A project of CSforCA

CS Equity Guide

A K-12 education leader's guide to designing, scaling, and sustaining equitable computer science in California

A project of CSforCA

CS Equity Guide

A K-12 education leader's guide to designing, scaling, and sustaining equitable computer science in California

September 2020

by district leaders, *for* district leaders

The CSforCA CS Equity Guide is designed for administrators interested in implementing equity-minded computer science (CS) in K-12 schools, districts, or counties. We hope this guide provides an important resource for school leaders to transform CS education implementation in California by increasing opportunities for all students, including low-income students, students of color, young women, English learners, and students with disabilities. If you are a principal, district or county leader, CS coordinator, lead teacher, teacher on special assignment (TOSA), counselor, family member, non-profit organizer, or just interested in bringing CS to all students, this guide is for you.

First and foremost, we would like to thank all the district leaders who shared with us their stories, successes, and challenges towards making CS education available to all in California. As part of a research effort to scale equitable implementation of CS, UCLA researchers interviewed leaders from early-adopter LEAs (local education agencies) across California to learn and share best practices. This guide is based on a compilation of their answers to the most frequently asked questions they receive from their colleagues.

In 2012, with generous support from the National Science Foundation (NSF), the **Alliance for California Computing Education for Students and Schools (ACCESS)** was formed to broaden participation in CS education in California, especially for girls, low-income students, and students of color. In 2016, ACCESS launched **CSforCA** as part of the national CSforAll movement, with the support of then-Lt. Governor Gavin Newsom, California State Superintendent Torlakson, and other high-ranking officials from the California Department of Education, Commission on Teacher Credentialing, district leaders, CS teachers, higher education faculty, industry leaders, parents, and non-profit organizations--all committed to equity in CS education. In addition, with CS stakeholders across the state, California's Governor, the State Superintendent of Public Instruction, the State Board of Education, and the Legislature developed a **CS Strategic Implementation Plan** to guide state and local efforts.

While CSforCA leverages the CS Strategic Implementation Plan to ensure all California high schools have the resources and capacity to offer CS, we also know it is up to school districts to implement CS education at the local level. In an effort to build capacity of educators, ACCESS was awarded a Research-Practice Partnership grant from the NSF to create a Network Improvement Community (NIC) to scale teacher professional development, build capacity of education leaders and policymakers, and contribute to the research base to support expansion of equity-minded CS education across California. This guide is a result of these efforts to inform interested education leaders statewide about how to bring equitable CS into their schools.

Since its inception, the mission of CSforCA has been to challenge and change inequitable educational structures that hinder students' access to quality CS learning experiences based on zip code, race, gender, or economic background. The coronavirus pandemic and the growing Black Lives Matter movement has brought greater attention towards issues of systemic racism and social justice. We hope this guide will aid you in using the lens of CS to reflect on inequities that are embedded in our education system and especially how the decisions you make in your role as an educational leader can address barriers, advocate for equitable access, and effect positive change in the movement towards high quality K-12 education for all.

As the landscape of California K-12 CS education continues to evolve, this guide will be updated to reflect best practices. We invite users of this guide to share their experiences with us for inclusion in future versions.

Contact us at **info@csforca.org**.

Julie Flapan, Ed.D.

Roxana Hadad, Ph.D.

