project with multiple sprites and multiple actions.
1-on-1	13	Students struggle often with coding. It is usually necessary to sit one on one with them to figure out the issue.
teacher knowledge	12	We all creatively struggled in implementing the Scratch program into the ELA content at first. Scratch was very new to our students. Through consistent teacher modeling, our students were able to grasp the fundamentals of coding to complete their Scratch projects. I am also learning how to code and teach my students using the Scratch program. Bootup training supported my own knowledge of Scratch through each professional development training that I attended. I was able to ask Brenda, Brett, or Miki for help in how to effectively teach coding in Scratch to my students.

Strategy	#	Representative Quote
none	10	I did not experience any of my students struggling when I taught them how to code.
Other strategies	< 10	guiding questions (9), review (9), discover (9), student understanding/knowledge (9), differentiation (5), examples (5), redirecting (5), robots (5), write it out (5), scaffolding (3), student interest (3), attention (2), direct instruction (2), follow along (2),, instructional coach (2), language (2), planning (2), social persuasion (2), tips (2), unplugged activities (2), abstraction (1), act it out (1), agency (1), attitude (1), chunking (1), connections (1), decomposition (1), encouragement (1), games (1), heuristic (1), multiple chances (1), reading comprehension (1), rephrasing (1), resources (1), technical problems (1), tutorials (1)

Definitions and explanations for some of the more common strategies follow:

- **Collaboration**: Teachers' most common strategy was to have students collaborate with each other in some way. This included paired programming, asking a peer, or asking a successful student to explain it in their own words.
- **Resilience**: Another common strategy was to encourage students to develop resilience. Sometimes this occurred intentionally by engaging students in "productive struggle," while other times, teachers taught resilience by encouraging students to be patient with themselves or to take a step back, take a breath, and go at it again.
- **Debugging**: These strategies involved helping students to identify and resolve problems with their code. Teachers often described working alongside the student to highlight the problem and to come up with a potential resolution.
- **Modeling**: A strategy that involves the teacher demonstrating to the student how to code a particular solution to a problem. These were not meant to provide a follow-along, "click here" response, but rather to show students strategies they might use in a similar way to solve the problem.
- **Teaching aides**: The creation or usage of additional materials in order to guide students through problems. Teachers described creating visuals, using lesson plans, finding helpful videos, etc.
- **1-on-1**: Sitting with an individual student to help resolve their problems.
- **Teacher Knowledge**: The teacher described increasing their own understanding of coding as a way to better help students solve their problems.

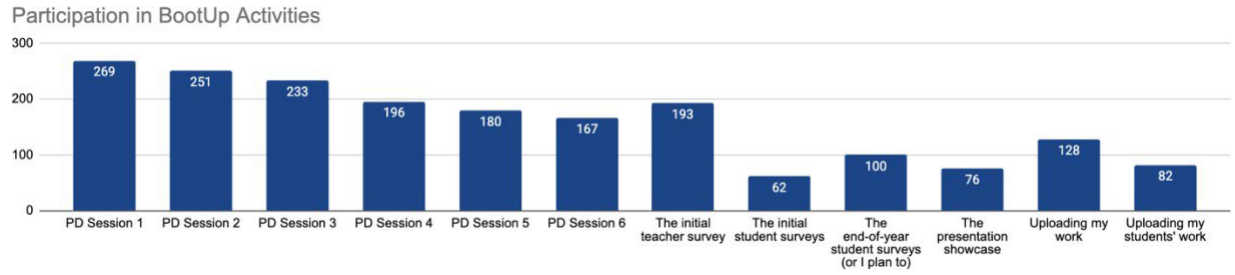
The full transcript of these strategies is available in Appendix F.

BOOTUP EVALUATION

This section presents teachers' evaluation of their BootUp PD experience

Participation

In the 2024-25 group, teachers attended 4.1 BootUp workshops on average. In addition to workshops there were several other activities they could participate in. The following chart shows which of these activities teachers reported engaging with overall.



Rating

We asked teachers, "Knowing what you know now, would you participate in BootUp again?" Only 4 teachers (1.6%) indicated "no". This seems like a fairly strong endorsement of the program.

