Evaluation Summary Report 2015-2016

Overview **2**
A Note on Dates 2
Overall by the Numbers 2
Table: Overall descriptives of teachers and students using our courses 2
Code Studio Students by the Numbers 3
Breakdown of Code Studio Users and Usage (2015-2016) 4
2016 Hour of Code Impact Study 4
Table: Largest perception shifts by group 4

Elementary School and CS Fundamentals **5**
Elementary School Teachers 5
Table: Overall satisfaction with and preparedness from Computer Science Fundamentals workshops 5
Outlier Research & Evaluation Findings - Elementary School Teachers 5
Elementary School Students 6
Table: Number of elementary school students reached by Computer Science Fundamentals (K-5) 6
Defining “Basic Coding Proficiency” in Computer Science Fundamentals 7
Table: Number of students who have achieved proficiency January 1, 2016 - December 31, 2016 7
Outlier Research & Evaluation Findings - Elementary School Students 8

Middle School Programs **9**
Middle School Teachers 10
Table: Overall satisfaction with and preparedness from middle school professional learning workshops 10
Middle School Students 10
Table: Number of middle school students reached by program 11

High School Programs **11**
High School Teachers 11
Table: Overall satisfaction with and preparedness from high school professional learning workshops 11
Outlier Research & Evaluation Findings - High School Teachers 12
High School Students 13
Table: Number of high school students reached by program 14
Outlier Research & Evaluation Findings - High School Students 14

Evaluation Plans 2016-2017 **16**

Overview

Code.org engages in extensive evaluation of its programs to measure success and find areas for improvement, improve internal processes, meet donor obligations, and ensure that we are maximizing our potential to realize the goal of having computer science offered in all K-12 schools. To this end, we have collected data about our professional learning programs, our curricular programs, as well as metrics related to student engagement and learning to help answer the following key questions:

1. What impact does Code.org professional development and use of Code.org curricula have on teachers?
2. What impact does Code.org curriculum have on student proficiency and learning of computer science?
3. Is teacher and student participation in Code.org programs associated with changes in their attitudes toward and self-efficacy with computer science?

For the third question, Code.org hired Outlier Research & Evaluation at UChicago STEM Education, a third-party evaluator, to conduct a systematic investigation to determine if teacher and student attitudes toward and self-efficacy with computer science change after teaching or taking a computer science course. For findings generated from Outlier's evaluation, it is important to keep in mind that the data come from a subsample of Code.org classrooms. The findings provide valuable information about some of our computer science classrooms and help us to better understand student and teacher experiences with computer science, but they are not necessarily generalizable to all Code.org classrooms or all computer science classrooms more generally.

In addressing the questions above, this report provides high level statistics about course usage to date to provide context related to course usage, then it shares more detailed information about the individual programs. First, results from a 2016 Hour of Code study are shared. Then, programs are examined by grade level (elementary, middle school, high school) as a way to zoom in on the questions. For each section, highlights are shared and next steps are discussed. Finally, we share plans for evaluating our programs in the 2016-2017 school year.

A Note on Dates

Unless stated otherwise, the data collected and reported on here comes from professional learning workshops that were held from June 2015-May 2016, and from students who participated in courses or the online learning platform from August 2015-July 2016. The Outlier data area from the 2015-16 school year.

Overall by the Numbers

Code.org's programs have reached a large number of students and teachers. Here are some highlights:

Table: Overall descriptives of teachers and students using our courses

Code.org Goal	2013-2014	2014-2015	2015-2016
Engage teachers in our computer science courses[1]	34K new teachers	110K new teachers	210K new teachers
Engage students in our computer science courses[2]	1.3M new students	2.8M new students	7.3M new students

[1] Source: Number of new Code Studio teacher accounts created each year.
[2] Source: Number of new Code Studio student accounts created each year.

Improve racial diversity in computer science[3]	N/A	37% African American or Hispanic/Latino	49% underrepresented minority[4]
Improve gender diversity in computer science	N/A	43% female[5]	44% female