3 Students and Recruitment

4 In the Classroom

5 Preparing and Supporting Teachers

1 Introduction

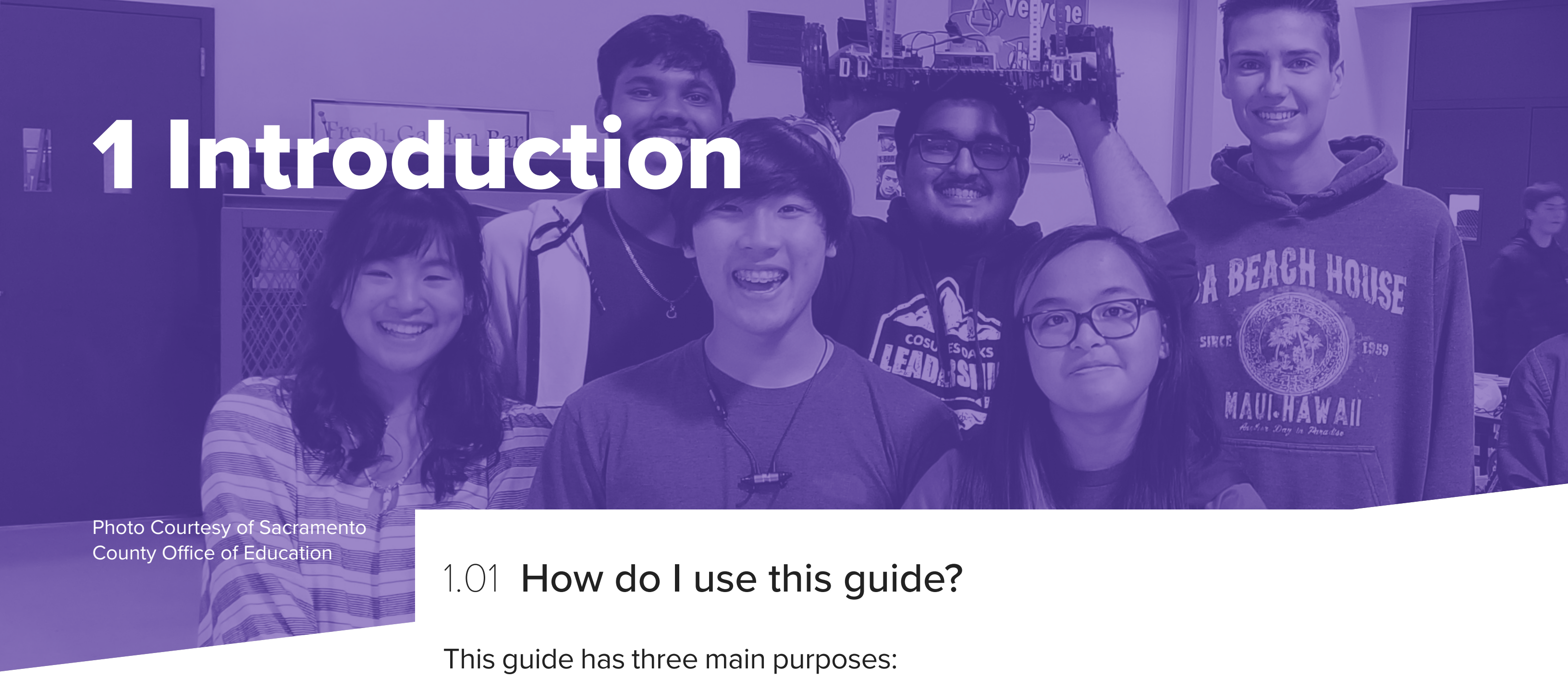


Photo Courtesy of Sacramento County Office of Education

1.01 How do I use this guide?

This guide has three main purposes:

1. To explore the challenges, opportunities, and consequences for implementing equity-minded K-12 computer science;
2. To demonstrate useful examples of schools and district leaders that have connected equity and computer science implementation; and
3. To offer suggestions for additional resources to confront inequity in computer science education.

With these goals in mind, this guide is more of a suggested starting point rather than an exhaustive resource of all information regarding K-12 computer science (CS) implementation.

What is equity?
Equity means that everyone gets the support they need to succeed based on where they are and where they want to go. This requires interrupting inequitable practices, examining biases, and creating inclusive environments for all, while discovering and cultivating the unique gifts, talents, and interests that every person possesses (National Equity Project, 2017).

