The Landscape of Computing Education in Utah

This report was funded by the National Science Foundation (award #1822011). The purpose of Expanding Computing Education Pathways (ECEP) is to better understand current pathways for computing education in public education at the state level, so that state education agencies, lawmakers, and administrators can make more informed decisions regarding computer science education based on local practices and needs.

The Utah CS4Utah ECEP team is the result of collaboration between public and private education and industry partners. Drs. Helen Hu (Westminster College) and Peter Rich (Brigham Young University) helped create, collect, and analyze the data. They worked with the STEM Action Center and the Utah State Board of Education to collect data from teachers and state agencies. Private industry partners from PluralsightOne (Lindsey Kneuven, Michael Mixon, Sara Epps Donahue and Ann Vance) and Dr. Suzy Cox (Utah Valley University) provided valuable editing and visualizations to better communicate the findings herein. Finally, this report would not have been possible without the participation from nearly 400 educators and administrators who responded to the state-wide survey and those who participated in regional focus groups.

Peter Rich, PhD	Brigham Young University
Helen Hu, PhD	Westminster College
James Christensen	Brigham Young University
Jordan Ellsworth	Brigham Young University[1]

Contributing Editors

The following individuals contributed their expertise to the editing and representation of information in this report. They represent a variety of partners and the Utah ECEP team.

Sara Epps Donahue	Pluralsight
Ann Vance	Pluralsight
Lindsey Kneuven	Pluralsight
Michael Mixon	Pluralsight
Suzy Cox, PhD	Utah Valley University
Tamara Goetz	STEM Action Center
Lynn Purdin	STEM Action Center
Brandon Jacobson	Utah State Board of Education

Focus Group

Catherine Bennett, Lisa Birch, Jay Blain, Jim Boyd, Myra Brown, Angie Card, Jim de St Germain Charles Christopher Elrod, Derek Hansen, Cecily Heiner, Deb Ivie, Shalini Kesar, Robert Kilmer, Jodi Lusty, Kristy Maclachlan, Steve Manning, Rocky Mazorow, Jon McGowan, Ed Mondragon,
Matt Patterson, Lynae Puckett, Nicole Reitz-Larsen, Trent Stokes, Karsten Walker, Christian Waters, Natalie Watts, Richard West, Don Yates.

---

[1] How to cite this report:

Rich, P. J., Hu, H., Christensen, J., & Ellsworth, J. (2019). *The Landscape of Computing Education in Utah*. CS4Utah.org.

Table of contents

1. Introduction

Private industry, government and public institutions are all trying to figure out how best to prepare Utah's growing student population for a computational world. **This report provides a comprehensive look at computer science (CS) education across Utah elementary, middle/jr. high, and high schools.** Computing-related careers have long been a part of Utah's industry, which currently employs the largest group of Utah professionals; yet, the study reveals that we lack in preparing our students for success in these careers in their own state. The intent of this report is to be informative and to provide a comprehensive look at the current state of CS Education in Utah. We attempt to avoid promoting any specific agenda; the opinions expressed herein are those of the authors and do not represent the National Science Foundation, the STEM Action Center, the Utah State Board of Education, or Pluralsight.

2. Key Findings

2.1. Successes

1. There is widespread support across elementary, middle/jr. high, and high school educators for computer-science education, with over 90% of respondents indicating that all students should learn CS before graduating from high school.

2. Students and administrators have been the most enthusiastic recipients of CS education. When asked about the greatest success, teachers repeatedly report high student interest in CS at all levels (elementary, middle, and high).

3. Over 95% of schools that teach CS courses report offering these during school hours, making CS education more available to all students, regardless of demographic.

4. More students in the state are graduating high school having taken at least one computer science course. Only 4% of high school seniors in June 2012 had taken a computer programming course, which increased to 7.7% of high school seniors graduating in June 2015 having taken one or more CS courses during their four years of high school. 20.4% of the seniors from the class of 2018 took at least one CS course.

5. Across all high school CS courses, female student enrollments have increased from 17.6% in 2012-2013 to 33.6% in 2017-2018. Racial/ethnic enrollments across all high school CS classes is actually slightly higher than the demographics for the state's public school student population.

6. A large number of Utah teachers have earned their ECS and CS Level 1 endorsements thanks to SB 93 in 2016 (Computer Science Initiative for Public Schools) and NSF funding for the Utah Exploring Computer Science (ECS) Initiative. When ECS teachers received stipends for taking part in ECS professional development and for teaching ECS, student enrollment in all high school CS courses increased annually an average of 170% a year (ranging in growth from 131% to 220%). Since stipends were shifted to earning endorsements rather than teaching the course, growth in these courses has slowed considerably.

2.2. Challenges

1. Educators who participated in the survey and focus groups recommend making CS a requirement at all levels (elementary, middle, and high school).

2. While enrollments by female students and under-represented minorities have increased in high school classes, this increase appears to be primarily in enrollments in new "introductory" CS courses that fulfill a .5-credit digital studies requirement. These populations remain under-represented in traditional and more advanced CS courses.

3. Teachers need to be trained how to teach CS during their preservice programs. With only a single graduate in CS Education in all of Utah's higher education institutions in 2017, there is a lack of teachers prepared to meet the impending increased demand of CS education.

4. There are even fewer CS graduates of higher education than other reports lead us to believe. As such, rather than 2 available jobs for every CS graduate, there are actually about 3 jobs currently advertised for each CS graduate in Utah.

5. Utah's new .5-credit digital studies requirement has resulted in half-year versions of courses that were designed to be full-year courses. The resulting courses may be insufficiently teaching computer science fundamentals to students, depriving them of more creative opportunities to adequately develop their computer science skill-set.

6. While there are 19 courses that count toward CS credit in secondary education, since 2014-2015, five courses account for over 90% of the Utah high school students enrolled in computer science courses, with Exploring Computer Science (ECS) enrolling more than 50% of Utah students taking CS courses by its third year in Utah.

7. Despite steady growth in CS enrollments over multiple years, CS enrollments declined in 2017-2018. This is concerning given that we expected to see an increase in CS enrollments given nationwide excitement for the new College Board AP CS Principles course and an influx of new CS endorsed teachers from SB 93. If this decline continues, it would be useful to speak with teachers and administrators from regions experiencing the decline to identify reasons for the decline.

8. Teachers at all levels report they need more time, training, and resources to teach CS.

3. Methods

In December 2017, the Utah Expanding Computing Education Pathways (ECEP) team sent out a survey to all public schools in the state of Utah. The survey indicated that a single representative from each school complete the survey for that school. To ensure a common understanding, the survey defined "computing education" as:

> *the study of computer science or related activities. Includes the act of scripting, coding, web development, or computer programming. Does NOT include non-coding uses of computer technology to solve problems (e.g., multimedia development, desktop publishing).*

The survey presented targeted questions depending on whether it was completed for an elementary, middle/jr. high, or high school. It was possible for one person to complete the survey for different schools. Likewise, it was possible for one school to represent multiple educational levels. In this case, the respondent was presented with each relevant section (elementary, middle, or high school) during the initial survey. The survey contained both closed and open-ended questions and took roughly 20 minutes to complete if a school indicated that it did teach computer science (CS). Schools indicating they did not teach CS were presented with fewer questions, though the survey still served a purpose by gathering information about their perceptions and plans related to CS education.

The survey continued to accept new responses through the end of March 2018. The primary method of analysis involved the use of descriptive statistics. These are manifest in counts, percentages, and averages. Qualitative data were analyzed using an emergent open-coding scheme (Glaser & Strauss, 1967). Researchers first analyzed teacher's responses to the different questions and allowed categories of codes to emerge across the first 50 responses for each question. This resulted in agreed-upon codes, defined in a shared codebook. Researchers then independently coded the remaining responses with this code set.

3.1. Limitations

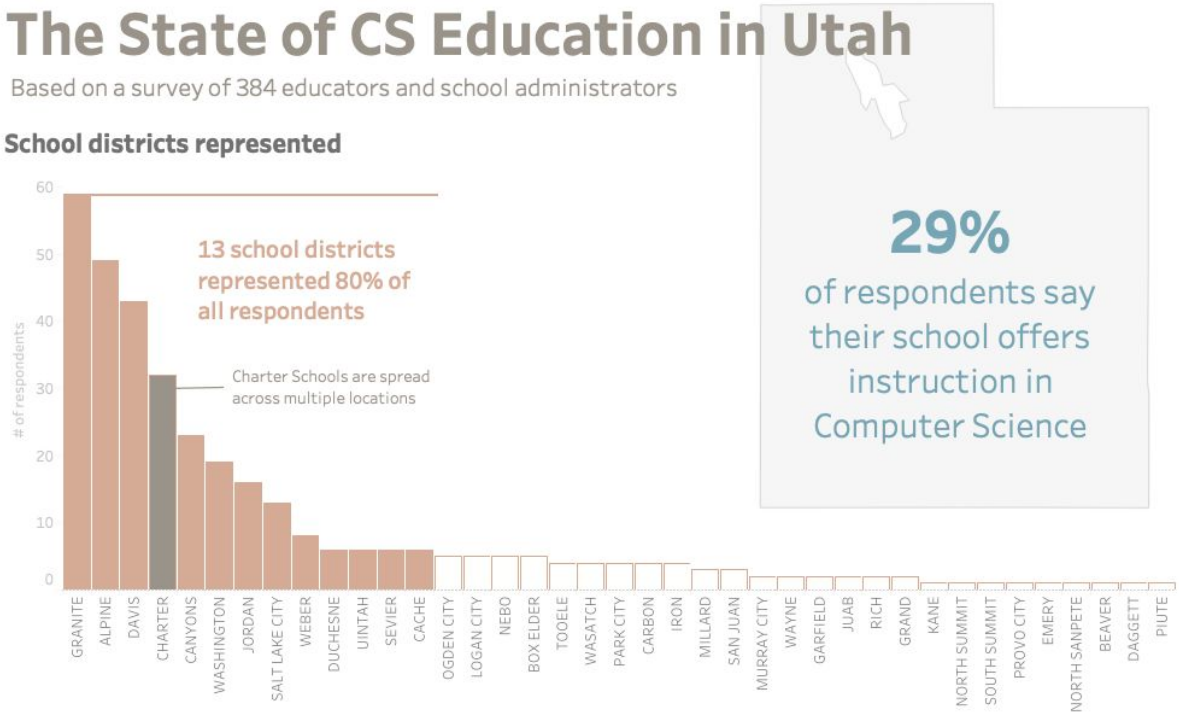
There are three major limitations to the data collected in this survey. First, while nearly a third of all Utah public schools responded to the survey, this may or may not represent the other two-thirds. Second, due to the length of the survey, very few questions were required to be answered. Consequently, many respondents chose to skip answering some questions. Thus, throughout this report, the total answers to a question often add up to be less than the total number of respondents for that category (elementary, middle, or high school). Finally, several schools answered the survey multiple times. In many cases, this occurred when the same individual answered the survey two or three times (often a month or more apart). In other cases, multiple teachers or a teacher and an administrator answered for their school. In all cases, researchers eliminated duplicates by choosing the most complete answers. Where answers were equally complete, researchers prioritized these by the respondent's role, first choosing the responses of Career and Technical Education (CTE) directors over teachers, and teachers over administrators. In the end, each school is only represented once in the quantitative data at each level (elementary, middle, and high). Notwithstanding these limitations, this report represents the most comprehensive description of CS education in Utah to date.

3.2. Results

67% of all respondents are classroom teachers, while school adminstrators provided **29%** of the responses.

Overall, 468 individuals provided usable responses. Some of these represented multiple schools or schools with multiple educational levels. Occasionally, there were multiple responses to a single school. Researchers eliminated duplicates and accepted a single response per school. This resulted in 395 schools: 100 high schools, 111 middle/jr. high schools, and 184 elementary schools. A school was counted at each level. Thus, a school that teaches both elementary and middle school would be counted twice (overall there were 353 truly unique schools).

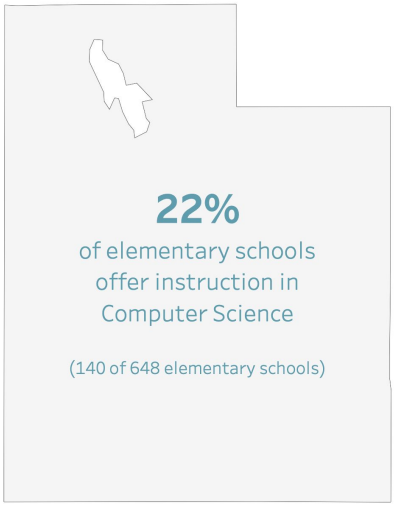
39 of 42 Local Education Agencies (LEAs) responded to the survey (grouping all charter schools together; see Appendix 11.2 for a full list of respondents by school district). Including the LEAs with no responses, this represents an average response rate of 32% per district, with a range of 0-67% response rate across the districts. These responses demonstrate educators' opinions on CS education across the diversity of Utah's urban and rural landscapes.



The following three sections detail trends in CS education across Utah at the elementary, middle/jr. high, and high school levels. Data for elementary and middle/jr. high school primarily derive from the statewide survey and focus groups. High school data is much more detailed due to the ability to track CS-specific courses that reveal enrollment trends, as well as students taking Advanced Placement (AP) exams.

4. Computer Science in Elementary Education

Computer science is seen as extremely important by teachers who responded to the statewide CS survey for their students to receive formal education in computing and that it should be taught in grades k-12. In fact, all stakeholders, view CS education to be a very positive experience. Similarly, the students in the elementary CS class are also the most excited to learn CS. The results of the statewide survey discussed below indicate that the state still needs to provide increased CS education and consistent exposure.

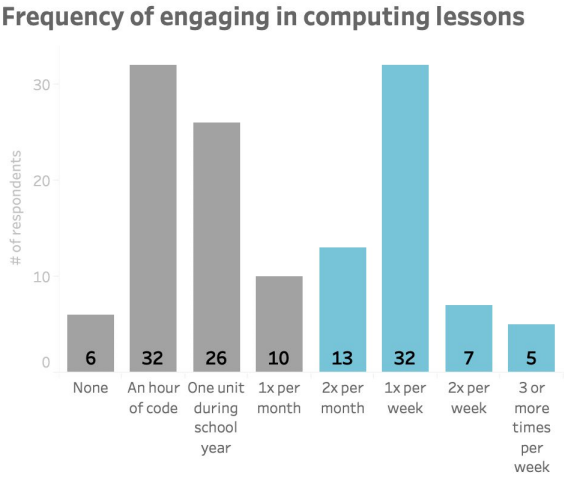


4.1. Who is being taught?

The statewide survey on CS education indicated that only 10% of the unique elementary schools that responded currently are required to teach CS in elementary schools demonstrating that those currently teaching CS are largely doing so either with the encouragement of their administration (60.4%) or simply of their own accord (29.6%). However, 21.6% of the schools said that they Offer Computing as a portion of the total number of public elementary schools across the state.