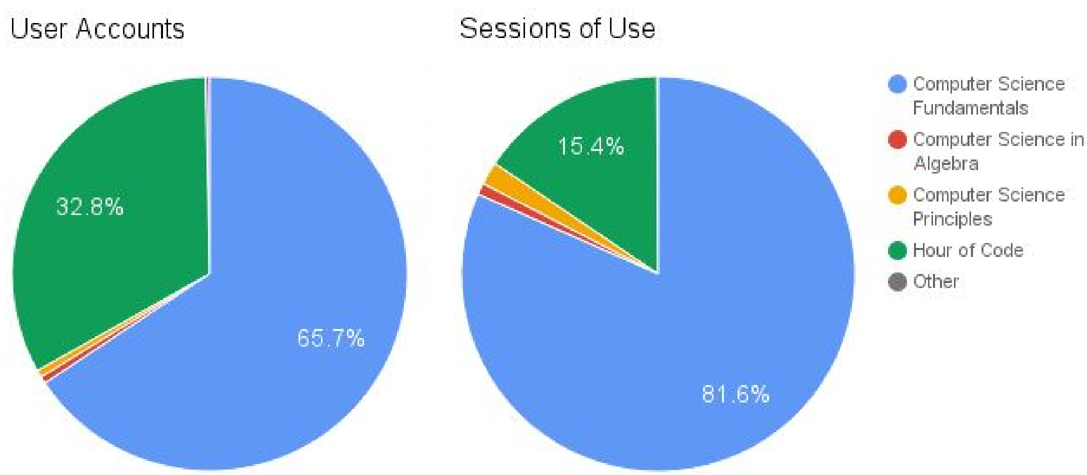
Note: Numbers reported are not cumulative

While these numbers are high, it is likely that our actual user base is larger than reported here. Because we do not require students or teachers to sign up or log in, many students and teachers use the platform without creating accounts, so they are not included in these numbers. Additionally, many of those who do create accounts subsequently use our curriculum without signing in; therefore we can not accurately account for those users. We estimate that 10-25% of the web traffic in our curriculum is from students engaging with courses without logging in - this number was calculated by looking at the number of page views of puzzles further into our Computer Science Fundamentals (K-5) curriculum. For the Hour of Code in the 2015-2016 school year, we estimate that approximately 80% of participants were not logged in.

Code Studio Students by the Numbers

Overall, Code.org evaluation focuses more on students engaging with our curriculum, and less so on those that just take part in Hour of Code activities. As of August 1, 2016, we had 11.5M student accounts on Code Studio. Seven million, three hundred thousand of those accounts were created in the 2015-2016 school year. Here is the usage breakdown.

Breakdown of Code Studio Users and Usage (2015-2016)[6]



[3] Source: Data gathered from surveys either sent to teachers or posted on Code Studio.
[4] An underrepresented minority in computer science is defined as a student who is Black, Hispanic/Latino, American Indian/Alaska native, native Hawaiian or Other Pacific Islander. Racial diversity data is from US classrooms only.
[5] Source: This percentage is calculated by examining all student accounts who have chosen to report gender as part of their student account.
[6] Note that students were not double counted here. If they completed scripts in multiple courses, we associated their usage in the following way. Computer Science Principles>Algebra>Computer Science Fundamentals>Hour of Code>Other. So, if someone completed scripts for Hour of Code and Computer Science Principles, they were only counted for Computer Science Principles.

2016 Hour of Code Impact Study
This pre- and post-survey study (N=8,040) aimed to assess whether or not students' attitudes towards and self-efficacy with computer science changed after engaging with an Hour of Code activity. ***Findings show that after completing one Hour of Code activity, students' positive attitudes towards computer science increase along with feelings of computer science self-efficacy, especially for female students. In other words, after just one Hour of Code activity, students report liking computer science more and report feeling that they are better able to learn computer science and are better at computer science than their peers.*** These findings suggest that using particular activities in the right context can produce large changes in attitudes towards and self-efficacy with computer science. The table below shows the largest perception shifts reported by students after taking part in just one Hour of Code tutorial. Note that all of these findings were statistically significant at the .01 level. Read the full report here.

Table: Largest perception shifts by group

Likert-Scale Item	Student Gender	Age Group	Prior Experience	N	Pre Hour of Code likert-scale rating (1-4)	Post Hour of Code likert-scale rating (1-4)	Absolute change in % who agree
I like computer science.	Female	High school	Never done an hour of Code	158	2.42	2.78**	20.3%
I am better at computer science than most kids at my school	Female	Elementary school	Only done an Hour of Code	289	2.35	2.61**	18.0%

Elementary School and CS Fundamentals

Computer Science Fundamentals is our introductory computer science curriculum for grades K-5. There are four courses that are part of our K-5 offering, and each course is comprised of a mixture of unplugged lessons and online puzzles. Course 1 is the most basic course and targeted for pre-readers and early elementary students. Courses 2-4 are generally meant for grades 2 and higher. The teacher professional development for Computer Science Fundamentals consists of a one-day workshop that familiarizes teachers with the course, teacher tools, and the Code.org learning platform. Code.org has prepared facilitators all over the United States to hold workshops for elementary teachers throughout the year.

Elementary School Teachers

In the 2015-2016 school year, Code.org Facilitators prepared approximately 24,000 teachers. The overall number of teachers attending our workshops is a 118% increase from the 2014-2015 school year, when approximately 11,000 teachers attended workshops. Teacher satisfaction with our workshops as measured by a post-workshop survey is incredibly high and continues to increase, even as we scale.