As with everything in education, there is no one-size-fits-all for how to implement CS learning equitably. In these chapters, you will notice a tension between what is high quality implementation and what is most realistic for your local education agency (LEA). This guide provides resources, examples, and questions to consider when developing a CS implementation plan, but ultimately, it is you and your team who will best understand your LEA and its local context to effectively begin and grow CS education opportunities and access in your community. In addition to reading this guide, we encourage your team to take part in the CSforALL SCRIPT Program to develop a larger vision and benchmarks for implementation that can be phased in over time.

There is a large amount of content in this guide, so it is organized to be used flexibly according to your needs. You may choose to read it from start to finish, or use the Table of Contents to jump to a question you need answered or to select a topic you want to focus on first. We suggest that you then follow up to review the chapter titles and reflection questions posed within each chapter, which can serve as discussion topics and assist you and your LEA team in planning and decision-making.

"We facilitated administrator-teacher teams in creating the bones of their customized CS implementation plans. As part of that process, we gave teams a variety of resources to look at, one of which was the CS Equity Guide. We jigsawed different sections of the CS Equity Guide, as well as the CS Standards, the CS Implementation Plan, and the SCRIPT Rubric. Teams used these resources as fuel and inspiration to support their discussions about what their CS implementation plan would look like. We also used a home group/expert group protocol. Teams split up and sent members to 'expert' groups. Each group focused on a different reading from the CS Equity Guide. Once they became experts in that section, they returned to their 'home group' to teach the team about what they had learned, and continue collaborative discussions."

— Lauren Aranguren, Coordinator, Santa Barbara County Education Office

Regardless of how you decide to move through the guide, we recommend that you begin with sections 1.05, 1.06, and 1.07 to gain a better understanding of what we mean by "equity" and why it's important to support CS educational opportunities for all learners.

There are several linked resources throughout the guide. If you are reading the guide in print format, a list of all websites and links referenced in the guide is available in the appendix.

1.02 Who is this guide for?

This guide is mainly for administrators at the county, district, and school level, as well as teachers, counselors, family members, and non-profit organizers interested in advocating for equitable CS education.

1.03 Why computer science?

CS education is foundational learning for the future of California. CS teaches skills of problem-solving, critical thinking, creativity, and collaboration needed to succeed in the 21st century. In order to become active, productive, and creative members of modern society, young people need to learn how to be creators of technology, not just passive consumers. CS education helps prepare students for college, careers, and community engagement. Increased access to quality CS educational experiences will allow students to discover innovative solutions to problems in their communities, to explore new means of communication and creative expression, and to learn about college majors or careers they might never thought were possible.

1.04 What is computer science? How is it different from coding, educational technology, digital literacy, and computational thinking? Are they all the same thing?

According to the National Science Foundation, computer science is the study of computers and algorithmic processes and includes the study of computing principles and theories, computational thinking, computer hardware, software design, coding, analytics, and computer applications. CS often includes computer programming or coding as a tool to create software, including applications, games, websites, and tools to manage or manipulate data; or development and management of computer hardware and the other electronics related to sharing, securing, and using digital information. In addition to coding, the expanding field of CS emphasizes computational thinking and interdisciplinary problem-solving to equip students with the skills and abilities necessary to apply computation in our digital world.

- ***Coding*** (or programming) is using a language in order to carry out solutions. Coding is one tool in the CS toolbox, but CS teaches more than just how to code. CS is about solving problems using computers.
- ***Educational technology*** refers to the use of technology in the learning process. These are often the tools teachers use to deliver a lesson (edx.org).
- ***Digital literacy*** is the ability to use information and communication technologies (including word processors, spreadsheets, video editing, or presentation software) to find, evaluate, create, and communicate information (ala.org).

- ***Computational thinking*** refers to the thought processes involved in understanding problems and their solutions in such a way that the solutions can be effectively carried out by a computer (Wing, 2010). Computational thinking is a skill that is developed in CS, along with programming, communication, and creativity. Computational thinking is applied in all types of learning, such as math and science courses and it is even established as one of the eight practices in the Next Generation Science Standards (NGSS). You don't need a computer to use CT, but the use of CT can capitalize on the use of computers to solve a problem. Skills that are often listed as part of CT include:
    - Recognizing and defining computational problems (decomposition)
    - Developing and using abstractions
    - Creating computational artifacts (constructing algorithms)
    - Testing and refining computational artifacts (debugging & evaluation) (K–12 Computer Science Framework, 2016).