44% of Elementary School students who engage in computing lessons, do so **2 or more times per month.**

The data indicates two basic patterns for students to participate in CS courses in elementary school. 43.5% of the elementary school students receive CS classes more than twice a month with more than half of that group (56%) participating once a week. In fact, 50 Utah elementary schools reported that students are participating in CS activities at least once a week. The remaining 56.5% of CS elementary students only participate in CS lessons once a month or less with just over half of that group (51.4%) participating in an hour of code or less.



While there is a positive trend toward introducing young children to CS, such infrequent exposure is unlikely to produce the skills and dispositions needed to encourage greater participation in CS in middle school and beyond.

In addition to the varying frequency of participation in CS lessons, lessons also vary in length. 42.7% of such lessons fall in the traditional class length of 31-45 minutes and more than a third of the lessons are 30 minutes

or less (36.%). Breaking this down by how often CS is taught reveals that students participating in more frequent lessons (1x/week or more often) are also participating in longer lessons (>30 min) at a greater rate (1.47x more often). In terms of exposure, there is a clear advantage for elementary students participating in CS education at least once a week, as they also participate in longer lessons.

4.2. What is being taught?

Class options on the type of computing language to teach are growing as the emphasis on CS education increases worldwide. Code.org is the most commonly used curriculum (see Table 3.2.1) and Scratch appears to be (https://scratch.mit.edu) the most commonly-taught language. Lego, Tynker, ScratchJr, Kodable, Wonder Workshop (i.e., Dash and Dot) are also used at dozens of schools (see glossary for descriptions of each of these products). This follows international trends, where Scratch, Code.org, and Lego appear to be the most used curricula or languages in elementary CS education (Rich et al., 2018).

Most elementary school CS curricula are block-based. A block-based language does not require a student to know the specific lexical and syntactical features of a language but rather represents coding methods and components through a drag-and-drop-based interface. The pieces typically "snap" together like a magnetic puzzle. This enables students to learn to code even before mastering typing. Some block-based languages meant for pre-school-aged learners (e.g., ScratchJr, Kodable, Code-a-Pillar) do not even require that students know how to read.

Table 3.2.1

What curriculum is being used to teach computing at your school?	
Code.org	112
Teacher-Developed Lessons	56
ScratchEd	42
Tynker	33
Project Lead the Way	13
Bootup	10
Lego	10
CS Discoveries	7

Table 3.2.2

What computing languages/technologies are used to teach computing at your school? *	
Scratch	63
Lego Mindstorms	43
ScratchJr	36
Dash & Dot robots	26
Kodable	23
Javascript	20
Code.org	18
HTML, CSS	15
Raspberry Pi	9
Micro:bits	8
Beebots	8
Arduino	8
Sphero	7
Code-a-Pillar	6
* with at least 5 responses. Other responses included: micro:bits, Ozobot, Vex Robotics, Swift Playgrounds, The Foos, Snap!, RunMarco, python, Osmo, and Makey Makey.	

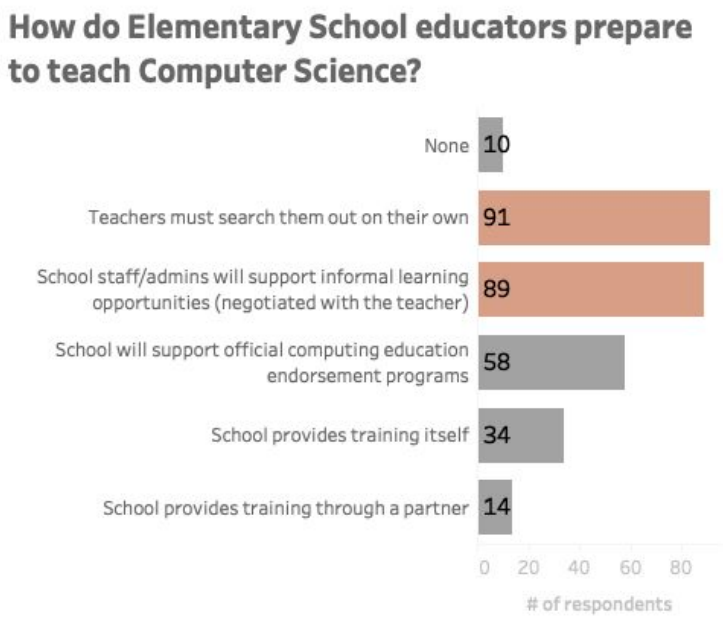
The high level of respondents indicating Teacher-Developed lessons to teach CS education demonstrates that many elementary CS teachers are not reliant on a specific vendor, but rather customize their instruction per their students' unique needs.

4.3. How are elementary educators preparing to teach CS?

There are many sources that teachers might look for CS training. The most common way for teachers to participate in professional development opportunities is by searching them out on their own and negotiating with their administrators. Thirty-four schools reported providing their own training, and only 14 schools reported providing CS professional development training through a partner. Furthermore, the majority of educators (52.4%) are unaware of any plans to increase computing education.

The most common way for Elementary School teachers to participate in professional development opportunities is by searching them out on their own and negotiating with their administrators

This may be a result of Utah not officially-recognizing CS-specific endorsements at the state level for teaching elementary CS. Furthermore, the Utah State Board of Education (USBE) licensure requirements do not currently mandate that preservice teachers learn any CS. Thus, teachers must either work with their districts or seek their own training to prepare to teach elementary CS. If Utah is to seriously consider promoting elementary CS education, it must do more to support teachers' professional development.



Of the schools that do invest in training their teachers, the majority appear to train teachers by having them complete the Code.org training, specifically Computer Science Fundamentals (CSF). This may be due in part to Utah's partnership with Code.org wherein teachers could learn to teach CS by participating in professional development workshops for CSF training. To date, 589 elementary teachers have completed the CSF workshops. However, since state funding has expired for promoting participation in this professional development, teacher participation in CSF training has likewise waned.

Table 3.3.1. *PD that teachers have participated in to learn how to teach elementary CS* *

PD Providers	# schools	Mean # Teachers	Median	Mode	Max	Min
Code.org	77	5.75	2	1	92	1
Project Lead the Way	19	13.95	3	2	100	1
Bootstrap	9	8.56	2	1	29	1
District-Provided	6	23	23	N/A**	31	15

* at least 5 schools represented. Other CS elementary education providers included TEALS, Lego, Bootup, and University courses
** no value occurs more than once

The survey requested the number of teachers at each school that participated in these official CS professional development workshops. The mean and median number of teachers attending varied greatly, indicating that a few schools that sent many teachers were perhaps improperly skewing the averages. Thus, the median number of participants at each school may provide the best picture of how many teachers at each school are receiving official CS professional development. At most schools, it appears there are only 2-3 teachers participating in CS professional development. However, there is a striking difference in district-provided training. In these cases, the median and average were both 23, with the minimum number of teachers

being 15 at one school. This suggests that district-provided CS professional development may provide a different model where all teachers are being trained, whereas most vendor-specific training may be focusing on training specialists or full-time teachers whose primary responsibility is to teach CS.

4.4. What do Elementary Educators say About Teaching CS?

While the aforementioned data show general trends in teaching CS at the elementary level, they do not reveal what the experience is like. Through a series of open-ended questions and through face-to-face meetings, educators shared their successes, voiced their concerns, and proffered advice about teaching CS education in elementary schools.

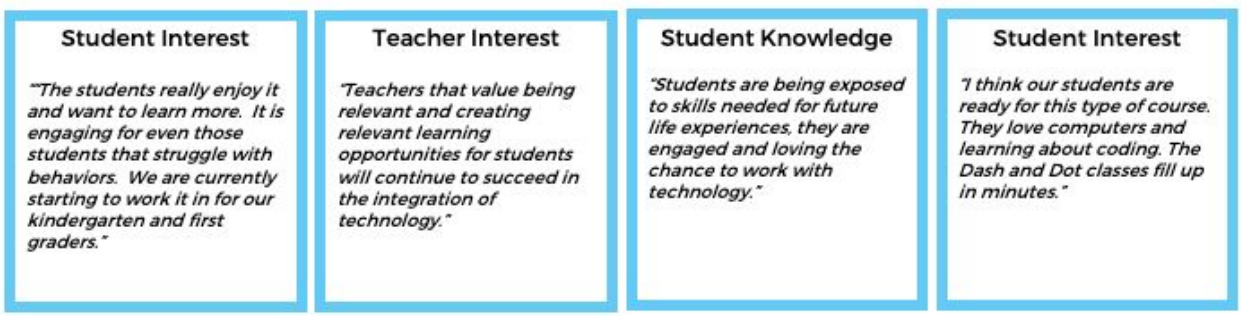
4.4.1. What SUCCESSES have elementary teachers had to teach CS?

Far and away the greatest success educators report with teaching CS to elementary students is how interested elementary students are in CS. While 51.8% of schools openly reported this as the key success, the next closest success was student knowledge, reported by 11.5% of respondents. Considering these were open-ended responses that could have focused on any reply whatsoever, it is telling that more than half of all comments emphasized students' enjoyment for CS education.

Table 3.4.1. *Top successes elementary educators report with teaching CS**

Success	Definition	% schools
Student interest	The degree to which students find enjoyment in computing	51.8%
Student knowledge	comments referring to students learning to code or excelling in their coding	11.5%
Teacher interest	teacher buy-in degree of enjoyment in teaching coding	9.0%

*Reported by at least 5% of schools.



Comments about student interest in computer science focus primarily on students' excitement. This excitement is not limited to one particular student demographic. Educators report students "loving" CS as early as kindergarten. There were specific comments about gifted students, struggling students, and female students.

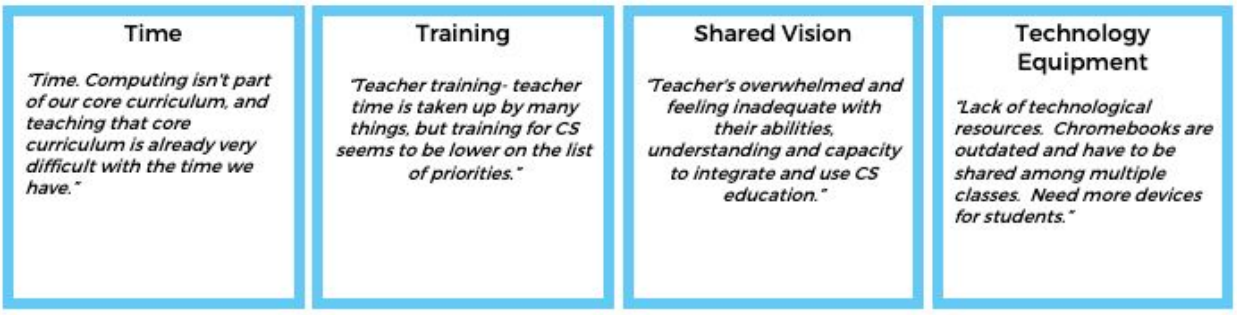
4.4.2. What CHALLENGES have elementary teachers had to teach CS?

While successes dealt primarily with students, challenges in teaching elementary CS education focused on administrative issues: time, funding, integrating CS with core subjects, and creating a shared vision.

Table 3.4.2. *Top challenges faced elementary CS educators**

Challenge	Definition	%
Time	Insufficiency or abundance of time necessary to teach the subject	44.75%
Training	The need for additional professional development and teacher education and/or experience	31.49%
Funding	Resources such as money to pay for coding related costs	18.78%
Tech & Equipment	Resources such as computers, robots, or technology for the coding	11.60%
Implementation	Preparing to, implementing, or integrating the coding in a schedule, program, or system	9.94%
Shared vision	The lack/presence of a shared valuation for coding in the classroom	8.29%
Core	Comments about the core curriculum or requiring coding in the curriculum(e.g., adding coding to the core)	7.18%

* Reported at least 5% of the time



Comments about time mostly focus on how little time teachers have to teach CS. In some cases, they ask that some of the time dedicated to teaching keyboarding be shifted to focus on coding. In other cases, educators indicate that they see students so infrequently, that the time is insufficient to develop an adequate understanding of CS.

Comments about teachers' own training were reported the second most frequently. Elementary teachers indicated that their own CS training was lacking or non-existent. Many

suggested that there need to be greater offerings for participating in professional development to help them acquire such skills.

The third most often cited challenge dealt with how to fund elementary CS education. At the elementary level, funding concerns focused principally on professional development and up-to-date technology.

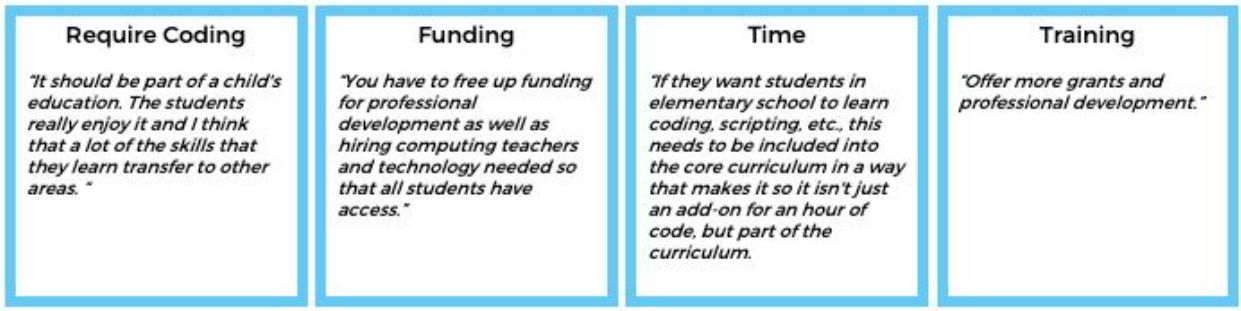
4.4.3. What ADVICE do elementary teachers give about teaching CS?

Toward the end of the survey, teachers were asked, "What advice do you have for policy-makers regarding CS education in Utah?" Answers to this question allowed teachers to provide guidance, either positive or negative, about CS in elementary education.

Table 3.4.3. *Top advice from educators to policy-makers about CS education**

Advice	Definition	%
Funding	Resources such as money to pay for coding related costs	37.76%
Training	Need for additional professional development and teacher education and/or experience	19.58%
Require CS	The lack/presence of a shared valuation for coding in the classroom and making it required	11.89%
Time	Insufficiency or abundance of time necessary to teach the subject	11.19%
Tech & Equipment	Resources such as computers, robots, or technology for the coding	6.99%

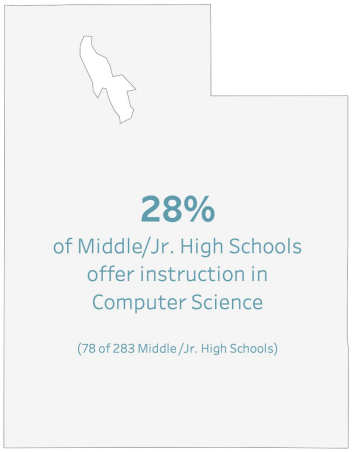
* Reported at least 5% of the time



The key advice elementary educators offer matches closely with their comments about key challenges. They ask for funding to pay for technology and training. They recommend that training be offered, both to those already teaching as well as to those who are training preservice teachers at the university level. They ask that time be dedicated to being able to teach CS. To accomplish this, over a tenth of teachers suggest that CS needs to be a required subject at the elementary level—that unless it were made mandatory, "only a handful will teach it." Finally, teachers recommend creating a shared vision for CS education and propose that this will help to ensure that CS is taught and that these other pieces of advice are prioritized.