Table: Overall satisfaction with and preparedness from Computer Science Fundamentals workshops[7]

	Computer Science Fundamentals workshops up to 7/15/15	Computer Science Fundamentals workshops up to 8/1/16
Overall, rate your satisfaction about the workshop.	97%	98%

[7] Source: Surveys given to teachers after attending a Computer Science Fundamentals workshop.

I feel prepared to teach the Code.org computer science courses.	89%	89%
This was the absolute best professional development I've ever participated in.	77%	78%*[8]

Note: Shows percent of teachers who had any level of agreement with the statements.

In addition to tracking numbers of teachers trained and satisfaction with our professional learning workshops, we felt it was important to understand not only what happens during professional development, but also what happens with our teachers throughout the school year. Understanding changes in attitudes and self-efficacy, as well as how teachers use our materials, can help us better support teachers throughout the school year.

Outlier Research & Evaluation Findings - Elementary School Teachers

In order to better understand teacher experiences with CS Fundamentals, Outlier Research & Evaluation examined relationships between teachers' experience teaching computer science with CS Fundamentals and their attitudes toward and self-efficacy with regard to computer science teaching. The findings showed no changes in teachers' affinity for teaching CS or CS teaching self-efficacy..

The findings also showed that after their experience with CS Fundamentals, teachers were more likely to think that all elementary schools should require computer science at the end of the year. There was a 0.47** increase, on a 6-point scale, for teachers agreeing with the following statement from pre- to post-survey: "I think all elementary school students should be required to take computer science."

Elementary School Students

In the 2015-2016 school year, the number of students engaging with our Computer Science Fundamentals courses nearly doubled. ***The number of students with Code.org prepared teachers increased over 310%.*** Additionally, diversity of students improved overall, and now mirror the demographics for K12 schools in the United States[9]. This mirroring is likely a result of school-based use of the course, rather than students who chose to do this on their own, since the student demographics in elementary school classrooms should match what we see in the general population.

Table: Number of elementary school students reached by Computer Science Fundamentals (K-5)[10]

	2014-2015 Computer Science Fundamentals	2015-2016 Computer Science Fundamentals
Number of students	2,400,00	4,400,000
Number of K-5 students	1,200,00	2,300,000
Number of K-5 students from	160,000[11]	660,000

[8] Note that anything in the report marked with a * is considered statistically significant at the *p<.05 level, and anything with **is statistically significant at the **p<.01 level, meaning findings have a 5% or 1% likelihood of being attributable to chance.

[9] Source: http://nces.ed.gov/programs/coe/indicator_cge.asp

[10] Source: Diversity data (females and % African American or Hispanic Latino) were calculated from responses to a Code.org teacher survey asking about student demographics. Numbers provided are rounded up or down and therefore approximate.

[11] This number may be inflated, as some of these teachers may have been trained after the 2014-2015 school year.

Code.org-prepared teachers		
% female	42%	43%
% underrepresented minorities	37% Black or Hispanic	46%

In addition to examining student demographics, starting in January of 2016, we identified a method for measuring Basic Coding Proficiency in Computer Science Fundamentals. When we refer to *proficiency*, we mean what a student is able to do at any given moment. We distinguish this from *learning*, which we define as a change in proficiency over time (i.e. if students become more proficient in something, we call this learning). Ideally, we would be measuring student learning; however, we must first understand proficiency before we can measure learning as a result in engaging with our courses.

Defining "Basic Coding Proficiency" in Computer Science Fundamentals

2016 is the first year Code.org is reporting on student Basic Coding Proficiency in Computer Science Fundamentals. This is a metric that is based on student performance on Code Studio puzzles in our Computer Science Fundamentals courses. Every puzzle in Computer Science Fundamentals is tagged with the concepts it addresses, and the difficulty level within that concept (from 1 to 5).

The Concepts are:

- Sequencing (basic algorithms)
- Loops (e.g., repeat, repeat until, repeat while, and for)
- Events
- Variables
- Functions
- Conditionals

To demonstrate Basic Coding Proficiency, a student must prove their skill in at least three different concepts. For each concept, the student must complete three or more puzzles at a difficulty level of 3 or higher without hints and with the optimal number of blocks. You can read more about Basic Coding Proficiency here.

In the 2016 calendar year, approximately 888K students demonstrated Basic Coding Proficiency in Computer Science Fundamentals.

The table below shows the number of students who have demonstrated proficiency broken down by concept area and difficulty level, from January 1, 2016 - December 31, 2016.

Table: Number of students who have achieved proficiency January 1, 2016 - December 31, 2016

	Difficulty 1*	Difficulty 2	Difficulty 3	Difficulty 4	Difficulty 5
Sequencing	3619302	3555217	3286079	1112935	981516
Debugging[12]	685152	662018	525237	135342	N/A
Loops	2343097	1922722	1734299	1387980	353852
Events	582198	546671	473030	106810	N/A

[12] While we track student progress on debugging, we do not count it as part of Basic Coding Proficiency. This is because debugging is difficult to separate from other skills, because the debugging is often done within the context of another skill, such as debugging loops, debugging conditionals, etc.