These competencies relate to critical thinking and problem-solving skills across subject matter, underscoring how the concepts of computing can be combined with other fields of study to assist in problem-solving.

How is computer science defined in the California Computer Science Standards?
"Computer science involves much more than use of computing systems. In order to provide universal access to computer science instruction for each and every student, we must thoroughly define computer science. Computer science can be defined as 'the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society' (Tucker et. al, 2006, p. 2)."
– **California K-12 Computer Science Standards (Introduction, p. 10)**

1.05 Why is it important to focus on equity in computer science?

There are current racial and gender disparities that limit students' full access to computer science education. CS has the potential to empower humans and have a positive impact on communities. Special attention must be paid to provide access to all students, especially those who have been historically underrepresented in CS, including (but not limited to) African Americans, Latinx, Native Americans, Pacific Islanders, women, English language learners, LGBTQIA, and students with disabilities. Not only does CS open the door to some of the highest paying and fastest growing jobs in America, but it can also help prepare students for college. Additionally, part of developing a well-informed citizenry is helping young people to understand and demystify computing technology and to be able to think critically about its potential and impacts.

California, home to Silicon Valley, is the 6th largest economy in the world and has one of the most diverse populations in the country. California is a "majority-minority" state of 6.2 million students who are over 60% Latinx, African American, and Native American (U.S. Census Bureau, 2017), but they make up only 24% of AP CS test takers.

According to a recent report by the Kapor Center, "Computer Science In California Schools: An Analysis Of Access, Enrollment And Equity":

- Just 3% of the 1.9 million high school students in California took a CS course in 2017
- Only 39% of high schools in California offer computer science courses
- Low-income schools are 4 times less likely to offer AP CS A than high-income schools
- 1 in 4 rural schools offer CS courses, compared with 39% of urban schools.
- Just 1% of AP CS test-takers in California are Black.
- 71% of students taking introductory CS courses are male, and 29% are female.

Expanding access to CS is important, but so is ensuring high quality engagement and increasing pass rates. Seven in 10 students who take AP CS exams receive passing scores, but only 4 in 10 Black and Latinx students receive passing scores (Scott, et al. 2019).

These inequitable outcomes are also reflected in the workforce. The computing industry does not reflect the diversity of our state (Harrison, 2019). When CS designs are created by a narrow subset of our population, those designs are not made with multiple perspectives, needs, or experiences in mind. In order to create robust and rich technology, it has to be designed and developed by teams that are as diverse as our communities are.

According to a recent poll by Edsource, 85% of CA voters believe schools should offer computer programming/coding classes, but only one in four schools offer computer programming/coding classes. Nationally, nine in ten parents want their children to learn CS in school, yet only one in four administrators believe parents are demanding it (Google and Gallup, 2016; EdSource-Berkeley IGS Survey, 2017).

The recent pandemic has put the inequities that exist in education and society at large in stark relief. CS education creates avenues for students from historically marginalized communities to make their voices heard and facilitates problem-solving in an increasingly complex digital world (CSforALL, 2020). Equitable access to the foundational learning CS provides will give every K-12 student the opportunity to thrive in the 21st century.

1.06 What is equity? How is equity different from equality?

Equity is accomplished when every student is provided with what they individually require to learn and succeed in fulfilling their personal, academic, and social advancement, and when success and achievement is not predicted by any demographic factor. This requires continually interrupting inequitable practices, examining biases, and creating inclusive environments for all, while at the same time discovering and cultivating the unique gifts, talents, and interests that every student possesses.

Equitable practices are based in the belief that every child's educational experience should be rigorous and relevant, and that everyone is capable of learning. These beliefs are supported when we provide a learning environment that is safe and respects every student.