5. Computer Science in Middle/Junior High School

Computer science in middle/junior high school is seen as extremely important by 49% of the teachers who responded to the statewide CS survey for their students to receive formal education in computing and that they do so after elementary school but before higher education. Results from the survey discussed below indicate that the schools, while still making progress towards proficiency, understand their role and the need for their students to be proficient in computer science before entering college.



5.1. Who is being taught?

The statewide survey on CS education indicated that over 70% of the unique middle/junior high schools that responded to the survey currently teach CS in middle school, with a median of 2 teachers teaching CS at each school. This accounts for at least 28% of Utah middle/jr. high schools across the state that offer computer science.

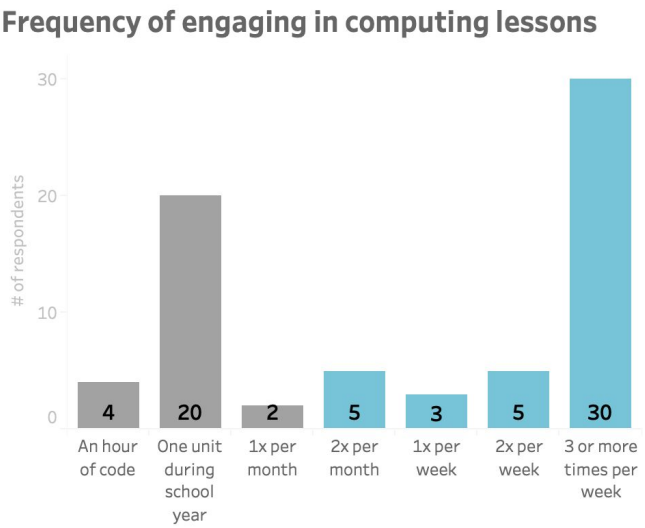
Fewer than half (45.1%) of educators currently teaching CS reported that they are required to teach CS courses at this level, though over a third (36.1%) indicated that they are encouraged to teach it. This is also validated by educators reporting that there is increased top-down support for CS education from administration having high enthusiasm levels cascading down to parents and teachers. However, as at other levels of schools, students appear to be the most enthusiastic about CS education.

Grades taught	Number of schools	% of schools teaching CS
6th	15	19.23%
7th	48	61.54%
8th	49	62.82%
9th	42	53.85%

As in Elementary Schools, there appears to be two basic patterns for students to participate in CS courses in middle/junior high school as indicated by the shape of the graph; either they attend class several times a week as they would a core course (usually in a 1-semester block), or

62% of Middle School students who engage in computing lessons, do so **2 or more times per month.**

they complete CS as a single unit in a course that includes other key topics.
Two thirds of respondents reported that classes last 45+ minutes, suggesting that CS courses are most often taught on a block schedule. Nearly all schools that teach CS reported doing so during school hours (98.5%), with almost a quarter (23.19%) offering additional after-school courses. This suggests that CS courses are available to most students at these schools. The 23% who offer after-school CS likely represent CS or STEM-focused clubs that are offered to students with deeper interests.



On average, there are 30 students enrolled in each CS course, with one school reporting a class enrollment of 75 students. Despite an enrollment far above the mean number of students for core courses (see Utah Administrative Code R277-463-4), there are no restrictions on non-core courses. Classes like this adversely affect both students and teachers; as a teacher with a large class size may reduce the depth and rigor of the material to ensure that s/he is able to attend to all students in the course. In an applied course, such as computer science, this may mean students do not get the needed attention to deal with coding difficulties.

5.2. What is being taught?

CS content is a fairly recent offering within Middle/Jr. High Schools with the median number of years of course offerings being three. Breaking down the course offerings by course provides insight into what is being taught to whom (see Table 5.2.1). For example, Exploring Computer Science (ECS) is a high-school course that officially fulfills the requirements for the newly required .5 Digital Studies credit for 9-12th graders. While nearly all schools that report that they offer ECS are high schools, it is clear that the course is also being offered to younger students at junior high schools. Creative Coding, on the other hand, is a newly-formed 8th-grade course. It would appear it is being offered mostly at middle/jr. high schools. More advanced courses, like web development, mobile development, and game development appear to only be offered at junior high schools.

Some of these courses allow teachers to choose among different curricula. To better understand which approach the different schools used, teachers also reported the specific curricula chosen. Table 5.2.2 demonstrates that though Code.org is not in itself considered a course, it is being used in many courses. Also, similar to elementary teachers, it appears that many middle/junior high school CS teachers create their own instructional materials.

Table 5.2.1. *CS Courses offered in Utah middle/junior high schools by grade level*

Course	# Schools	6th	7th	8th	9th
Exploring Computer Science	38	2	22	22	35
Creative Coding	32	8	24	28	15
Web Development I	12	0	8	7	11
Computer Programming 1	8	0	4	5	8
CS Principles	7	2	7	7	5
Digital Literacy	6	3	4	6	3
Game Development Fundamentals	5	1	3	2	4
Exploring Technology	5	3	5	4	2
HTML 5	4	0	4	4	2
College & Career Awareness	4	2	4	2	1
Robotics	4	1	4	4	0
Computer Programming 2	3	0	1	1	3
Code.org	2	1	1	1	1
Mobile Development Fundamentals	1	0	0	0	1
Web Development II	1	0	0	0	1
CS-Discoveries	1	0	1	1	0

Table 5.2.2. *CS curricula used to teach CS in Utah middle/junior high schools*

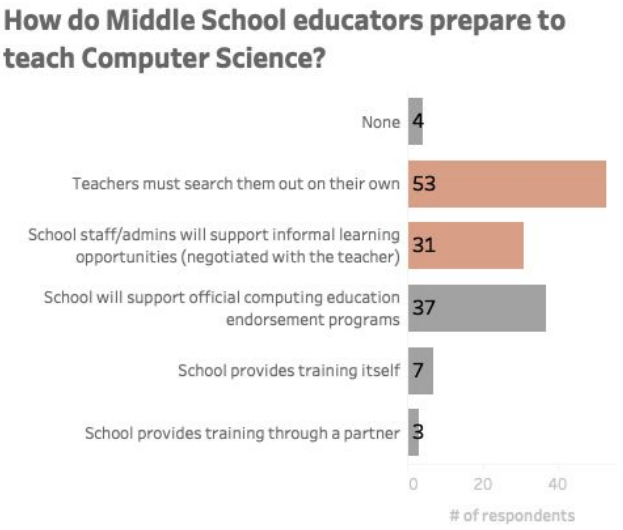
Curriculum	# Schools	6th	7th	8th	9th
Code.org	55	13	42	41	32
Exploring Computer Science	31	1	16	16	29
Teacher-Developed Lessons	23	5	18	19	12
ScratchEd	20	5	18	17	12
CS Discoveries	16	7	14	15	8
Project Lead the Way	8	1	5	5	6
Tynker	6	4	5	5	2
CS Principles	4	1	3	3	4
Touch Develop	3	1	3	3	0
Project GUTS	2	0	2	2	1
Creative Coding	2	0	1	2	0
College & Career Awareness	1	0	1	1	0

.

5.3. How are teachers preparing to teach CS in middle/junior high school?

Middle/junior high school CS teachers have prepared to teach CS in a variety of ways. Professional development occurs primarily through conference attendance and multi-day trainings.

The most common way for Middle School teachers to participate in professional development opportunities is by searching them out on their own and negotiating with their administrators



It is perhaps revealing that the least-reported training middle/junior high school teachers have received is through their preservice preparation programs. This suggests that those teaching CS at the middle/jr. high school level were not formally trained in CS-related content. Most likely, middle school CS teachers were initially hired to teach other subject(s) and, in the past three years, have shifted to teaching more CS-related content as expectations and interests have changed.

5.4. What do teachers say about teaching CS in middle/junior high school?

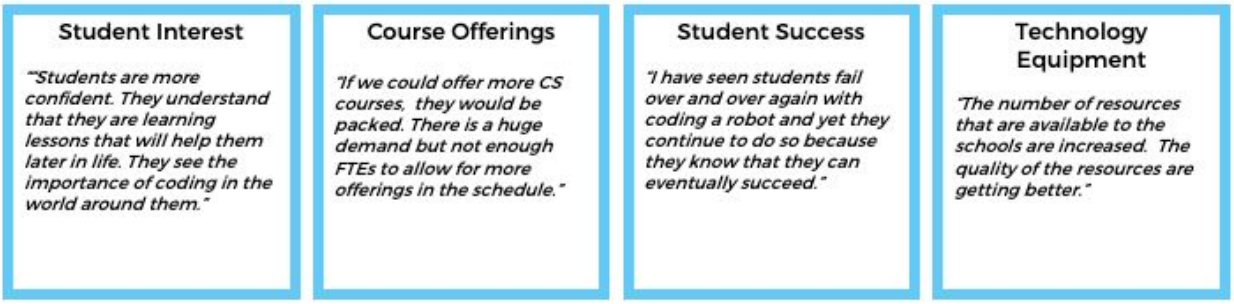
Educators offered their feedback on the experience of teaching CS in middle/junior high school. The following comments were provided by teachers in response to open-ended questions about their challenges and successes in teaching CS in Utah middle/junior high schools, and their advice for policy makers regarding CS education in Utah.

5.4.1. What SUCCESSES have CS middle/junior high school teachers had with CS?

Middle/junior high school educators indicate that they see student interest as the key factor to success with teaching CS as more than three out of every five responses focused on students' excitement or enjoyment for participating in CS-related activities. The next closest success, reported by an 8th of educators, is the new CS program offerings.

Table 5.4.1. *Top successes reported by Utah middle/junior high school CS educators*

Success	Definition	%
Student interest	The degree to which students find enjoyment in computing.	58.62%
Course offerings	The types of programs offered for coding such as afterschool programs, Lego League, etc.	12.64%
Technology & Equipment	Resources such as computers, robots, or technology for the coding.	5.75%
Student success	Students winning awards, competitions, recognition, or being reported as successful in regards to their coding activities.	5.75%
*Reported by at least 5% of educators		



Teachers often presented student interest in CS courses in terms of students' excitement for the topic. Educators repeatedly stated that course enrollments exceeded expectations and capacity. Comments focused on students' increased confidence and ability to deal with difficult situations. They mention students' initiative to learn new coding languages in order to solve problems. But educators' comments about student interest went beyond excitement and focused on real-world issues and equity. For example, several educators mentioned increased interest in CS by girls, minorities, and students who struggle in other subjects, discovering previously "hidden talents" for CS.

5.4.2. What CHALLENGES have CS middle/junior high school teachers had with CS?

Many of the challenges reported by middle/junior high school educators focus on administrative issues. Time, training, and technology & resources were all mentioned by over 20% of respondents.

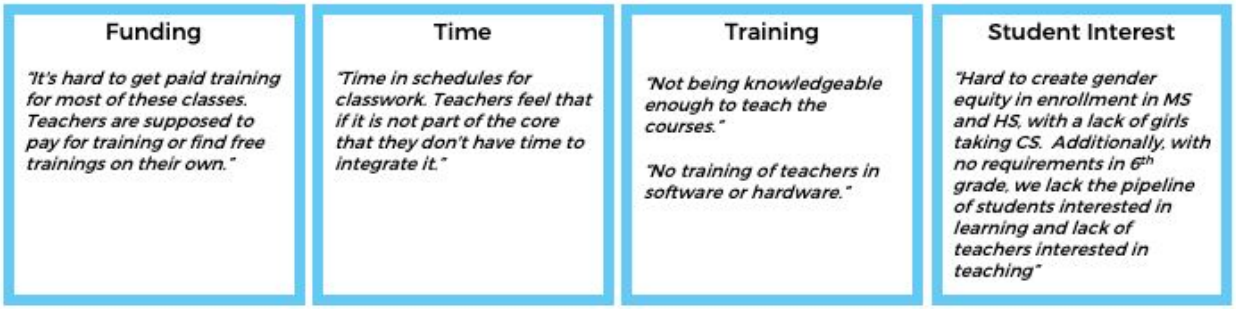
Comments about time were varied. One challenging aspect of time was trying to fit CS into students schedules at schools that offer little room for electives. Other challenges dealt with getting time in the computer lab, which is shared with other classes. Still other concerns dealt with teachers finding the time to upgrade their skills to learn new CS content. This tied in closely with middle/junior and high school teachers' next most common concern of training. Educators repeatedly emphasized their lack of training, the difficulty in participating in training, and "keeping ahead of the learning curve."

While 12.5% of teachers indicated success in obtaining equipment and resources, 20% indicated this was actually a challenge. Mainly, these comments focused on aging equipment or lack of access to the needed hardware and software.

Table 5.4.2. *Top challenges faced by CS middle/junior high school educators in Utah* *

Challenge	Definition	%
Time	Insufficiency or abundance of time necessary to teach subject	24.21%
Training	Need for additional professional development and teacher Education and/or experience	23.16%
Resources & Equipment	Resources such as computers, robots, or technology for the coding	20.00%
Student Interest	The degree to which students find enjoyment in computing	14.74%
Funding	Resources such as money to pay for coding-related costs	12.63%

* Reported by at least 5% of respondents



Student interest was also mentioned as a challenge. Garnering interest in CS repeatedly appeared as a challenge. In other cases, educators felt that interest waned as the courses increased in difficulty. Educators reported problems with student boredom because they are quickly moving beyond current CS teachers' capabilities and are not challenged enough. And still others indicated that the increased amount of student interest itself is a challenge because the school is unable to offer enough seats to those interested in taking CS courses.

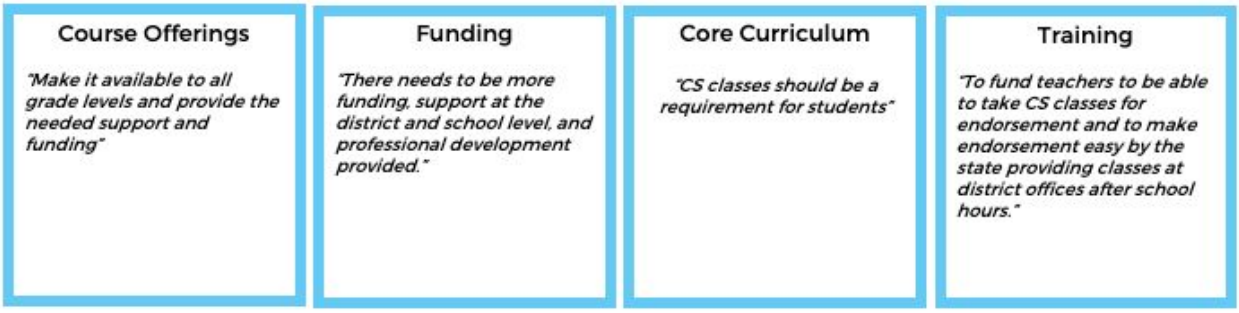
5.4.3. What ADVICE do middle/junior high school educators have for policy-makers about CS?