Variables	341402	249063	146267	13906	7438
Functions	547077	450360	313938	22290	7525
Conditionals	1018804	1013773	771871	431386	93874

* Difficulty 1 is the least difficult and 5 is the most difficult.

Naturally, not all of the students reach proficiency. Students who only take Course 1 cannot reach Basic Coding Proficiency (difficulty level 3), as the puzzles are not difficult enough for them to hit that bar. Additionally, about 30% of students try the courses and do not continue (perhaps during or as follow-up to an Hour of Code activity), and about 40% go on to use the courses for more than 30 days.

Outlier Research & Evaluation Findings - Elementary School Students

Another pathway we have gone down to try to better understand student experiences with our Computer Science Fundamentals course and differences across students more generally was to expand our sources of data by having Outlier Research & Evaluation conduct a study of students participating in the Computer Science Fundamentals course in school. To this end, they did a pre/post comparison of 3rd-5th grade students in three school districts, which produced matching data for 574 of the participants. Only two of the three districts are represented here, as data was never returned from the third district. It is important to note that after the study concluded, the researchers discovered that 182 students out of the pool of 574 student participants were not actually taking part in the course, providing us with a comparison group to determine if there were differences between students who took the course and those who did not.

The intent of the study was to measure changes in students attitudes towards and self-efficacy with computer science. The majority of students who participated in the study are considered to be underrepresented minorities in computer science (only 24% reported that they were White or Asian). The gender breakdown was nearly 50-50. Some of the high level findings from the surveys, as reported by Outlier, are listed here:

Comparing the pre and post surveys show:

- ***For the following categories, there was no difference in outcomes from pre- to post-survey for the students who took part in Computer Science Fundamentals and those who did not:***
    - ***Positive perceptions towards computer science***
    - ***Positive perceptions of their own self-efficacy with computer science.***
    - ***Relevance of computer science to real life.***

Interestingly, the data for both of these groups show very slight declines from pre- to post-, however, they were still on the positive side of the scale for all of the categories listed above. This tells us that over the course of the year student attitudes and self-efficacy might go down, but we cannot attribute that to the Computer Science Fundamentals course given the differences in the two groups of students.

- ***The following questions show statistically significant differences for students who took part in Computer Science Fundamentals and those that did not from pre- to post-survey.***
    - ***For the item,"I think it would be cool to choose a job/career in computer science," the non-Computer Science Fundamentals students decreased by 0.51 points on a 4-point scale, while the Computer Science Fundamentals students decreased by 0.2 ($p = 0.03$). For both groups, the data was still on the positive side of the likert-scale.***
    - ***For the item, "People who do computer science work alone most of the time," non-Computer Science Fundamentals students reported a 0.58 point increase on***

> ***a 4-point scale in their CS misperceptions for the item, while CS Fundamentals students reported an increase of 0.15 points ($p < 0.01$).***
>
> Therefore, for two items, non-Computer Science Fundamentals students experienced slightly more negative changes over the school year compared to Computer Science Fundamentals students. This suggests that taking part in Computer Science Fundamentals has a small positive impact on student attitudes towards computer science as a profession, and a small positive impact on reducing misperceptions about how computer scientists work.

We are pleased to see that positive attitudes towards and self-efficacy with computer science maintain throughout the year. Broadly speaking, we believe these attitudes wouldn't have been possible without the global increase in emphasis on the importance of coding and computer science in education over the past few years, which has been a community effort of hundreds of organizations. When considering the study referenced above, all of the districts who took part in the study had large-scale Hour of Code programs the prior year, which means that many (or even most) of the students in the study may have had some prior experience with computer science. This may have impacted their survey results on the pre-survey since most, if not all, of these students had engaged with computer science prior to participating in this study. This is further supported by the data showing minimal differences between students who engaged with Computer Science Fundamentals and those who did not. However, as mentioned, taking part in Computer Science Fundamentals does appear to have a positive impact on student's attitudes towards and perception of careers in computer science.

While these results are positive, we will continue to explore how we can increase students' self efficacy with and attitudes toward computer science as we build out an updated versions of the K-5 courses this year. We are currently working on a new version of the course which incorporates much of the feedback we have received from students and teachers, including ensuring that we have materials specific to each grade level, K-5.