While often used interchangeably, equality and equity are not the same. Equality suggests that all people should simply have access to the same resources, regardless of need. With equity, resources are distributed according to different students' needs, while taking into account how certain students have been systematically denied access to educational resources, opportunities, and experiences based on race/ethnicity, gender, sexual orientation, socioeconomic class, and disability.

An equity-based approach means acknowledging and challenging: 1) the institutional barriers impacting youth differently based on the way they look or where they come from, 2) countering practices rooted in stereotypes about who can participate or who should excel, and 3) recognizing that people both present themselves and are treated differently in different contexts depending on how their various identities overlap and intersect. This requires an ongoing and cyclical approach to examining factors impacting youth's experiences.

CS and CS education have been documented as being highly segregated along race/ethnicity, gender, and socioeconomic lines due to a lack of access to high-quality computer science learning opportunities for all students (Margolis, Estrella, Goode, Jellison-Holme, & Nao, 2017). However, an increased awareness of equity issues in the CS education community presents a special opportunity to structure learning experiences and environments with equity considered throughout the progression from grades K through 12, as many CS frameworks, policies, and courses are being built or have yet to be built. Not only is CS an emerging field of study that leads to high-wage and high-demand careers that can address socio-economic inequality, it can empower students to be critical users of technology and creators in all fields touched by technology while expressing themselves in our digital world. By using equity-based approaches to build your K-12 CS program from the ground up, you are making progress against persistent inequities, while at the same time providing young people pathways for career and college success that will ultimately lead to a more just, equitable, and productive society.

1.07 What does equity in computer science education mean?

Historically, students have been denied access to CS due to a variety of factors: counselors, teachers, and administrators enrolling students in computing classes based on stereotypes about who they thought could/should excel with computing, lack of professional development for educators who were isolated in their schools, and/or education policies and structures that did not allow all students to participate (Margolis, Estrella, Goode, Jellison-Holme, & Nao, 2017).

In order to be equitable, schools and teachers should distribute resources and opportunities according to individual needs, taking into account how certain students have been systematically denied access to educational resources, opportunities, and experiences based on race/ethnicity, gender, sexual orientation, socioeconomic class, and disability.

What is preparatory privilege?
As documented in *Stuck in the Shallow End*, students who are considered 'talented' at computer science often come from homes where there are multiple computing devices and the parents are familiar with CS. They enter the computing environment at school with a considerable advantage (Margolis, Estrella, Goode, Jellison-Holme, & Nao, 2017).

While making CS classes available to all populations is a major first step, it's also important that all students are able to feel included, valued, and engaged in the classroom. Thus, intentional, equity-minded implementation must be present in multiple levels of the educational system, taking into account the structure of the school, belief systems, pedagogy, and policy impacting student access to quality CS education (Goode, Flapan, & Margolis, 2018). All students must receive education that is based on research-informed best instructional practices, including (1) rigorous, engaging, and culturally responsive content knowledge, (2) a belief that with appropriate support, all students are capable of learning and succeeding, and (3) an empowering learning environment that welcomes, incorporates, and respects the identities, cultural assets, and cognitive skills that each student brings into the classroom, so that the contributions of each student are valued and CS has meaning for students' lives (Ladson-Billings, 1995; Sleeter, 2012).

CSforCA has a data tool to get a clearer picture of how computer science opportunities are distributed in each community in California. You can look at your district or county and see how it measures up in regards to gender and racial/ethnic diversity in CS classes.

1.08 Who should lead a computer science initiative?

To set up your CS program for success, it is important to designate a district/school/county individual or team who will be responsible for leading the implementation. In certain LEAs, this team may consist of networks across schools or districts that can help build and sustain a CS initiative. This person or team will lead the district's or county's work on CS, coordinate professional learning experiences, and be your point of contact with the state to troubleshoot implementation challenges and to network with other districts across the state. In some LEAs, the staff in these positions may also work with schools to add CS to the master calendar, depending on their level of authority. Consider in what department these positions would be located, which relationships they will need to maintain, and where they will need to advocate for CS education. They should not be the same as the person who supports technology integration or personalized learning, as these are entirely different roles that would take away the focus from CS specifically.