Advice for policy-makers about CS in middle/junior high school can be summed up in one word—more. Educators clearly emphasized the need for more funding and training. Other top advice emphasized the need for more course offerings and more teacher certification (e.g., licensure, endorsements). A tenth of educators actually recommended that policy-makers *require* more CS education of students if they really want to see it succeed. This same sentiment was echoed by educators in the face-to-face focus groups.

Table 5.4.3. *Top 5 categories of advice CS middle/junior high school educators provided policy-makers*

Advice	Definition	%
Funding	Resources such as money to pay for coding-related costs	39.53%
Training	need for additional professional development and Teacher education and/or experience	19.77%
Course offerings	the types of programs offered for coding such as Afterschool programs, Lego League, etc.	15.12%
Certifications	Comments regarding official endorsements, certifications, or licensure for CS teachers.	11.63%
Make CS part of the core	Comments about the core curriculum or requiring coding in curriculum(ex: adding coding to the core)	10.47%

* Top 5 responses reported by at least 5% of educators



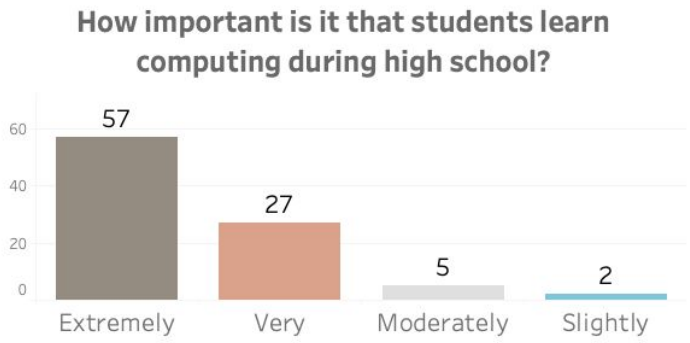
Funding concerns focused primarily on teacher pay and financing for professional development. Some teachers also emphasized funding for hardware or software so that every student might have unfettered access to materials needed to learn CS. Concerns about training were often related to providing CS training so that teachers might earn the related endorsements, thereby enabling more educators to teach CS courses. Due to pay difference between professional CS careers and teaching, some educators suggested that the best path for CS middle/junior high school was to better train those who are already teachers rather than to look to current CS professionals. There were suggestions that training ought to take place during school hours so as to increase the likelihood that teachers would participate.

6. Computer Science in Utah High Schools

Information for this section comes from the statewide CS survey, focus groups, course enrollment data from the Utah State Board of Education, and Advanced Placement (AP) examination trends over the past five years. For this reason, information on Utah CS courses and enrollment in high schools is more comprehensive than what was available at the lower levels.

6.1. Who is being taught?

92.3% of high school educators indicated that it is important for every student to learn CS during high school. This rate is higher than teachers at any other level reported in this survey, demonstrating widespread support for CS education for all Utah high-schoolers. Eighty-five of the 100 high schools that responded to the survey indicated that they currently offer CS courses, a surprisingly high percentage given that as recently as 2012, only 1% of Utah high school students were taking CS courses each year.



100% of the courses at these schools are offered during school hours, meaning that high school CS courses are potentially available to all students enrolled in a CS-offering school, not accounting for enrollment limits. On average, each of these schools offers four CS courses with a median of two sections per course. 84% of courses are offered 2-3x/week, usually lasting more than 45 minutes each. The median number of years schools have offered CS courses is four, suggesting that the teaching of CS has only recently taken place at the majority of these schools. On average, there are 25 students in each CS course.

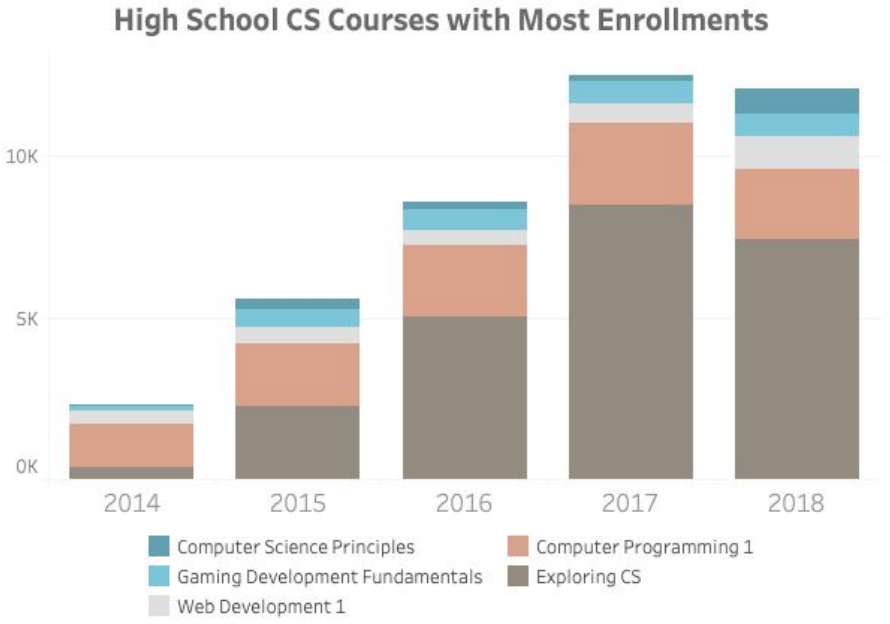
As with elementary and middle schools, all key stakeholders have responded positively to CS education, with educators reporting students and administrators to be the most enthusiastic.

6.2. What is being taught?

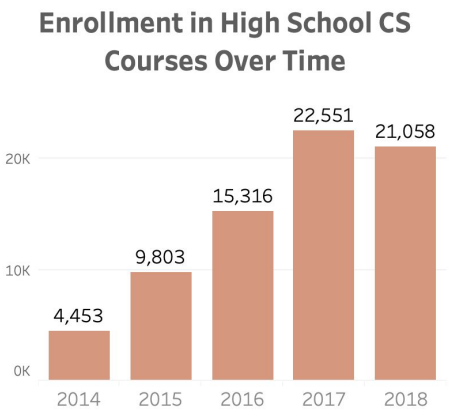
The Utah State Board of Education classifies 18 courses as high school computer science courses (see Section 11.3), but the vast majority of high school students are taking only one of the five most popular CS courses. Ordered starting with the most popular in 2017-2018, these top five courses are: Exploring Computer Science 1, Computer Programming 1, Web Development 1, Computer Science Principles, and Gaming Development Fundamentals 1. Please note that for the 2019-2020 school year, these course names have been updated and can be found in the Appendix (section 11.4.). The graph below shows the enrollment of each of these courses over time. A complete list of high school computer science courses, ordered

from most to least enrollment, can be found in the Appendix (section 11.5.), but many of the remaining 13 computer science classes have been introduced in the past ten years and are taught in few schools.

These five courses (listed in the below graph) are so popular that they account for over 90% of Utah high school students enrolled in computer science courses since 2014-2015. Exploring Computer Science (ECS) in particular has enrolled more than 50% of Utah high school students since 2015-2016. Preliminary data suggests that Web Development 1 may be on the cusp of seeing large increases in enrollments as well. One positive aspect of this growth in ECS is that many more students are being exposed to computer science than ever before. Unfortunately, the state is not seeing a corresponding growth in more advanced computer science classes, which suggests that students are not staying interested enough in computer science to pursue further study. Nor do we expect a surge in Web Development 1 enrollment to lead to an increase in student enrollment in more advanced computer science classes, because the vast majority of Web Development 1 classes in Utah do not introduce computational thinking.



The growth in popularity of Exploring Computer Science, Computer Science Principles and Web Development 1 can most likely be attributed to recent changes to high school graduation requirements. These three introductory computer science classes may now be used towards high school graduation classes, and their enrollments increased soon after they started to "count" for graduation. Similarly, Computer Programming 1 has counted as a Math graduation credit for years, and it was historically the most popular computer science course in the state (until 2015). On the other hand, allowing more advanced computer science courses like Computer Programming 2 to count as a Science graduation credit has not noticeably affected their enrollments.



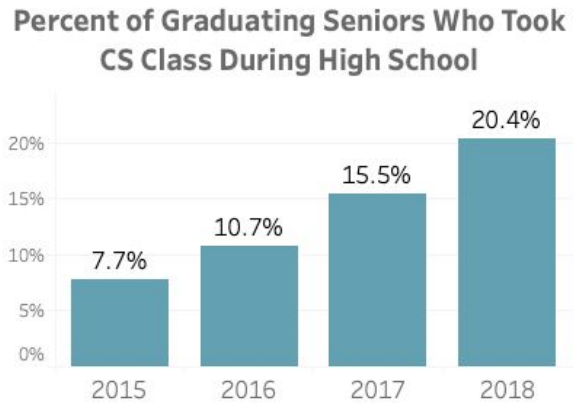
High school graduation requirements that accept computer science courses are:

- 1 Applied Math credit
    - Computer Programming
- 1 Science credit (since 2014-2015)
    - AP Computer Science
    - Computer Science Principles
    - Computer Programming 2
- 0.5 Computer Technology (phased out in 2016-2017)
    - Exploring Computer Science 1
- 0.5 Digital Studies graduation (since 2017-2018)
    - Computer Programming 1
    - Computer Science Principles
    - Exploring Computer Science 1
    - Web Development 1

The overall effect of allowing Exploring Computer Science (ECS) to count towards graduation has been dramatic. As illustrated below, enrollment in all high school computer science courses increased annually by more than 5,000 students each year for three consecutive years (2014-2016) until 2017-2018, when enrollment decreased by almost 1500 students.

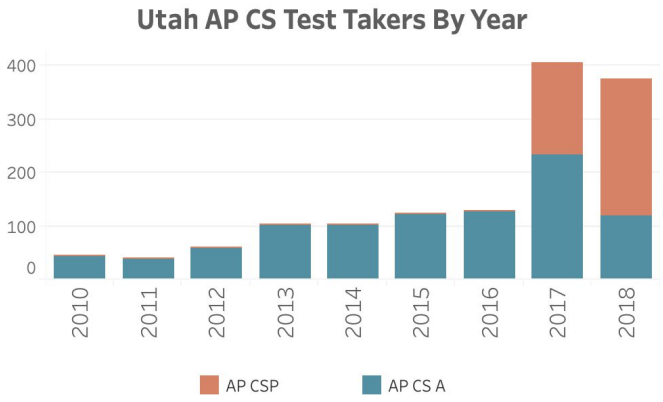
While enrollment in ECS classes grew faster than in other classes, most CS classes saw increased enrollments from 2012 to 2017. By 2016-2017, 20,501 unique students were enrolled in 22,551 CS courses (11% of all Utah high school students, compared to 1% in 2012-2013), including 13,634 in ECS (66% of the students taking CS courses). Given these increased enrollments, we were surprised that after gaining more than 5,000 students each year for three consecutive years, 2017-2018 enrollment in CS courses *decreased* by almost 1500 students with the roll-out of the new Digital Studies graduation requirement. Among the five most popular courses, only Web Development 1 and Computer Science Principles saw increases in enrollments in 2017-2018. A change in course coding may have contributed to the increase in Web Development 1, whereas the College Board's introduction of the AP Computer Science Principles exam in 2017 may have increased interest in the Computer Science Principles course. Another factor that may have contributed to this general decline in enrollment may include reduced incentives for teachers teaching CS courses like ECS.

Despite this one year decrease, more students in the state are graduating high school having taken at least one computer science course. Only 7.7% of high school seniors graduating in June 2015 had taken one or more CS courses during their four years of high school, whereas 20.4% of the seniors from the class of 2018 had taken at least one CS course. While the state is far from having all high school students take at least one CS class, the percentage of unique students exposed to computer science has increased dramatically since 2014.

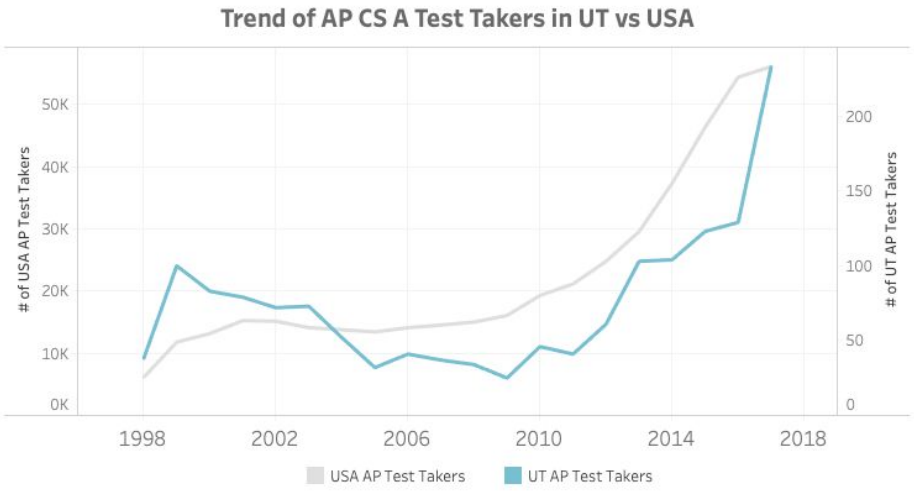


6.2.1. What are the long-term trends in Utah high school computer science?

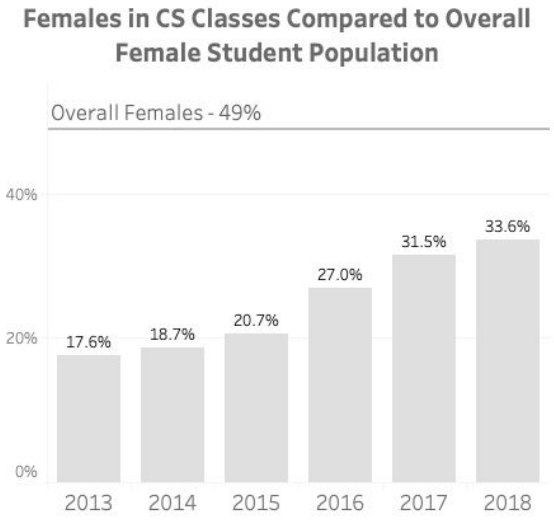
While fewer than 500 Utah students take the College Board Advanced Placement Computer Science (AP CS) exams each year, this data serves as a useful source of long-term trends. Statewide test data and demographics are available since 1998 for all AP exams offered in computer science. Similar to the four AP Physics exams which cover different areas of Physics with varying levels of difficulty, the College Board has offered three AP CS exams: AP Computer Science A, AP Computer Science AB (last offered May 2009), and AP Computer Science Principles (first offered May 2017). Utah Computer Programming 2 students would sometimes take either the AP Computer Science A exam or, prior to 2010, the more advanced AP Computer Science AB exam. In the 2016-2017 academic year, the College Board introduced AP Computer Science Principles (AP CSP), an introductory CS course that emphasized six computational thinking practices and seven big ideas of computer science rather than programming in Java. The AP CSP launch was the largest in College Board history. In Utah, not only did more Utah students took the AP CSP exam in May 2017 than had taken the AP CS A exam in the previous year, but the total number of Utah AP CS test takers exam more than doubled, from 129 in May 2016 (all AP CS A) to 405 in May 2017 (233 AP CS A / 172 AP CSP).