Elementary School Highlights
We continue to grow our K-5 teacher pool via our Computer Science Fundamentals professional learning program, and overall satisfaction for the workshops has been high and continue to move in a positive direction. Growing our teacher pool has also meant growth in the number of students reached every year. Additionally, as teachers teach CS, we see their opinions change in a positive direction about requiring that all elementary students take a CS course. As for students, we expect almost 1 million students will demonstrate Basic Coding Proficiency on our platform this year, and we also know that the more students have exposure to CS, the more they report knowing what CS is. Finally, our diversity numbers are almost representative of the student population in the United States, showing that we are gaining ground on our mission to bring computer science to females and underrepresented minorities.

Elementary School Next Steps
While we do have a huge number of teacher accounts on Code Studio, it is difficult to track all of the use cases for both our curriculum and Hour of Code activities. Two goals for the 2016-2017 school year are to better track which teachers are using our curriculum, and systematically examine the ways in which we can can encourage both new and existing teachers to stay active until their students reach a target level of coding proficiency. Our mission cannot be realized without teachers, and we want to explore the ways in which we can encourage them to stay involved in teaching computer science to their students.

In general, it is difficult to measure the number of teachers implementing the Computer Science Fundamentals curriculum after attending workshops. We know some teachers teach the course without having their students set up accounts and login. In some schools, a single technology teacher sets up accounts for the whole school, and only that technology teacher is counted as 'teacher teaching the class.' Additionally, teachers often sign up for multiple accounts, using one when attending a workshop and a different one when they begin teaching. We are continuing to explore different use cases of the curriculum to help us determine if we are accurately measuring these numbers, and will be exploring teacher engagement overall as a means to discover optimal pathways towards implementation.

We have a lot of work remaining in studying student proficiency, including the impact of teachers and professional development workshops on proficiency. Also, our proficiency reports only measure what skills students demonstrate, but do not measure actual *learning*.

Middle School Programs

Code.org currently prepares teachers in two courses that are targeted for middle school, Computer Science in Science and Computer Science in Algebra. These are not stand-alone computer science courses, rather they are integrated into middle school algebra or science. For science, Code.org has partnered with Project GUTS to deliver this middle school science program. The goal of the program is to situate computer science practices and concepts within the context of life, physical, and earth sciences. For algebra, Code.org partnered with Bootstrap, who developed a curriculum in which teachers algebraic and geometric concepts through computer programming. Teachers engage with professional learning for these courses at multiple points throughout the year.

Middle School Teachers

For the 2015-2016 school year, Code.org prepared 392 Computer Science in Science teachers and 275 Computer Science in Algebra teachers, for a total of ***667 new middle school teachers. This is a 283% increase from the 2014-2015 school year***, when a total of 174 new middle school teachers attended our workshops (85 Computer Science in Algebra teachers and 89 Computer Science in Science teachers).

Table: Overall satisfaction with and preparedness from middle school professional learning workshops

	2014-2015 cumulative (weighted)	2015-2016 Cumulative (weighted)	2015-2016 Computer Science in Science	2015-2016 Computer Science in Algebra
I would recommend this professional development to others.	99%	90%	90%	89%
This was the absolute best professional development I've ever participated in.	85%	75%	75%	75%
I feel prepared to teach the Code.org computer science courses.	83%	76%	74%	79%

Note: Percent of teachers who had any level of agreement with the statements. Statistical significance was not calculated here.

While our overall satisfaction numbers with our middle school workshops have decreased from last year, satisfaction still remains high***. When taken together, 90% of our teachers would recommend this professional learning workshop to others, showing that teachers value our workshops after attending them***. One reason for the decline may be due to scale - when programs grow larger, it can be difficult to maintain the same level of fidelity to the program, perhaps compromising quality. We will talk with our partners and facilitators in the coming months to see if we can glean any other reasons for this decline.

Middle School Students

In the 2015-2016 school year, Code.org reached approximately 40,500 middle school students through the courses Computer Science in Science and Computer Science in Algebra. In 2014-2015, Code.org served approximately 12,000 middle school students, showing ***a 238% increase in middle school students served in 2015-2016***. It is important to note that the vast majority of this increase was from Computer Science in Science.

Table: Number of middle school students reached by program[13]

	2014-2015 Computer Science in Science	2015-2016 Computer Science in Science	2014-2015 Computer Science in Algebra	2015-2016 Computer Science in Algebra
Number of students	6,000	34,000	6,000	6,500
Percent female	N/A	49%	N/A	50%
% African American or Hispanic/Latino	N/A	51%	N/A	46%

Middle School Highlights

Overall our student base is increasing, and we are seeing more racial diversity in our middle school programs. Given Code.org's focus on underrepresented students in computer science, the increase in racial diversity is a positive trend. Our middle school courses have the highest rates of female participation of all our courses. This is likely due to the fact that these courses are usually integrated into pre-existing math or science courses that students are required to take, and therefore the gender diversity there is similar to what it would be for any other required course.