It's useful to look at sample job descriptions for a district or county lead in CS, like these from San Francisco Unified School District and Chicago Public Schools to know what skills and requirements to look for in a good CS education leader.

Some necessary qualifications:

- Knowledge of CS
- Experience as an effective teacher and/or instructional leader
- Ability to motivate and inspire principals
- Ability to effectively develop and leverage relationships
- Ability to motivate and inspire other key decision makers within a district
- Ability to motivate and inspire students
- Knowledge of other related programs like career and technical education (CTE), Perkins, and CTE incentive grants (CTEIG)

"After reading Stuck in the Shallow End, I went from having to work with schools to implement CS because I had been asked to ... to then thinking 'I really have to do something about this -- this is seriously important.' Knowing about research that supports equitable CS implementation gave me the buy-in, and has been really instrumental in doing this work."
— **Sonal Patel, Digital Learning Innovation Coordinator, San Bernardino County Superintendent of Schools**

Photo Courtesy of Riverside Unified School District

2 Program Design and Development

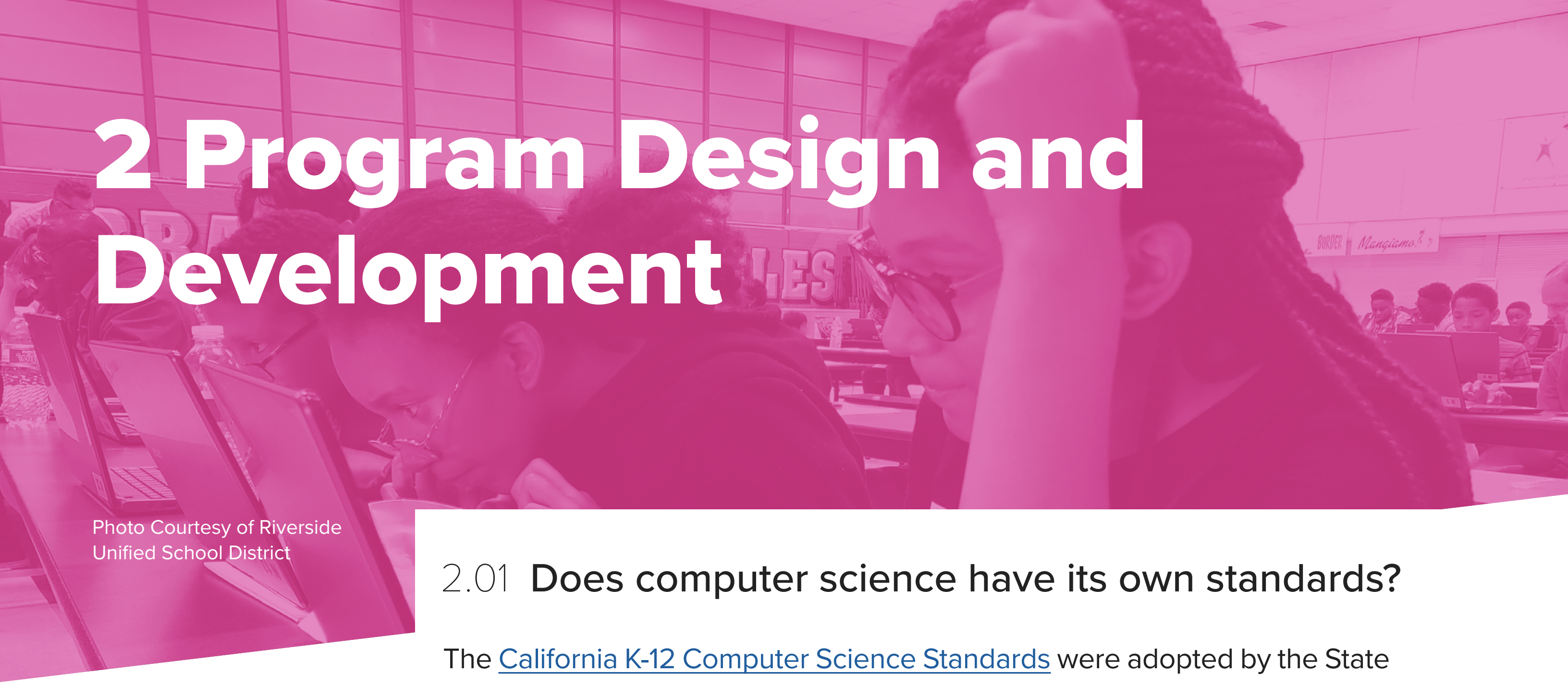


Photo Courtesy of Riverside Unified School District

2.01 Does computer science have its own standards?

The [California K-12 Computer Science Standards] were adopted by the State Board of Education in Fall 2018. These K-12 CS standards are designed to be accessible to each and every student in California. The standards are meant to inform teachers, curriculum developers, and educational leaders to ensure all students receive quality CS instruction. Each standard includes a descriptive statement as well as examples for classroom application. More information about the standards can be found in **4.01 What are students supposed to know and be able to do in a computer science class?**.

2.02 How can I add computer science into the master schedule?

When creating your master schedule, CS should be made available to all students. This inclusion may look different at different levels, but universal exposure at each grade band is important. Setting time aside for CS daily would be ideal, but this may be difficult given other course schedules and requirements. In elementary and middle school, CS is often offered as a "special" alongside physical education, art, and music. Consider placing CS in a compulsory exposure wheel, so that all students can experience CS along with the arts, health, foreign languages, etc., as seen in San Francisco Unified