Prior to 2017, Utah had the most AP CS test takers in May 1999, after which, there was a gradual drop in AP CS test takers until 2012, when interest began to pick up again. This trend contrasts with national trends, where AP CS test takers were flat from 2001 to 2009, before starting to grow in 2010. This difference suggests that Utah may have been ahead of the curve in CS in 1999, but we lost ground between 2000 and 2011 and may have been slower than other states to grow CS in the past decade. More troubling, data since the introduction of the AP CSP exam suggests that Utah may once again be falling behind the nation. Whereas most of the rest of the nation continues to have more interest in all AP CS exams, Utah saw a decline in AP CS test takers in May 2018, when the total number of Utah AP CS test takers dropped down from 405 in May 2017 (233 AP CS A / 172 AP CSP) to 376 in May 2018 (120 AP CS A / 256 AP CSP). Most troubling is that the number of AP CS A test takers is below 2015 numbers, which may be a sign of Utah losing ground in more programming-intensive computer science classes, even as more Utah students are exposed to introductory computer science.



6.2.2. Which students are enrolling in Utah high school courses?



We compared course enrollments for these CS courses to the state's general student population demographics, available on the State Board of Education website and found that demographics of Utah high school CS students vary widely based on the class. Across *all* CS classes, female student enrollments have increased from 17.6% in 2012-2013 to 33.6% in 2017-2018. Racial/ethnic enrollments in high school CS classes approximately match the demographics for Utah's public school K-12 student population. In fact, while female students are still under-represented in computer science courses, most traditionally underrepresented minorities (with the exception of Pacific Islanders) are slightly over-represented in computer science courses. The diversity of these courses may reflect the presence of introductory computer science courses in larger, more urban districts and the lack of these same courses in smaller, more rural districts.

While these gains might initially paint a bright picture of an increasingly diverse and growing enrollment in Utah CS courses in high school, a deeper understanding of the data reveals the diversity is only superficial. We explore the data in depth in the remainder of this section.

Table 6.2.2.1. *Percentage of Students Enrolled in High School Computer Science Classes by Race / Ethnicity*

Demographics	UT Student Population	2013	2014	2015	2016	2017	2018
American Indian	1%	0.8	1.0	1.6	1.7	1.2	1.3
Asian	2%	4.2	3.3	3.0	3.4	2.7	2.7
Black/African American	1%	1.1	1.3	1.3	1.4	1.7	1.7
Hispanic/Latino	17%	12.6	14.2	15.3	16.9	17.4	20.3
Multi race	3%	1.5	2.0	2.4	2.5	2.5	2.4
Pacific Islander	2%	0.9	1.0	1.1	1.4	1.5	1.6
White	75%	78.9	77.2	75.3	72.7	73.1	70.0

*Public school enrollment demographics are for 2017 and come from the 2017-18 Utah State Board of Education Fingertip Facts.

Although female CS enrollments appear to be on the rise, most female students are enrolled in Exploring Computer Science (ECS), one of the most basic of computer science courses. If we count all CS classes except ECS, female enrollment only increased from 17.6% in 2012-2013 to 20.6% in 2017-2018. Female enrollments are even lower in courses where teaching a programming language is a large part of the course content (Computer Programming 1 and 2, Game Development 1 and 2, Mobile Development, Computer Science Principles, AP Computer Science, IB Computer Science, and Algorithms). More insights can be found by examining the three most popular CS classes that have a follow-up class (Computer Programming 2, Game Development 2, and Web Development 2). Female participation dropped dramatically between the first level CS class (2016-2017) and the second level CS classes (2017-2018). Over the past five years, 13.6% of male students in Computer Programming 1, Game Development 1, and Web Development 1 continue on to second CS classes, whereas only 8.3% of female students continue on to second CS classes.

Table 6.2.2.2. *2017-2018 Enrollment in High School CS Courses by Gender*

	ECS	All Other HS CS	Programming Courses	First Class in Sequence*	Second Class in Sequence	5 Year Continuation Rate
Female	39.5%	20.6%	18.9%	19.5%	13.0%	8.3%
Male	60.5%	79.4%	81.1%	80.5%	87.0%	13.6%

* Enrollments for the first class in the sequence are given for 2016-2017.

For all Utah students, enrollment in the second course is about 11.3% of the enrollment in the first course. Like male students, White and Asian students have higher continuation rates than average (see Table 6.2.2.3), whereas all other racial and ethnic demographics have lower continuation rates. The lowest rates of students taking a second CS course (compared to first course enrollments) have been among Native American (3.2%), Hispanic (6.8%), African American (6.8%) and Pacific Islander (9.0%) students.

Table 6.2.2.3. *2017-2018 Enrollment in High School CS Courses by Demographics*

	ECS	All Other HS CS	Programming Courses	First Class in Sequence*	Second Class in Sequence	5 Year Continuation Rate
White	67.6%	76.1%	75.8%	76.8%	85.1%	13.8%
Hispanic	22.2%	14.7%	15.0%	14.7%	6.9%	6.8%
Black	2.1%	1.3%	1.3%	0.9%	0.6%	6.8%
Pacific Islander	1.9%	1.0%	1.1%	0.7%	0.6%	9.0%
Asian	1.6%	3.2%	3.4%	3.5%	3.2%	16.2%
American Indian	1.6%	0.7%	0.7%	0.9%	0.5%	3.2%
Low SES	38.7%	33.4%		33.7%	19.5%	7.3%

Students from low socio-economic status (SES) households also take a second CS courses at a lower rate (7.3%) than female students (8.3%), but at a higher rate than some traditionally underrepresented races and ethnicities. 35.2% of Utah students qualify for free and reduced lunches, whereas an average of 34.5% of CS students over the past six academic years qualified. In 2017-2018, 38.7% of ECS students; 33.7% of Computer Programming 1, Game Development 1, and Web Development 1 students; and 19.5% of Computer Programming 2, Game Development 2, and Web Development 2 students came from economically disadvantaged households. Whereas participation of female students and students from traditional under-represented racial/ethnic backgrounds have increased over the past six years for more basic CS courses, the number of students from low socio-economic backgrounds has not increased in recent years. In fact, because absolute enrollment numbers across all demographics have increased in even the second CS courses, the percentage of students from low socio-economic backgrounds has actually decreased since 2012-2013.

In summary, the number of students completing CS courses has steadily increased in each of the past five years, with female enrollments nearly doubling in five years. Utah's evolving digital literacy high school graduation requirement (most recently called Digital Studies and fulfilled by one of four introductory CS courses or two business applications courses) has led to an increase in enrollments in introductory CS courses but has not pushed more under-represented students to enroll in more advanced CS courses. Unfortunately, because digital literacy is a half-year requirement, the vast majority of Utah students are taking a half-year version of ECS and CSP, which were originally created as year-long courses. The first half of these courses help students learn what computer science is and discover how it can be applied to almost any field, discipline, or interest area. However, students miss out on the bulk of the creative aspects of ECS and CSP by taking only half-year versions of these courses. Thus, the same graduation requirement that is encouraging many students to take their first CS course may be short-changing the students, causing them to miss out on the most creative parts of the courses.

6.3. How are teachers being prepared?

Because few CS majors and are choosing to become educators after graduation, Utah has filled the CS teacher gap by providing professional development to existing, in-service K-12 teachers. While anyone can teach computer science in elementary school, Utah in-service secondary teachers wishing to teach CS may apply for a State Approved Endorsement Plan (SAEP), permitting them to teach CS for two years while completing the endorsement requirements. Teachers who failed to complete the SAEP within two years were not eligible to apply for a new plan, and would no longer be allowed to teach CS classes until they completed the full endorsement.

Prior to 2013, Utah had a single CS endorsement that required CS courses (fulfilled by completing a CS major, a CS minor, or five college level CS courses) and a CS pedagogy course that was not taught at any Utah college. In 2013, the Utah Board of Education created three levels of CS endorsement:

- Exploring Computer Science (ECS)
- Computer Science Level 1
- Computer Science Level 2

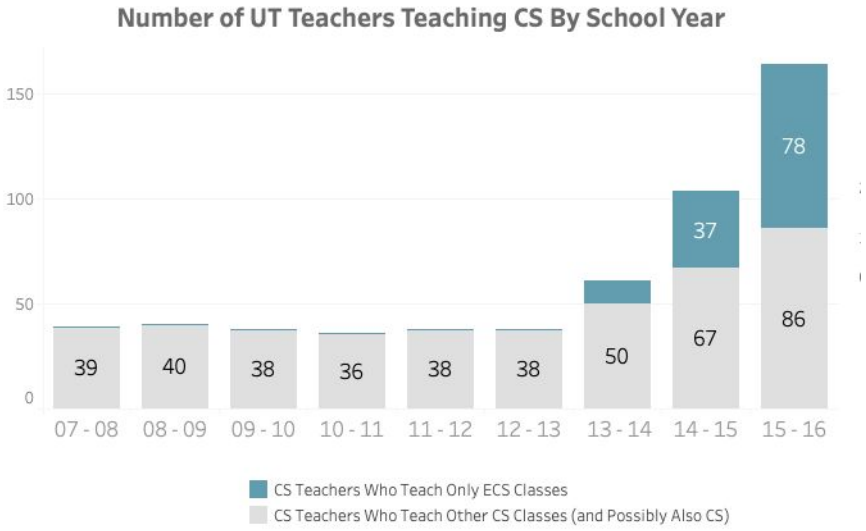
The first of these three endorsements allows a secondary teacher to teach Exploring Computer Science. Many new-to-CS teachers are able to complete this endorsement in less than a year. A CS Level 1 endorsement certifies a secondary teacher to teach first year CS classes like Computer Programming 1 and AP Computer Science Principles. Because this endorsement requires completing three additional college-level CS courses beyond what is required for the ECS endorsement, many new-to-CS teachers complete an ECS endorsement first before starting their CS Level 1 endorsement. A CS Level 2 endorsement currently requires completing everything in a CS Level 1 endorsement, plus two additional college-level CS courses. USBE staff have identified multiple free college-level CS courses for in-service teachers to earn the first two endorsements, but the coursework still requires a significant time commitment.

Utah also has several other IT-related endorsements: A+ Computer Repair, Cisco Certified Networking Associate, Database Development, Geographic Information Systems, Introduction to Information Technology, Linux, Microsoft Certified Professional, Multimedia, Network+, Security and Web Development. The requirements for all CS and IT endorsements can be found on the USBE Information Technology educator endorsement web site (https://www.schools.utah.gov/cte/it/educatorendorsements).

These endorsements may be added to any existing Utah teaching license, such as a secondary (6-12) or Career and Technical Education (CTE) license. The vast majority of Utah computer science teachers have specialized in another area (e.g., math or business) in college rather than computer science. Many have never taken a college-level computer science course, which makes computer science unusual in that many Utah high school teachers are not content experts in the area that they are teaching. By pursuing three separate endorsements consecutively instead of one single CS endorsement, a new-to-CS high school teacher now has six years to complete these endorsements instead of two years to complete the original CS endorsement (prior to 2013).

6.3.1. How many Utah teachers are teaching computer science courses?

As seen in the graph below, the number of Utah teachers teaching CS was relatively flat from 2007 to 2012. From 2013 to 2016, the number of CS teachers more than quadrupled, growing from 38 CS teachers in 2012-13 to 164 CS teachers in 2015-16. Most of the growth in high school teachers were in ECS endorsements, which were introduced in 2013 and were funded by a National Science Foundation grant from 2013-2016. However, this increase was not limited to ECS teachers; the number of teachers teaching CS courses requiring a CS Level 1 or Level 2 endorsement increased from 38 in 2012-13 to 86 by 2015-16.



Full-time equivalent (FTE) data is the number of full-time CS teachers that would be employed if all CS teachers taught exclusively CS courses. The ratio of FTEs to actual number of CS teachers correlates to how many CS classes are taught by the average CS teacher statewide. From 2007–2016, this ratio has remained below 50%, indicating that the average Utah teacher who taught CS classes spent most of their day teaching non-CS classes. Teachers teaching only ECS courses have a slightly lower ratio than those teachers teaching more advanced CS courses that require CS Level 1 and/or CS Level 2 endorsements (31% vs 33% in 2015-16). Thus, Utah teachers teaching CS may not consider themselves computer science teachers.

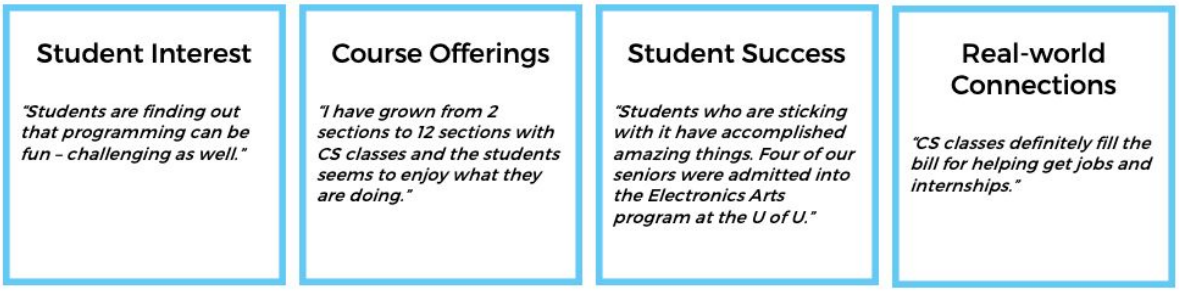
Finally, it is crucial that Utah CS teacher data be collected and analyzed for the years since 2016, especially as enrolments in high school CS courses have started to flatten and possibly decrease in 2018-2019.

6.4. What are teachers saying about teaching CS in Utah high schools?

6.4.1. What SUCCESSES have high school CS teachers had?

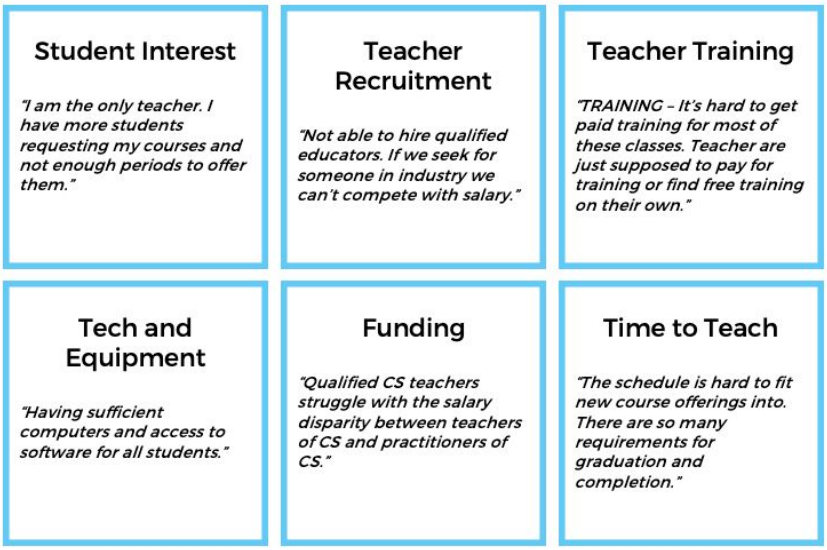
We asked high school CS teachers to share the successes that they have seen in their programs thus far. The most common successes were: high level of interest among students, a growing

amount of courses being offered, students finding success both in the classroom and beyond, and a fostering of real-world connections to help prepare students for the future. Below are the most common themes we saw in their responses, as well as a few sample comments we pulled from the surveys to illustrate each theme.