Middle School Next Steps

It is important that teachers feel prepared to teach following a professional learning workshop, and in comparison to our other programs, these numbers could improve. We carefully monitor teacher satisfaction following workshops by administering surveys after every workshop. We then share this data with facilitators. For future school years, we've prepared school district employees to be facilitators for their districts, and they will continue spreading these programs as part of their district work.

High School Programs

Code.org currently offers two high school courses, Exploring Computer Science and Computer Science Principles. Exploring Computer Science is a nationally recognized introductory computer science course. Our Computer Science Principles course, which has been endorsed by the College Board, is designed to be a rigorous, engaging, and approachable course that explores many of the foundational ideas in computing. This course can be taught as both an AP and non-AP course. As with our middle school courses, teacher preparation for Exploring Computer Science and Computer Science Principles happens at multiple points throughout the year.

High School Teachers

For the 2015-2016 school year, Code.org prepared 113 new Computer Science Principles teachers and 435 new Exploring Computer Science teachers, for a total of ***548 new high school teachers**. **This is a 71%***

[13] Source: Diversity data (females and % African American or Hispanic Latino) was calculated from responses to a teacher survey asking about student demographics.

increase from the 2014-2015 school year, when we prepared 320 new high school teachers (4 Computer Science Principles teachers and 316 Exploring Computer Science teachers).

Table: Overall satisfaction with and preparedness from high school professional learning workshops

	2014-2015 Exploring Computer Science	2015-2016 Exploring Computer Science	2014-2015 Computer Science Principles	2015-2016[14] Computer Science Principles
I would recommend this professional development to others.	96%	85%	N/A	94%
This was the absolute best professional development I've ever participated in.	76%	77%	N/A	86%
I feel prepared to teach the Code.org computer science courses.	84%	83%	N/A	86%

Note: Shows percent of teachers who had any level of agreement with these statements.

Teacher satisfaction for our Exploring Computer Science professional learning workshops has declined. One explanation for this decline is that the 2014-2015 teachers were "early adopters," passionate about spreading computer science in their school whereas 2015-2016 were more likely to be required by their district to attend the workshops. In addition, workshops were shortened this year – a factor that likely contributed to the decline in the numbers. For Computer Science Principles, this was a year for gathering baseline data and we will be monitoring the results of our professional learning workshops carefully to ensure that teachers are receiving high quality experiences in our workshops.

Outlier Research & Evaluation Findings - High School Teachers

As with our elementary program, in addition to examining teacher satisfaction with our professional learning workshops internally, we believe it is important to look beyond our professional development to try to understand teacher experiences throughout the year, so we can support them in the best ways possible. In the interest of that goal, Outlier Research & Evaluation Outlier administered questionnaires, pre- post- with teachers across the United States. The questionnaire focused on teachers' attitudes towards and self efficacy with teaching these courses.

For Exploring Computer Science, 30 teachers took the pre-questionnaire and 51 took the Post-Questionnaire. We matched valid pre- and post- data for 26 teachers, representing 86.67% of the original pre-questionnaire sample. There were few changes in teachers' attitudes towards or self-efficacy with teaching computer science, or in their understanding of the Exploring Computer Science learning objectives from the beginning of the school year to the end of the school year. However, it was found that ***these teachers' positive attitudes towards using a set curriculum to teach introductory computer science increased by 0.52* points on a 6-point scale from pre- to post-questionnaire, suggesting that teachers find that a set introductory computer science curriculum is helpful or supportive to their teaching.***

For Computer Science Principles, 60 teachers responded to the pre-questionnaire and 83 responded to the Post-Questionnaire. Outlier matched 27 teachers with valid pre- and post- data, representing 45% of the original Pre-Questionnaire sample. No changes occurred in teachers' computer science teaching

[14] 2015-2016 is the first year we have tracked this for Computer Science Principles.

self-efficacy over the course of the year. ***While teachers decreased by -0.26* points on a 6-point scale in their perceived understanding of the CS Principles learning objectives from the pre- to post-questionnaire, their attitudes towards using a set curriculum to teach introductory computer science increased by 0.78** points on a 6-point scale***. While these findings appear contradictory, this may be a result of the course increasing in difficulty as the school year progresses, making it so that teachers feel like they do not have the same level of understanding of the content later in the year as they did earlier in the year. In terms of attitudes, the positive increase may be due to the course becoming more fun further along in the year. Either way, the responses are on the positive side of the likert scale, showing that in general teachers do understand the learning objectives and do like teaching with a set curriculum.