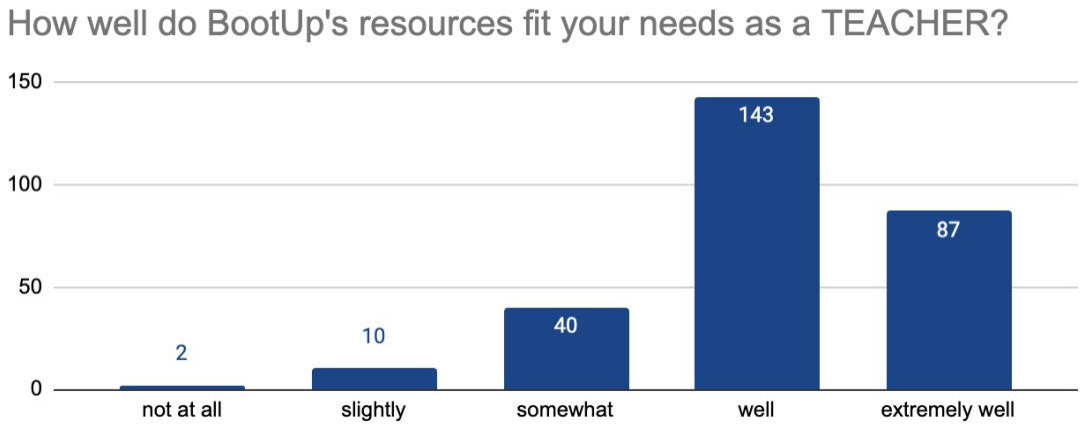
A common metric for measuring success is the **Net Promoter Score (NPS).** NPS is calculated by asking a single question, "*On a scale of 0-10, how likely are you to recommend BootUp PD to a colleague or peer?*" Ratings are split into 3 groups: (a) promoters: 9 – 10, (b) passives: 7-8, and (c) detractors: 0-6. The percentage of detractors is then subtracted from the percentage of promoters. The **overall 2024-25 NPS was 58%.** When averaging all ratings (including passives), **the average score was 8.8/10**.

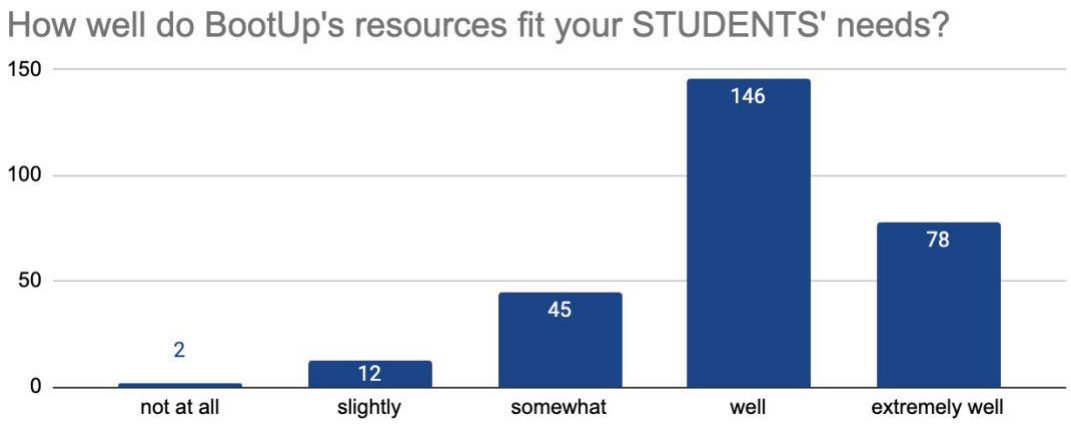
Breaking down the BootUp experience further, teachers rated each of its key components on a scale of 1-10. Teachers enjoyed the hands-on aspect of BootUp the most. They also highly appreciated the materials BootUp makes available to aid in teaching. Modeling (i.e., seeing a BootUp facilitator demonstrating how to teach specific lessons) and the regularly, monthly meetings, were also popular. Collaborative activities were slightly behind, though still popular.

The one element that was not as well-received was the showcase, an exposition in which teachers demonstrate programs that they have created to other teachers.

BootUp Element	
Hands-on Learning	8.64
Coder Resource	8.34
Lesson Plans	8.15
Model Teaching	8.13
Monthly PD Sessions	8.13
Sharing Projects/Ideas	7.91
Paired Programming	7.46
Showcase	6.64

We likewise asked teachers how well they believed BootUp's resources fit both teacher and students needs. The charts below demonstrate that most teachers thought these fit well or very well. This is consistent with their ratings of resources above as well as the way they relied on BootUp resources to prepare lessons.





Teachers shared many ways in which they translated what they learned in the BootUp workshops to their classroom practice. Comments such as the following were common:

"I think that my Bootup professional development training in coding with BeeBots Robots, micro:bit, (soon to be Climate Control Kits), and Scratch effectively supported my instructional planning development. I was able to transcend the knowledge that I learned from attending my Bootup PD sessions and apply it to my classroom instructional activities with my students."

— 3RD GRADE TEACHER

While the following comment is a bit long, it exemplifies many of the ways in which teachers saw the benefit of BootUp in their classroom and overall experience. All responses are in Appendix G.

I believe that my BootUp professional development translated very effectively to my classroom practice. One of the key takeaways for me was the emphasis on student-led learning. I loved how the training highlighted the value of collaboration, both between students and between students and teachers. This approach has made a significant difference in how I facilitate coding lessons in my classroom.

For example, I started incorporating more group projects where students work together to tackle coding challenges. I noticed that when students collaborate, they not only learn from each other, but they also develop important teamwork and communication skills. This collaborative environment encourages them to take ownership of their learning, and I've seen their confidence grow as they share ideas and support one another in problem-solving.