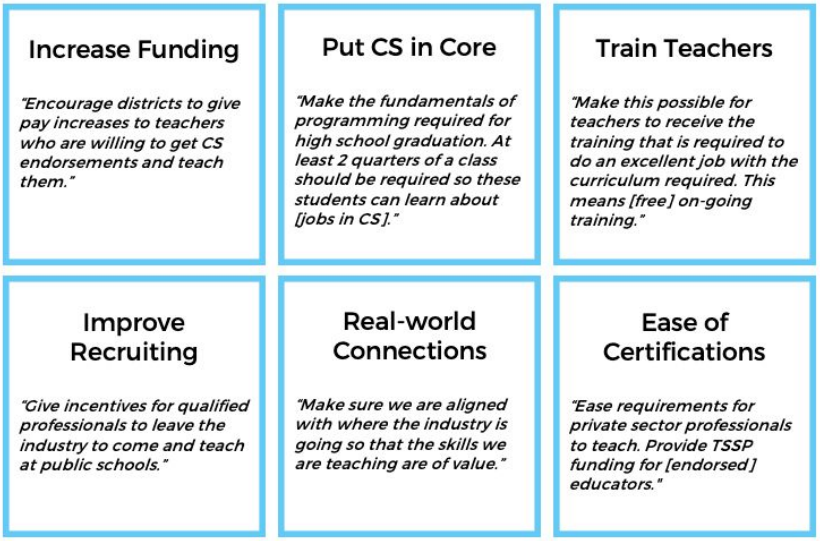
6.4.2. What CHALLENGES have high school CS teachers had?

We also asked high school CS teachers to share the challenges that they have experienced. The most common issues identified were: not enough courses to match demand of student interest, lack of qualified teachers, difficulty of training teachers, lack of technology and equipment, insufficient funding, and difficulties fitting CS courses into the schedule. We've listed a few comments pulled directly from the survey to further illustrate these common challenges faced by high school CS teachers.



6.4.3. What ADVICE do high school CS teachers have for policy-makers?

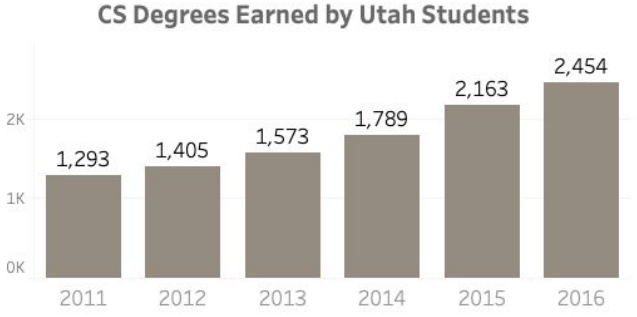
Finally, we asked high school CS teachers to share any advice they have for policymakers with regards to CS in high schools. The most common advice was to increase funding, include CS in the Core curriculum, train current teachers, recruit people from industry to be teachers, ensure that the CS curriculum aligns with where the industry is going, and ease the requirements of certification for CS teachers.



7. Higher Education Computer Science Degrees

The following data of all post-secondary degrees earned by students residing in Utah is taken from the National Center for Education Statistics (NCES) IPEDS Completions Survey for six consecutive academic years, up to 2015-2016. These computer science degrees include certificates, associates, bachelors, masters, and doctorate degrees.

As enrollment in high school computer science courses have increased, the number of computer science degrees earned by Utah students has also increased. While data is not yet available for the most recent two academic years, computer science departments report a continued growth in students both nationally and across Utah.



65% of computer science degrees awarded in the state of Utah are Bachelor's degrees. Across all types of degrees in Utah, women have earned only 10-12% of the CS degrees each year between 2011-2016.

In 2015-2016, only eight Utah universities awarded more than 100 CS Bachelor's degrees, with the remaining Utah institutions awarding a combined 100 CS Bachelor's degrees. Most notably, Western Governors University, an accredited online competency-based university that serves students in all 50 states and several territories, awarded more than 50% of the 2,525 Bachelor's degrees. However consulting with WGU revealed that only 89 of the 1,353 WGU CS Bachelor's degrees were awarded to students living in Uta. Similarly, only 14 of Western Governors University's 305 CS Master's degrees were awarded to Utah students.

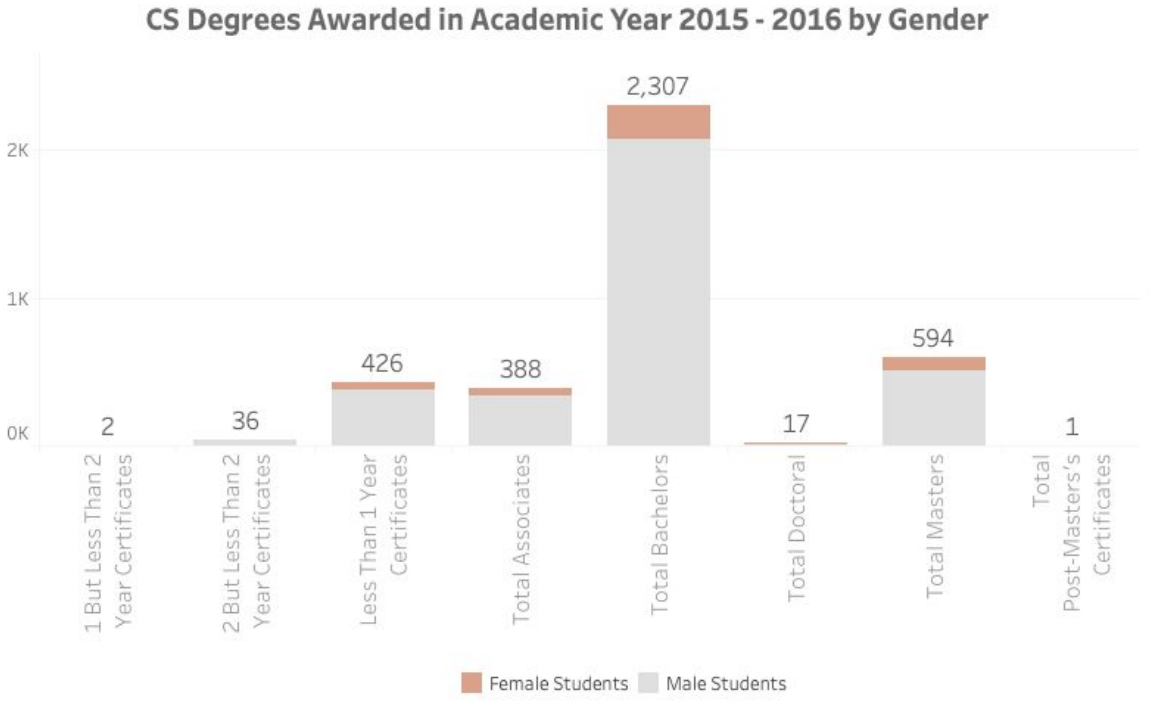


Table 7.1. *CS bachelor degrees awarded at Utah Higher Ed. institutions in 2015-2016*

Institution	Bachelor CS Degrees Awarded in Academic Year 2015-2016
Western Governors University	1,353 total (89 awarded to Utah residents)
Brigham Young University	219
Utah Valley University	194
University of Utah	148
Utah State University	147
Weber State University	140
Stevens-Henager College (all campuses)	115
Neumont University	109
All other universities and colleges	100
All Utah Higher Ed Institutions	2,525 (1,261 by Utah students)

According to the National Center for Education Statistics (NCES) IPEDS Completions Survey, the following Utah institutions have awarded the most computer science degrees from 2011-2015:

Table 7.2. *Utah higher education institutions awarding the most computer science degrees*

Institution	Degrees Awarded from 2011-2015	% Females
Western Governors University	4,613 total = 303 Utah + 4,281 non-Utah	10.08%
University of Utah	1,113	12.58%
Brigham Young University	1,064	6.20%
Utah Valley University	1,040	12.88%
Salt Lake Community College	813	12.92%
Weber State University	802	7.61%
Utah State University	742	12.94%
Stevens-Henager College (all campuses)	478	13.18%
ITT Technical Institute	401	13.72%
Neumont University	329	7.60%
All Utah Higher Ed Institutions	12,533	11.43%

Of the 4,613 Western Governors University, 4,281 (92.8%) were Bachelors and Masters degrees earned 2011-2015 by students ***not*** living in Utah.

Table 7.3. *CS Bachelor degrees awarded by Western Governors University*

	2011	2012	2013	2014	2015
WGU Utah Degrees	34	49	62	73	85
WGU Non-Utah Degrees	332	531	812	1173	1433

The National Center of Education Statistics data is often used as the source for reporting degrees earned in each state. Due to its being headquartered in Utah, Western Governors University (WGU) graduates are all counted toward Utah. However, WGU serves students in all 50 states and several US territories. Because WGU awards more CS degrees than all the other Utah higher education institutions combined, and because so few of those degrees are awarded to students actually residing in Utah, the number of Utah CS degrees awarded to Utah graduates is often grossly overestimated.

In summary, even discounting WGU degrees awarded each year to students who have never lived in Utah, Utah's higher education institutions continue to award more computer science degrees each year. Just over half of these degrees are bachelor's degrees. One quarter of all Utah CS degrees are being awarded by three schools (University of Utah, Brigham Young University and Utah Valley University). From 2011-2015, female students earned only 10-12% of the CS degrees awarded. While a few higher education institutions averaged fewer than 10%, some colleges and universities, especially those with smaller classes, are having successes drawing a higher percentage of female students. As CS enrollments continue to grow at colleges across the state, departments should study and adopt best practices for promoting the retention and success of students from all backgrounds and demographics.

8. Conclusion

The need for individuals with computer science skills is greater than ever in Utah. This report reveals that there are even fewer Utah graduates with CS-related degrees than previously believed. At the same time, there is a growing need from Utah's robust local economy for individuals with computer science training (see Appendix, Section 11.1). Fortunately, the landscape of computing education in Utah is changing in a positive way.

Over the past decade, Utah has seen a surge in enthusiasm, courses offered, and student enrollment in K-12 computer science classes. Computer science has been introduced into at least 22% of Utah elementary schools and 28% of middle/jr. high schools, with a fifth (20.4%) of the graduating class of 2018 having taken at least one CS course in high school. Female participation in CS high school courses has nearly doubled, and under-represented minorities enroll in CS courses at levels consistent with their population representation in Utah. There is a growing sense across the state that all students would benefit from learning some computer science and, at a minimum, all Utah K-12 schools should offer computer science. In fact, nearly 90% of all educators who participated in this study indicated that all students need computer science knowledge prior to graduating.

There is both growing interest in and support for computer science education in Utah. For example, this past academic year, the Utah State Board of Education released guidance for local school districts on establishing standards for computer science in K-12. For the past 2 years, the governor's office of Economic Development has provided funding for local education agencies interested in increasing computer science participation in its schools. These funds have been used by dozens of local education agencies throughout urban and rural Utah,

bringing computer science to thousands more students statewide. Local industries have likewise committed to match funding efforts if supported by the Utah legislature. Finally, the Utah State Board of Education recently created a position dedicated to overseeing computer science education. Over the next several years, these support mechanisms should work together to bring greater attention to and interest in computer science education in Utah.

Despite this growth, computer science at the K-12 level remains an opportunity for a limited number of students. A large number of elementary and middle/jr. high schools do not yet offer CS classes, while others do not have the capacity to support all the students who wish to enroll. Nearly 4 out of every 5 graduating seniors still have not completed a computer science course. What's more, the increase in CS participation may be superficial, as the vast majority of students are only participating in .5-credits of new introductory CS courses. In some cases, these .5-credit courses are shortened versions of year-long courses, and the shorter versions fail to adequately prepare students with the 21st-century skills to compete in Utah's workforce. In addition, fewer students are actually completing more advanced CS courses. Even though these more advanced computer science courses can count toward the of a high-school science credit, they are rarely used as such. This may be due to competing priorities, in which computer science does not qualify as a science credit for other opportunities, such as the Regent's scholarship. Or, as revealed in educator focus groups, it may be due to misunderstandings by school counselors about the importance of computer science and what requirements it might fulfill.

The keys to successful computer science education in Utah may have been revealed in feedback provided by educators in this study. First, the vast majority of educators at all levels (elementary, middle, and high school) indicated that they are insufficiently prepared to teach computer science. They do not receive computer science education in their teacher preparation programs. What's more, many teachers responded that they must seek out and pay for their own professional development. In many cases, there is little incentive to increase their capacity to teach computer science beyond personal interest. In some cases, science and math teachers would actually lose salary by teaching computer science. Computer science is not treated the same as other STEM subjects from a compensation perspective despite the fact that most school districts indicate high difficulty in finding and retaining qualified individuals.

What is needed to better support teachers, counselors, administrators, and LEAs who wish to increase their capacity and offer more CS classes? According to the educators who contributed to this report, there may be several factors that could lead to the successful implementation of computer science education in Utah. Namely:

1. Require computer science education at each level of schooling (elementary, middle, and high school), with at least a year of computer science for graduating high school seniors;
2. Provide effective training for preservice teachers and professional development for inservice teachers;

3. Increased funding for CS-related resources and courses;
4. Teacher incentives to teach CS courses;
5. State-level support and direction on what CS concepts to teach at each level.
6. More consistent data collection and tracking of CS courses and student enrollments in elementary, middle and high schools across the state can help identify best practices that other LEAs can adopt.

Utah is in a good position to promote computer science education, with increasing interest from students, educators, policy-makers, legislators and industry. However, the current level of support does not match the increasing local needs. As we look to the future, addressing the aforementioned factors will be necessary in order to maintain pace with Utah's needs to support computer science education, an important component of a 21st-century education.

9. References

Glaser, B., & Strauss, A. L. (1967). *The discovery of grounded theory: Strategies for qualitative research*. Chicago: Aldine De Gruyter.

Rich, P. J., Browning, S. F., Perkins, M., Shoop, T., Yoshikawa, E., & Belikov, O. M. (2018). Coding in K-8: International Trends in Teaching Elementary/Primary Computing. *TechTrends*. doi:10.1007/s11528-018-0295-4

10. Glossary

Throughout this document, there are several terms that are specific to computer science education, including specialized software. These are defined below for those unfamiliar with these products and programs.

Arduino: A programmable micro-controller created with the intent of allowing affordable and easy hardware development. Arduinos are licensed under a license that allows them to be produced and sold by anyone (LGPL or GPL). At under $10 each, they are extremely affordable for classroom use, though they also require additional hardware, such as breadboards, wiring, and external sensors. Arduinos are generally programmed using either Java or a C-based language. See https://www.arduino.cc/.