As part of this study, Outlier Research & Evaluation conducted interviews with a subsample of Exploring Computer Science and a small group of Computer Science Principles teachers, focused on the following questions: 1) How do teachers implement Code.org's Exploring Computer Science/CS Principles course?; and 2) What are teachers' opinions about the Code.org Exploring Computer Science/CS Principles course?
Keep in mind that these findings are not necessarily generalizable to all Exploring Computer Science or Computer Science Principles teachers or classrooms given the small number of teachers interviewed. Here are some of the main findings from the interviews:

Exploring Computer Science Teacher Interviews (N=12)

- Teachers taught Exploring Computer Science for a range of reasons, from being told by school leaders to teach the course to a desire to provide all students with Computer Science opportunities.
- The majority of teachers had positive attitudes towards Exploring Computer Science, and especially liked the format and sequencing of the curriculum as well as the hands-on approach.
- Teachers had some challenges implementing Exploring Computer Science, including timing/pacing, student needs, access, and school administration and communication.
- Teachers valued the introductory unplugged lessons for creating a level playing field and engaging students in deeper understandings of computer technology.
- Most teachers felt the curriculum as "too young" for high school students, noting that the unplugged activities, Scratch, and programming languages did not appeal to their students.

Computer Science Principles Teacher Interviews (N=30)

- The Computer Science Principles materials are comprehensive and easy to use for lesson planning. The teaching tips and assessments were named as being particularly helpful.
- Nearly all teachers felt that the professional development was high-quality, professional, and useful.
- Most teachers felt that the Computer Science Principles professional development was some of the best they had ever experienced.
- Most teachers would recommend the Computer Science Principles curriculum to other teachers.
- Two-thirds of the teachers did not spend much time using online resources due to time constraints. Those that did sometimes felt they were useful when extra support was needed, while others didn't feel that the time invested in the online professional learning was worth the payoff, in that they didn't learn a lot from the online resources.

High School Students

In the 2015-2016 school year, Code.org reached approximately **21,000**[15] high school students through the courses Exploring Computer Science and Computer Science Principles. In 2014-2015, Code.org served

[15] Source: The number of students was calculated by estimating the average number of students per teacher, then multiplying that by the number of known teachers.

approximately 13,000 high school students, showing a **62% increase in high school students served in 2015-2016**. While this increase is notable, it is primarily because Computer Science Principles was for the first time in the 2015-2016 school year.

Table: Number of high school students reached by program[16]

	2014-2015 Exploring Computer Science	2015-2016 Exploring Computer Science	2014-2015 Computer Science Principles	2015-2016 Computer Science Principles
Number of students	13,000	18,000	N/A	2,700
Percent female	34%	37%	N/A	30%
% African American or Hispanic/Latino	60%	56%	N/A	57%

While this data is helpful for us in tracking how our programs are growing and how they are reaching students, we wanted to understand our students attitudes towards and self-efficacy with computer science change over the course of the year.

Outlier Research & Evaluation Findings - High School Students

Two studies, one with Exploring Computer Science students, and the other with CS Principles students, were conducted by Outlier. The questionnaires measured students' attitudes toward and self-efficacy in computer science, the extent to which they identify as someone who does computer science or as a computer scientist, the relevance of computer science to their future academic and professional endeavors, their interest in pursuing a career in computer science, and their misperceptions around computer science. The students in both studies were from three large urban school districts in the United States, however, only two of those districts are represented here, as data was never returned from the third district. Notably, in both studies, the number of male participants was much higher than the number of female participants. Unlike the elementary and secondary courses, Exploring Computer Science and CS Principles are typically offered as electives, so students self-select into these courses. So, the disproportionate number of males compared to females could be a result of boys self-selecting into these courses more than girls.

For Exploring Computer Science, in Fall 2015, 1,156 9^{th}-12^{th} graders completed the pre-questionnaire, and in Spring 2016, 828 completed the post-questionnaire. Outlier was able to match valid pre and post data for 306 students, representing approximately 26.47% of the original pre-questionnaire sample. This sample included nearly twice as many males as females. Some of the high level findings from the post-survey, as reported by Outlier, are shared here:

- ***There were no changes in student attitudes towards or self-efficacy with computer science from pre to post.***
- ***There was a -0.31** point decrease on a 6-point scale in response to the statement: "Computer science doesn't apply to most aspects of real life." Even with the decrease, students were still on the positive side of the likert scale, showing that even while relevance decreased, overall CS is perceived as more relevant than not.***[17]

[16] Source: Diversity data (females and % African American or Hispanic Latino were calculated from responses to a teacher survey asking about student demographics.

[17] Statements with (-) were reverse coded, such that higher averages for these items represent lower levels of misperception while lower averages represent stronger misperception.