"There are schools with courses that are nine weeks, while others are a full year. We've created pacing guides for those variations of schedules, but it comes down to having a flexible curriculum. We've found that when there is any larger spacing than once a week, it does not contribute to the consistency for students to learn important CS skills. When students have CS at least once a week, then they can see connections and foster computational thinking skills and CS content knowledge to work through a multi-day project. One way schools made that happen is by developing a semester schedule, where CS is offered to grades K-2 in the fall and then grades 3-5 in the spring. It benefits the teacher in terms of planning and has benefits for students by providing more consistent CS experiences."
— Bill Marsland, Computer Science Content Specialist, San Francisco Unified School District

School District's middle grades model. Be mindful of schedule overlaps and students who could be unintentionally excluded, for example, English learners, students with special needs, students in orchestra, etc. It will be important to communicate and advocate the importance of CS to all school staff. Consider how schedules might work dependent on whether CS staff is itinerant or site-based. See more in **2.05 What is an appropriate course sequence for computer science?**.

2.03 Can computer science be integrated with other disciplines?

There are many opportunities to integrate CS with other courses and resources. By supporting a diverse set of teachers to be certified (e.g. history, art, physics, etc.), integrated coursework may develop organically. Find and empower educator advocates in your districts who are excited about finding creative ways to incorporate CS. True integration requires extensive planning, so setting aside time for teacher collaboration, support, and curriculum planning is essential. CS can be integrated into existing curricula for elementary, middle, and high school. For example, students can collect and analyze data in science class, program interactive stories in language arts, or create visualizations in math class. Integration can also be a beginning for an LEA, serving as a bridge for CS to ultimately exist as a stand-alone class.

2.04 What curricula and professional development are recommended?

While we do not endorse specific curricula, we suggest considering providers who share the following basic principles: 1) commitment to equity in their curriculum and professional learning experiences, 2) alignment with the California K-12 Computer Science Standards (and other content standards through the integration of CS standards), and 3) demonstrated history of success providing curriculum and professional learning in California that is aligned with California's Quality Professional Learning Standards (QPLS).