BeeBot: An all-in-one programmable robot intended to teach pre-K aged children how to code using sequence-based programming. BeeBots are shaped like bees, have wheels with which to move, and an array of buttons on its back. Students enter in the sequence of commands that they would like the BeeBot to follow and then execute those commands. See https://www.terrapinlogo.com/beebot.html.

BootUp: A non-profit organization dedicated to training elementary teachers how to teach fundamental computing concepts to children in K-6. BootUp partners with school districts to designate a district-based coach, and trains teachers through a 3-year program of continuous professional development. See https://bootuppd.org/.

Code-a-Pillar: A programmable robot intended for pre-school aged children. Code-a-pillar is programmed by placing the actual physical components in order. Physical components each appear as a section of a caterpillar. See http://www.codapillar.com/.

Code.org: A non-profit organization dedicated to exposing all students worldwide to coding. Code.org created and popularized the "hour of code," each December in which students participate in a series of coding challenges. Since its inception in 2013, Code.org has helped over 15 million students experience at least an hour of code worldwide. Code.org has expanded its original offerings by creating semester and year-long courses that can be used in elementary (CS Fundamentals), middle (CS Discoveries) and high school (CS Principles). See https://code.org.

Creative Coding: A semester-based, Utah-based course taught in early secondary education as an introduction to coding using a real software development platform. Students design, develop, and publish mobile games. See https://www.schools.utah.gov/file/45a97db2-5911-473a-93d6-28ba0fe3cf1a.

CS Discoveries: A semester or year-long course created by Code.org for 6th-10th grade students that teaches a broad-based understanding of computer science, including topics such as programming, physical computing, web-development, and data analytics. See https://code.org/educate/csd.

CS Principles: A year-long introductory high school (9-12) course created by Code.org meant to provide a rigorous entry to computer science that includes: the internet, digital information, introductory programming, big data, and application development. The course can be offered as an Advanced Placement (AP) course, if desired. See https://code.org/educate/csp.

Dash & Dot: Programmable robots created with the intent to teach young children how do program robots by issuing a series of commands. Dash is a mobile robot with 3 wheels and various sensors, while Dot is immobile, but can communicate with Dash. See https://store.makewonder.com/.

Exploring Computer Science: A year-long, introductory course created as the result of a National Science Foundation grant, intended to provide a broad overview of computer science, including programming, networking, web-design, data analysis, human-computer interaction, and robotics. See http://www.exploringcs.org/.

The Foos: A tablet-based app created to teach young children how to code. Children control "foos" (cute creatures) to solve a series of puzzles using a block-based programming language. See https://codespark.com/.

HTML, CSS: Acronyms for the two languages in which most web pages are coded (Hypertext Markup Language) and styled (Cascading Style Sheets). HTML provides the structural framework that is interpreted by common browsers for web presentation.

Javascript: A high-level programming language used in web development. By several measures, javascript is the most used programming language worldwide. It is used by beginning and professional programmers alike.

Kodable: A tablet-based app intended to teach computational logic to early elementary and preschool-aged children. See https://www.kodable.com/.

Lego: Programmable lego-based robotic-kits created and sold by the Lego group®. Different kits used in Utah schools include Lego Mindstorms and Lego WeDo. See https://www.lego.com/en-us/mindstorms.

Makey Makey: An inexpensive invention kit that enables turning almost anything conductive into a "button," that can control a computer (including bananas, playdough, pencil drawings, and more!). See https://makeymakey.com/.

Micro:bit: A single-board micro-controller created by the British Broadcasting Company with the intent of providing an affordable device that all students could use to learn to program. At under $20/ea. Micro:bits are some of the most affordable programmable hardware available to teachers and students. Micro:bits can be programmed using a web-based editor in either Blockly (a block-based programming language), javascript, or python. See https://microbit.org/.

Osmo: An ipad-based game that combines physical components with software development to teach simple sequence-based programming. See https://www.playosmo.com/en/.

Ozobot: A small programmable robot that moves along colored lines. The robot "reads" the lines and executes a command as instructed by its programming for that color of lines. A series of curricular units have been created for teaching coding to early elementary students using Ozobot. See https://ozobot.com/.

Project GUTS: A middle-school curriculum meant to teach science and computer science in an inquiry-based manner. Students are encouraged to "think scientifically" and use data and computer science to investigate STEM-related questions. See http://www.projectguts.org/.

Project Lead the Way: A non-profit curricular program created with the express purpose of preparing pre-k—12th grade students to perform inquiry-based investigations in science, technology, engineering, and math (STEM). PLTW partners with schools and districts to train teachers to teach their specific lessons and incorporate these with the school's curricula. See https://www.pltw.org/.

Python: A dynamic, high-level programming language. Unlike many programming languages, python uses white space to indicate code hierarchy, eliminating the need for much of the punctuation used in other languages (semicolons, curly brackets). This combination of high-level, dynamically-typed coding and white space lends itself to rapid development and easy interpretation. As such, python is simple to learn, but it is also powerful and is used by beginners and professionals alike. In the past 5 years, python has become one of the most sought-after computer languages worldwide and is used for development across a variety of contexts, such as web development, data analytics, programming of simple hardware, and more. See https://www.python.org/.

Raspberry Pi: An inexpensive micro-processor (i.e., computer) created with the intent of enabling easy experimentation and creation of programmed devices. Created by the Raspberry Pi foundation, an organization dedicated to teaching CS education worldwide. Over 20 million Raspberry Pis have been sold since its introduction in 2014, making it the best-selling British computer . See https://www.raspberrypi.org/.

RunMarco: A web-based game built with the intent of teaching everyone to code. RunMarco is programmed through a block-based app wherein Marco or Sophia have to solve a series of increasingly difficult puzzles. See https://runmarco.allcancode.com/.

Scratch: A block-based coding environment created by the MIT media lab to help young children learn to code. This is the most-used coding environment to teach elementary computing throughout the world. Freely available at https://scratch.mit.edu.

ScratchEd: An online community dedicated to helping teachers learn and share curricular resources for teaching Scratch. Maintained by Karen Brennan, a professor in the Harvard Graduate School of Education. See http://scratched.gse.harvard.edu/.

ScratchJr: A tablet-based coding environment intended to to teach emergent readers and early elementary students sequence and event-based coding. While inspired by the Scratch programming environment, ScratchJr was created and is maintained by a different entity. See https://www.scratchjr.org/.

Snap!: A block-based programming language that is a derivative of the openly-licensed Scratch language. Snap! allows for more advanced programming concepts, such as multi-dimensional arrays. Some of the concepts first introduced in Snap! (originally called "Build Your Own Blocks") were later introduced into Scratch (e.g., variables). Several specialized standalone open programs have been built on the Snap! Interface. See https://snap.berkeley.edu/site/.

Sphero: A programmable, spherical robot. While Sphero can be used as a remote-controlled robot, it can also be programmed using a number of different mobile platforms. These platforms include block-based, javascript, and python programming options. SpheroEdu offers many options for integrating Sphero into school curricula and provides training on how to program Sphero. See https://www.sphero.com/education/.

Swift Playgrounds: An tablet-based Apple® product created with the intention of teaching programming fundamentals through Apple's high-level Swift programming language. Swift Playgrounds presents users with a series of increasingly difficult challenges to learn coding principles such as loops, conditionals, functions, variables, arrays, etc. See https://www.apple.com/swift/playgrounds/.

Touch Develop: A touch-based coding development environment meant for iOS and Android-based devices. The project was created by Microsoft and officially retired in 2019 in favor of MakeCode (the same editor used to code micro:bit). See https://makecode.com/touchdevelop**.**

Tynker: A proprietary, block-based online platform for teaching fundamental programming. Inspired by Scratch, it is similar, but offers greater animation options, built-in challenges, and classroom management tools for teachers. See https://www.tynker.com/.

Vex Robotics: An entire ecosystem of robotics parts with a programmable interface built with the intent of helping everyone learn how to create and control robots. Vex robotics holds annual state, national, and worldwide competitions for middle and high school students. Robots are programmed using the C-based ROBOTC language, with both a drag and drop or more versatile text-based interface. See https://www.vexrobotics.com/.

11. Appendix

11.1. Current Utah CS Industry

The CS industry in Utah is well-established and flourishing. Because Computer Science (CS) and Information Technology (IT) are closely related, some of the reports reviewed include both IT and CS industry data, which are both flourishing. For example, Utah's Department of Workforce Services report focuses specifically on "IT coding workers" and CS occupations (Utah Workforce Services, Knold, Smith, & Stahle, 2018, p.3). CBRE's Research defines 'Tech talent' as "a group of highly skilled workers in more than 20 technology-oriented occupations driving innovation across all industry sectors" (CBRE Research, 2017, p.3). CompTIA's report also includes all 'Tech talent' and defines it as "of technology professionals working in technical positions" (CompTIA, 2018, p.6).

Forbes magazine recently wrote: "As for the area that tech entrepreneurs have taken to calling Silicon Slopes... the corridor that connects Provo to Salt Lake to Ogden is chock-full of tech companies and startups...an ecosystem of entrepreneurial activity has sprung up against the backdrop of the Wasatch Mountains" (Feldman, 2017, para. 2-3). This kind of "entrepreneurial ecosystem" has many positive economic implications as well. In 2016, The Deseret News reported, "Utah has the best state economy in the U.S., according to a report published Monday by WalletHub, a national financial research and consulting organization" (Jacobsen, 2016, para. 1). Such descriptions are supplemented by national and state level reports on the current tech industry in Utah.

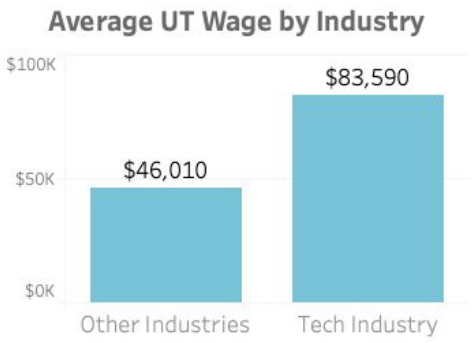
11.1.1. What is the economic impact for the state?

The influence of tech companies on the Utah economy is substantial. The tech industry in 2017 accounted for 10.2% ($14.9 billion) of the state's approximately $150 billion in economic activity, where net tech employment accounts for only 8.6% of the overall workforce. Furthermore, it appears that it will continue on that way (CompTIA, 2018, p.60). Tech occupations form an integral, and even fundamental part of Utah's economy. The state's economy is booming and much of the credit goes to companies that employ persons with skills in computer software and other related tech skills.

An article from the Deseret News references the Computing Technology Industry Association statistics in regards to Utah's economy, reporting that there are 6,531 tech businesses (firms with payrolls) in Utah and that Utah had 26,020 tech job postings in 2017 (Raymond, 2018b). That means that about 20,000 people are hired for work in a tech occupation each year in a variety of companies. It's no wonder Utah's 2018 workforce services report calls Utah IT "a fast-growing and highly successful market. It is thriving with labor that is partially of a lesser education level than other successful IT markets" (Utah Workforce Services et al., 2018, p.20).

Utah's IT ecosystem is still small and lacking when compared to other hubs around the nation including San Francisco, California and Seattle, Washington. However, the workforces services report states, "Though Utah's IT environment may not be the 'top tier', it still provides great benefit to the Utah economy" (Utah Workforce Services et al., 2018, p.20).

11.1.2. What is the economic impact for individuals?



With all the benefits of the Utah CS industry to the state's economy, some of the highest benefits are received by the workers. Utah currently has high demand for workers with CS skills. Utah's annual tech job salary average was reported at $83,590 in stark comparison to the $46,010 overall average salary for state wages (CompTIA, 2018, p.60). These figures suggest that a tech occupation would yield a salary that nearly doubles the state average. Table 12 below shows some of the internal pay comparisons in the Utah core IT industry, and how they compare to other tech cities. There are many different occupations within core IT as stated in the workforce services report, "Computer and mathematical occupations make up half of the core IT industry's occupational mix; and, within that, two-thirds are programmer, analyst and developer occupations." Those varying occupations have differing salaries to accompany them. However, the report also suggests that, "There are not a lot of occupational pathways that yield such a high average wage with education below a bachelor's degree. The broader Utah economy benefits when lesser-educated workers have occupational avenues that deliver middle-income earnings" (Utah Workforce Services et al., 2018, p.20). Source: Computing Technology Industry Association (Utah Workforce Services et al., 2018, p.18).

Purchasing Power Adjusted Core IT Occupation Wages by Education - 2016 American Community Survey

	SLC Area	SF Area	Raleigh	Austin	Seattle	New York
Below Bachelor's	$84,390	$112,056	$97,701	$90,707	$62,174	$62,378
Non STEM Degrees	$75,023	$97,761	$115,043	$76,523	$128,440	$101,293
STEM	$92,172	$110,419	$126,159	$117,442	$126,420	$102,082

This is not to say that a bachelor's degree is not useful, especially a STEM-focused bachelor's. In fact, the majority of the Utah Tech workforce have at least a bachelor's degree and one of the reasons for attaining these degrees is seen in Table 12 above. That being said, the report states, "Bachelor's degree and higher labor is used, but it appears that much of Utah's IT industry thrives with labor of a lesser credential" (Utah Workforce Services et al., 2018, p.21). "Table 5 reveals that close to half of Utah's tech workforce have been educated to the certificate, vocational training and associate degree levels" (Utah Workforce Services et al., 2018, p.20) (Appendix C). As Mark Knold, supervising economist for Workforce Services commented, "There are ample opportunities for job seekers, with or without a four-year degree, to establish a rewarding and lucrative career in Utah" (New study of Utah's "core IT" sector, 2018, para. 6). (Utah Workforce Services et al., 2018, p.14).

Core IT Workers by Educational Attainment - 2016 American Community Survey

	SLC Area	SF Area	Raleigh	Austin	Seattle	New York
Below Bachelor's	44%	7%	15%	26%	14%	16%
Non STEM Degrees	11%	12%	14%	14%	11%	20%
STEM	46%	81%	71%	60%	74%	65%

11.1.3. What is the current trajectory of Utah's CS industry?

Not only is Utah's CS industry well-established and flourishing, but growing as well. According to a Deseret News article on April 27, 2018, "Utah's growing tech sector continues to be an economic bright spot for the state and helped drive the nation-leading 3.7 percent job growth reported in March" (Raymond, 2018a, para. 19). The Deseret News reports that there was a 42.2 percent growth in emerging tech job postings last year (Raymond, 2018b). This job growth is impacting the economy as well as individuals and families directly. According to Utah's Employment Summary this growth "[added] 48,000 jobs to the economy since March 2017. Utah's current employment level registers 1,501,800...March's seasonally adjusted unemployment rate remained unchanged from the prior month at 3.1 percent... [compared to] the national unemployment rate...at 4.1 percent" (Utah's Employment Summary: March 2018, 2018, para. 2). This demonstrates that the creation of tech jobs in thriving tech companies is driving down unemployment and giving more people an opportunity to have a job that supports them and possibly their families.