- ***Compared to males, females reported a 0.41* point increase on a 6-point scale in their attitudes toward pursuing a career in computer science. In other words, males decreased in their positive perceptions towards pursuing a degree in computer science by -0.06 on a 6-point scale, whereas females increased in their positive perceptions towards pursuing a degree in computer science by 0.35 on the same 6-point scale.***

For Computer Science Principles, in Fall 2015, 624 9th-12th graders completed the pre-questionnaire, and in Spring 2016, 251 students completed the post-questionnaire. Outlier was able to match valid pre- and post- data for 160 students, representing approximately 25.64% of the original pre-questionnaire sample. This sample was over 80% male. Some of the high level findings from the post-survey, as reported by the researchers, are shared here:

- ***There were no changes in student attitudes towards or self-efficacy with computer science from pre to post.***
- ***There was a -0.27* point decrease on a 6-point scale in response to the statement: "Computer science doesn't apply to most aspects of real life. As with Exploring Computer Science, even with the decrease, students were still on the positive side of the likert scale, showing that even while relevance decreased, overall, CS is perceived as more relevant than not.***
- ***There was no change in attitudes towards enjoying a career in computer science.***

While some of the changes reported here are statistically significant, they may not be educationally or conceptually meaningful. For example, a 5% increase or decrease is only a change of .3 out of 6 points. This is a small shift, and therefore does not provide us with enough information to inform changes. However, these findings can suggest future directions for our work, as discussed below.

High School Highlights
Teachers have very positive attitudes towards both the curriculum and professional learning for all secondary courses. Hands-on activities and real-world relevance of the content for students were mentioned in all of the courses as well as helping maintain student interest and engagement. The in-person professional development experiences were highly valued by teachers.

The number of students enrolled in our high school courses continues to increase, and we hope to see further growth for Computer Science Principles for the 2016-2017 school year. In addition, because these programs are largely implemented in urban school districts with a focus on broadening participation, our racial diversity continues to be better than national averages in high school computer science, and we hope to continue to maintain a high level of diversity in our courses in the coming years.

High School Next Steps
In regard to our teachers, online professional development resources for teachers were not as well utilized or liked as the in-person experiences. In order to scale more broadly and reach teachers who are not able to access in-person workshops, we are exploring different methods of delivering effective virtual professional learning to teachers. Additionally, teachers' likelihood of recommending their workshops to others decreased this year.

One area where we can markedly improve is by understanding teacher preparation from the workshops *after* the teachers are in the classroom implementing what they learned in the professional learning workshops. This will give us an actual measure of how well they were prepared to teach, rather than how well they think they were prepared to teach.

As far as students are concerned, while we are improving our racial diversity numbers, our relative enrollment of females participating in high school computer science classes remains low (although much higher than AP Computer Science A, which is 22% female[18]). It is important that we find ways to increase female participation in computer science, especially in high school. We're hopeful that many of the female students in our elementary and middle school programs will continue to study computer science, and that over time our diversity numbers in the younger grades may carry through to high school.

Additionally, we would like to know more about student learning in these courses in addition to their attitudes towards and self-efficacy with computer science. In the winter and spring of 2017 we will have results from mid-year assessments in Computer Science Principles classrooms. In the summer of 2017, we will begin reporting results of AP Computer Science exams taken by students in our Computer Science Principles classrooms.

Evaluation Plans 2016-2017

For the 2016-2017 school year, Code.org evaluation will become more coordinated and streamlined. Some of our chief goals for research and evaluation for the upcoming year:

- Gather data needed for reporting to funders and continuing fundraising efforts.
- Reduce and streamline survey use across programs to prevent survey fatigue and more carefully capture the data we need and the data we will *use*. We will start limiting some of the data we collect by focusing on gathering information that is actionable in terms of improving our programs and student learning.
- Better understand the factors related to student attitudes towards and self-efficacy with computer science, especially for females and underrepresented minorities in computer science. In particular, we would like to focus on the relevance of computer science to real life across all of our courses. When looking across our courses and programs, this is an area for improvement. Perceived relevance may help increase engagement, so we want to help students understand the applicability of computer science to life. We intend to look across all of our courses and professional learning programs to see if there are ways to increase relevance for students.
- Improve our measurement and understanding of student learning across our courses.

To meet these goals, we have already started the following work:

- Ensuring that our new course, Computer Science Discoveries, has a "purpose" section in each of our lessons, which highlights the relevance of what students are learning to real life.
- Created and deployed a post-professional learning workshop survey that will be used after all professional learning workshops K-12 so that we can more easily track and compare across our own programs.
- Developed a large study around Computer Science Principles that includes tracking the following:
    - Student pre- and post attitudes towards and self-efficacy with computer science.
    - Teacher pre- and post attitudes towards computer science and self-efficacy with computer science teaching.
    - How these factors correlate with one another, and how they correlate with student learning outcome data both on in-course assessments and the AP exam.
- Began Regional Partner Evaluation, which is a comprehensive evaluation plan to both help professionally develop and track the progress of our new Regional Partner program.
- Building a study with the Character Lab to determine if engaging in computer science can help increase grit, which is theorized to be a predictor of success.

[18] Source: Summary of source data for Code.org infographics and stats.