Another important lesson from BootUp was the understanding that I don't need to know everything about coding to be an effective teacher. This realization was liberating! I learned that I can embrace my role as a guide rather than an all-knowing expert. The various coding programs we explored during the training, which come with ready-made lessons, have provided me with a wealth of resources to draw from. Not only do these programs offer structured lessons, but they also include visual examples that help students grasp complex concepts more easily.

As I implemented the strategies and resources from BootUp, I made sure to use visuals extensively in my lessons. This has helped all my students, but especially those who benefit from seeing content presented in diverse ways. For instance, I created visual aids and used screenshots from the coding platforms to demonstrate specific coding blocks and their functions, which made it easier for my students to understand how to apply what they were learning.

Overall, I feel more empowered as a teacher because of the BootUp training. I now approach coding instruction with a more open mindset, understanding that the journey of learning to code is a collaborative one. My students are thriving with this new approach, and I'm excited to continue building on these foundations in the future!

— 6TH GRADE TEACHER

The BEST Things

We asked teachers what was the BEST thing about their BootUp PD experience. The most common response was the opportunity to collaborate, share with, and hear from their peers, whether from their own school or another. Teachers also appreciated gaining new knowledge, the hands-on nature of the training, the way students responded to learning to code, and the BootUp facilitators themselves (often calling them by name). The full transcript of responses is available in Appendix H.

The Best Thing	#	Representative Quote
collaboration	64	The BEST thing about this experience with BootUp is that I was able to work with other teachers and network. I loved that I can reach out at any time to have a presenter teach a lesson with my class.
new knowledge	49	I think the best thing about this experience is being able to learn and use scratch and beebots. Initially, I did not know about all this programs but I think having these programs is really helpful for the students to practice coding.
hands-on	43	Getting hands on experience with both Bee Bots and Scratch Jr--its the best way to learn. Also the classroom visit was so great. The scholars had a blast learning new things to do with Scratch!
resources	33	Receiving already made resources
student response	27	Experiencing the excitement of the students to learn and implement coding techniques into their projects and increased parental involvement/support.
facilitators	20	Attending the BootUp training is great! It made me confident about coding. It helped me realized that coding is something that I should not be scared because there are facilitators that are willing to guide and assist people who want to learn.
increased confidence	14	the BEST thing is the empowerment and confidence gained because of Bootup to effectively teach coding to all students, regardless of their own prior coding experience.
physical computing	13	I loved getting a classroom set of Micro:Bits. The students LOVE using them.
Other “best things”	< 10	engaging (7), practical (6), modeling (5), everything (4), in-person (4), structure (4), classroom visits (3), pd (3), relevant (3), support (3), 1-0n-1 (2), creativity (2), instructional coach (2), multiple formats (2), showcase (2), trainers (2), trainers (2), variety (2), virtual meetings (2), information (1), low-risk environment (1), meeting experts (1), nothing (1), strategies (1), time for planning (1)

Everything was honestly the best for me. I was surrounded by CS Teachers. I am not a CS Teacher, but I never felt ostracized because of that. I felt that everyone was very helpful to me by explaining basic and more advanced CS concepts so that I could understand. I really loved my Bootup course experience!

— 3RD GRADE TEACHER

Recommended Changes

This subsection discusses changes that teachers recommended in order to improve the BootUp experience

Finally, we probed teachers to know in what ways they would change the BootUp experience in order to improve it. While there were common categories of responses, there was more variety in teachers' responses to this question than any other question asked in this evaluation. This shows that there are idiosyncratic things that affect individual teachers in different ways.

Perhaps the most telling response in teachers' recommendations for change was the most common response — nothing! In fact, the "nothing" response was so prevalent, that more teachers said they wouldn't change anything about their BootUp experience than all of the other top responses combined (by a fair margin, at that). This does not diminish the other, more singular, response. But it does speak to the fact that many teachers who completed the year-end survey enjoyed their BootUp experience enough to not want to change anything at all. Add to that the second most requested change was to have more PD, and it paints a picture of satisfied teachers.

Recommended Change	#	Representative Quote
nothing	74	I wouldn't change anything about this course. I felt supported throughout the entire year and was able to look through the resources to give me a better understanding of coding.
more pd	20	Make it more frequently or offer a second round of PDs.
more time	12	One thing I would change about the BootUp PD experience is having more time to collaborate with other teachers across schools to share ideas, troubleshoot challenges, and see how coding is being integrated in different subjects.
more in-person meetings	10	One thing I would like to change is having more in person meeting to use and practice coding.
more virtual meetings	10	I would give the teachers more opportunities to work together and I would definitely keep virtual sessions! I am a late start school and having the opportunity to meet virtually allowed me to participate in this wonderful PD!

There was an interesting tension between teachers' desire for in-person versus virtual meetings, which were equally requested. Those requesting more in-person meetings typically focused on the quality of the interactions, while those wanting more virtual meetings often cited its convenience in regards to scheduling and access.