According to CompTIA's report, in Utah there are 135,500 jobs in tech, with 4,680 of those having been added in the previous year (CompTIA, 2018, p.60). The majority of those jobs are held by software and web developers at 16,350, with computer support specialists at 8,660, and network architects, administrators, and support specialists as the next most common at 6,750 (CompTIA, 2018, p.60). These occupations may be very different in their daily tasks and responsibilities, but they share the need to understand, at least on a general level, computers and programming languages.

The usefulness and desirability of these skills is not limited to Utah alone. On a national level, Tech companies employed 7.0 million in 2017 with an increase of 128,500 from 2016 (includes tech and non-tech jobs). Tech occupation jobs in both tech and non-tech companies reached 7.4 million in 2017 with an increase of 128,700 workers from 2016 (CompTIA, 2018, p.10). Within tech companies, packaged software companies had the greatest growth in jobs at 5.1% from 2016 to 2017 (CompTIA, 2018, p.10). This shows that Utah's growing tech-linked economy is a reflection of the nation's economic demands and successes as well. This means that those possessing CS skills are not limited geographically or categorically in their career choices. They are afforded options of place and position.

11.1.4. What is the state of CS worker migration in Utah?

A report was conducted by the Utah Workforces Services (Utah Tech Report) which explores and focuses on Utah's Information Technology (IT) landscape and the IT coding workers. They

define "Core IT" as jobs that primarily employ IT coders in fields such as software publishing and computer systems design segments. The report confirms that Core IT is a fast-growing industry in Utah.

To determine the amount of IT workers moving into and out of Utah, the report stated, "Net overall Utah IT labor migration is neutral; in- and out-migration largely offset…However, Utah IT labor in-migration carries in lower education levels (50 percent bachelor's or better) than what migrates out (75 percent bachelor's or better)…When restricting the migration view to just coding workers, and then further to only STEM-educated coding workers, Utah becomes a net labor exporter" (Utah Workforce Services et al., 2018, p.3). Utah has many IT coders leaving the state for better opportunities, particularly for those with STEM degrees. In regards to IT migration the Utah Tech Report stated, "According to 2013-2016 ACS data [Appendix C], an average of 1,052 individuals came to Utah into these coding occupations per year. In contrast, 1,505 left the state, for an average net outflow of 453 each year. While Utah has overall IT employment in-migration, much of this is not coders. The state is a net exporter of computer coding talent." (Utah Workforce Services et al., 2018, p.12).

The CBRE report also analyzed where the talent workers are coming from and where they are headed in terms of degrees awarded. Salt Lake City, Utah was listed as 13,155 tech degrees awarded from 2011-2015 while tech jobs added between 2012-2016 totaled 9900. This difference of awarded degrees and tech jobs added is referred to as a brain gain or drain. Salt Lake City, Utah is listed as a brain drain of -3255 (CBRE Research, 2017, p.18). However, Dr. Helen Hu from Westminster college found that the tech degrees awarded in the report may have also included some degrees awarded to students outside of Utah. These were students that graduated with a degree from Western Governors University because they did an online program and were not Utah-awarded degrees. Dr. Helen Hu wrote, "The 13,155 degrees are likely overcounting by 4281 (WGU Bachelors and Masters degrees earned 2011-2015 by students not living in Utah). The number of tech degree earned by Utah students should be 8874 and a "brain gain" of 1026" (Hu, 2018).

	2011	2012	2013	2014	2015
WGU Utah Degrees	$84,390	$112,056	$97,701	$90,707	$62,174
WGU Non-Utah Degrees	$75,023	$97,761	$115,043	$76,523	$128,440

11.1.5. Future of the Industry

All these data might suggest that Utah residents give ourselves a figurative pat on the back. Utah Business Magazine noted: "So, kudos to us. But before we wax too self-congratulatory, we'd do well to notice some small clouds on the horizon. Yes, Utah, even with its economic victories, faces a few challenges as well. We have a shortage of skilled knowledge workers" (Andra, 2018). Not only skilled workers, but workers in general. From 2016 until 2026, an estimated 545,000 people will retire, change careers, or have job displacement from tech jobs. Replacing baby boomers is also a concern mentioned in the report (CompTIA, 2018, p.7).

11.2. Total unique schools represented per school district and response rate

The survey described in Section 3 (results provided in Sections 4, 5, and 6) had 468 usable responses from 353 unique schools from 39 Local Education Agencies (LEAs). This table lists the response rate for each of the 42 LEAs in Utah.

LEA (District, Charter)	Unique School Responses	Total schools in LEA	% Response
ALPINE DISTRICT	49	84	58.33%
BEAVER DISTRICT	1	5	20.00%
BOX ELDER DISTRICT	5	24	20.83%
CACHE DISTRICT	6	25	24.00%
CANYONS DISTRICT	23	48	47.92%
CARBON DISTRICT	4	10	40.00%
DAGGETT DISTRICT	1	3	33.33%
DAVIS DISTRICT	43	87	49.43%
DUCHESNE DISTRICT	6	14	42.86%
EMERY DISTRICT	1	10	10.00%
GARFIELD DISTRICT	2	9	22.22%
GRAND DISTRICT	2	3	66.67%
GRANITE DISTRICT	59	91	64.84%
IRON DISTRICT	4	15	26.67%
JORDAN DISTRICT	16	53	30.19%
JUAB DISTRICT	2	5	40.00%
KANE DISTRICT	1	9	11.11%
LOGAN CITY DISTRICT	5	8	62.50%
MILLARD DISTRICT	3	10	30.00%
MORGAN DISTRICT	0	4	0.00%
MURRAY DISTRICT	2	10	20.00%
NEBO DISTRICT	5	44	11.36%
NORTH SANPETE DISTRICT	1	8	12.50%
NORTH SUMMIT DISTRICT	1	3	33.33%
OGDEN CITY DISTRICT	5	21	23.81%
PARK CITY DISTRICT	4	7	57.14%
PIUTE DISTRICT	1	3	33.33%
PROVO DISTRICT	1	21	4.76%
RICH DISTRICT	2	4	50.00%
SALT LAKE DISTRICT	13	38	34.21%
SAN JUAN DISTRICT	3	12	25.00%

SEVIER DISTRICT	6	12	50.00%
SOUTH SANPETE DISTRICT	0	8	0.00%
SOUTH SUMMIT DISTRICT	1	4	25.00%
TINTIC DISTRICT	0	4	0.00%
TOOELE DISTRICT	4	25	16.00%
UINTAH DISTRICT	6	12	50.00%
WASATCH DISTRICT	4	7	57.14%
WASHINGTON DISTRICT	19	46	41.30%
WAYNE DISTRICT	2	4	50.00%
WEBER DISTRICT	8	46	17.39%
CHARTER	32	117	27.35%

11.3. High School Computer Science Courses

The table below shows a complete list of high school computer science courses offered in Utah in 2017-2018, ordered from most to least enrollment.

1. Exploring Computer Science 1	10. Gaming Development Fundamentals 2
2. Computer Programming 1[1]	11. Exploring Computer Science 2
3. Web Development 1	12. Web Development 2
4. Computer Science Principles	13. HTML5 Application Dev. Fundamentals
5. Gaming Development Fundamentals 1	14. IB Computer Science SL 2
6. Computer Programming 2	15. Algorithms and Data Structures
7. Mobile Development Fundamentals	16. IB Computer Science SL 1*
8. AP Computer Science Principles	17. IB Computer Science HL 1*
9. AP Computer Science	18. IB Computer Science HL 2*

*zero enrollment 2017 - 2018

The Utah State Board of Education also has one middle school elective course classified as a computer science course: Creative Coding. While only recently introduced in 2016-2017, it is growing in popularity and would be ranked second on the above list if included. In 2017-2018, 4159 middle school students enrolled in Creative Coding.

Computer Programming 1 and 2 (second and sixth on the above list) have been offered as one-year courses for the past decade. LEAs who wished to offer half-year versions would split Computer Programming 1 and offer Computer Programming 1a and Computer Programming 1b.

Starting in 2018-2019, they were restructured and renamed into two half-year courses (Computer Programming 1 and Computer Programming 2) and a full-year Computer Programming 3 course, as were the Web Development courses:

Course Name	Course Length and Detail
Web Dev 1	0.5 (previously known as Web Dev 1a)
Web Dev 2	0.5 (previously known as Web Dev 1b
Web Dev Capstone	1.0 (previously known as Web Dev 2)
Computer Programming 1	0.5 (previously known as Comp Prog 1a)
Computer Programming 2	0.5 (previously known as Comp Prog 1b)
Computer Programming 3	1.0 (previously known as Comp Prog 2)
Computer Science Principles 1*	0.5
Computer Science Principles 2*	0.5
Digital Media 1a will now be Digital Graphics Art Intro	0.5
Digital Media 1	0.5 (previously known as Dig Media 1b)
Digital Media 2	1.0 (no change)

** Computer Science Principles is currently being considered for this change. The reason for this change is to facilitate the Digital Studies Graduation requirement as well as a hope to gather better data on student enrollment and course offerings. When this change is completed, a new set of strands and standards will be posted. For 2019/2020 continue with the current 1.0 course – plan on this mentioned change taking place for 2020/2021.*

This landscape report has used the 2018-2019 course names throughout, but any future report or data analysis will need to account for these new course namings.

11.4. How can Utah produce more educated, diverse CS professionals?

At the Silicon Slopes Tech Summit in 2018, with almost 15,000 in attendance, Pluralsight CEO Aaron Skonnard noted that "the state needs to double down on efforts to build a robust education pipeline to help address the 4,000 or so unfilled tech positions in Utah." He also said that failing to prepare students in public schools for these tech-centered jobs could interrupt Utah's growth in the tech sector. In addition, he said it is important to "build the tech employment pipeline" and "to provide our students with the things they need to occupy these jobs" (Raymond, 2018b, para. 12). Mr. Skonnard is not alone in his request. As the workforce services report described, "Paralleling this profile of strong IT job growth are accounts of companies struggling to find qualified tech workers — and they are not alone. With an unemployment rate below 4 percent since January 2014, many employers in Utah have reported challenges related to hiring. Yet, despite these difficulties, tech has continued to thrive, adding more than 2,000 jobs per year" (Utah Workforce Services et al., 2018, p.7). Many professionals, both in educational and corporate spheres are looking for ways to improve CS education and training. Given the brain gain of 1026, as calculated by Dr. Helen Hu, many of these new tech jobs (9900) may mean that they are being filled by non-Utah residents from

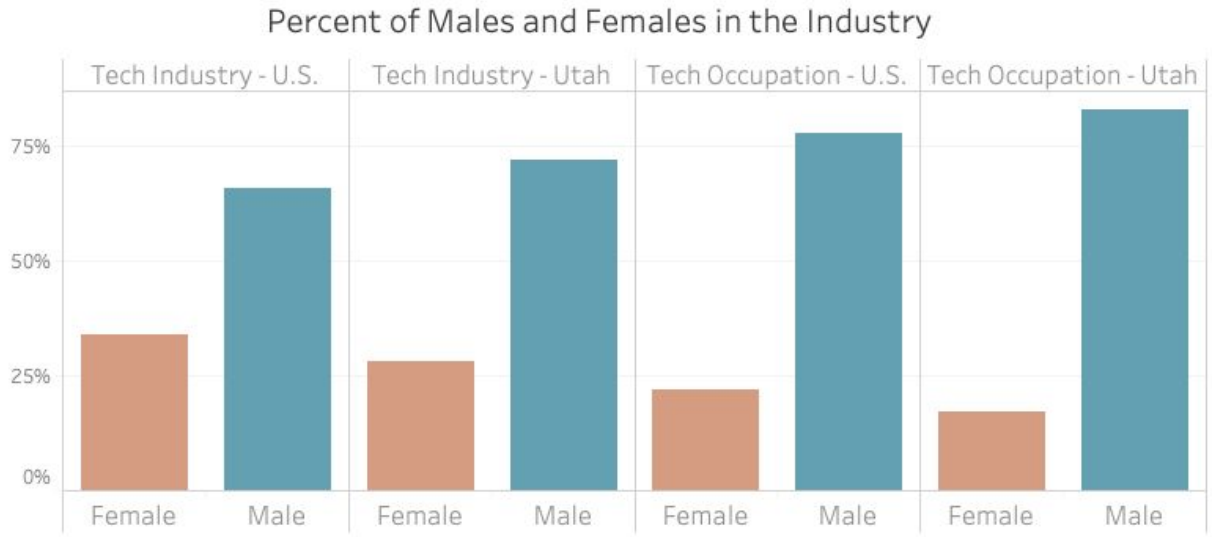
other states (Hu, 2018). In addition another report, the Utah Workforce services reported, "Net overall Utah IT labor migration is neutral; in- and out-migration largely offset. However, Utah IT labor in-migration carries in lower education levels (50 percent bachelor's or better) than what migrates out (75 percent bachelor's or better). When restricting the migration view to just coding workers, and then further to only STEM-educated coding workers, Utah becomes a net labor exporter" (Utah Workforce Services et al., 2018, p.3).

Regardless of the education level, Utah's Workforce Services stated, "Within Utah, the IT industry offers well-paying jobs. For those with education below a bachelor's degree, the IT industry offers rewarding career opportunities" (Utah Workforce Services et al., 2018, p.3). Encouraging students to participate in and learn about both CS and IT at all levels of schooling could lead to an above average salary and help fill these newly added tech jobs in Utah.

The Utah Workforce Services Tech Report identifies another one of Utah's deficiencies when it comes to Core IT employees. It states, "Nearly half of the Salt Lake/Provo tech workforce has been trained to only the certificate, vocational, and associate degree levels; levels that establish foundational coding skills. Other notable tech cities employ a labor force with predominantly STEM-focused bachelor degrees or higher" (Utah Workforce Services et al., 2018, p.3). This means that we need to encourage more people, especially the young, to take a good look at STEM higher education and its benefits to see if it's not right for them. In addition, the report stated, "Economic theory suggests that a market will operate with a labor-education level that best harmonizes with its business needs" (Utah Workforce Services et al., 2018, p.20).

11.4.1. Females in the Industry

There is an especially low number of females in the field. CompTIA's report found 2,362,00 (34%) female workers in the Tech Sector vs. 4,594,131 (66%) male workers (includes all females working for tech companies) in the United States (CompTIA, 2018, p.131). (CompTIA, 2018, p.131-132)



These numbers include females working in tech as an occupation and for tech companies in a non-tech occupation. To clarify, the report also specifies females in a tech occupation in any company or field (tech or not), shows lower percentages even further at 1,635,821 (22%) females workers and 5,804,872 (78%) male workers in the U.S. in tech occupations (CompTIA, 2018, p.132). Utah's ranked 26th for women working in the tech industry/sector (including in marketing, accounting, etc. in tech companies) were reported at 25,069 (28%) females and 64,768 (72%) males. Looking specifically at tech occupations in Utah, 14,133 (17%) females and 68,843 (83%), ranking 29th out of 51. In addition, CBRE Research reported that in Utah, they estimate that only 21% of tech workers are female (CBRE Research, 2017, p.A24).