"We organized a CSforALL network of administrators, teachers, and counselors in our county who are interested in growing CS in our districts. At one of our quarterly meetings, we did some Venn diagramming on the standards for CS, the Standards for Mathematical Practice, the ELA Portraits, and the NGSS Science and Engineering practices, and found a lot of overlap that could be considered for developing lesson plans that integrate CS."

– Matt Zuchowicz, Director, Educational Technology Services, Santa Barbara County Education Office

An integrated course must address standards of at least two subject areas, not just one. In addition to specific CS standards, there are CS principles embedded in the Next Generation Science Standards (NGSS); specifically, "Using Mathematics and Computational Thinking" is listed as one of the eight Science and Engineering Practices. CS standards are also aligned with CTE/ICT standards. The University of California Curriculum Integration (UCCI) provides examples of integrated courses that meet both CTE and General Education requirements. The appendix to the California K-12 CS standards has alignment maps to all state standards. The CS Content Standards database lists examples for CS activities that align with CS standards and with the standards from other subject areas. For more ideas, see **2.05 What is an appropriate course sequence for computer science?**.

A curriculum should be creative, collaborative, and inclusive. At every level, students should find CS opportunities that are inviting to them. Having courses that are only programming or game development and do not prioritize collaborative work can alienate students who may not see themselves as programmers or game developers. Consider courses that highlight the aspects of computing that are creative and altruistic in order to encourage participation from students with diverse interests

In addition, be sure to consider if the curriculum is equity-minded and what professional development and teacher support is provided. Curricula that is culturally responsive and/or adopts ideas similar to that of Universal Design for Learning (UDL) or Specially Designed Academic Instruction in English (SDAIE) (Cline & Necochea, 2003), in which the curricula is designed to meet the needs of special populations can help everyone learn. The University of Florida's Creative Technology Research Lab has a resource to help align UDL guidelines with CS. See more in **5.09 How can teachers best support students with disabilities in computer science classes?**.

Finally, keep in mind that you do not merely need a curriculum package, but you also need teachers prepared to teach it. Consideration should include research-backed opportunities that incorporate active learning, support collaboration, use models of effective practice, provide coaching and expert guidance, offer feedback and reflection, and is of sustained duration (Linda Darling-Hammond & Gardner, 2017). For more information on teacher preparation and professional learning, see **Chapter 5 Preparing and Supporting Teachers.**

The following are some popular K-12 CS curricula. These are by no means the only choices. There are lots of excellent providers and programs. You can search the Computer Science Teachers Association (CSTA) and CSforALL databases to explore lesson plans and professional development offerings.

- **K-2nd grades: ScratchJr Curricula** - ScratchJr is an introductory programming language available as a free app for both iPad and Android tablets. These curricula are intended to introduce critical thinking and problem-solving ideas, as well as equip students with skills like programming, self-expression, and user-centered design.

- **3rd-5th grades: Creative Computing** - This multi-unit curriculum introduces young people to the creative nature of programming using a free online programming software called Scratch. Students learn to express themselves through creating projects and sharing their creations with others. For more detailed lessons for grades 3-5, see CSinSF's adaptation of Creative Computing.

- **6th-8th grades: CS Discoveries** (CSD) - CSD is an introductory CS course that empowers students to create authentic artifacts and engage with CS as a medium for creativity, communication, problem-solving, and fun. CSD is designed from the ground up to be an accessible and engaging course for all students, regardless of background or prior experience. It provides students opportunities to engage with culturally and personally relevant topics in a wide variety of contexts and aims to show all students that CS is for them.

- **8th-9th grades: Bootstrap Algebra** - These curricular modules reinforce core concepts from algebra. Their introductory class can be integrated into a standalone CS or mainstream math class, and aligns with national and state math standards. They also have physics and data science modules.

- **9th-10th grades: Exploring Computer Science** (ECS) - A year-long, project-based, inquiry and equity minded curriculum that grew out of a partnership with LAUSD. ECS consists of 6 units, approximately 6 weeks each, developed around a framework of both CS content and accompanying professional development for teachers. Instruction and assignments are contextualized to be socially relevant and meaningful for diverse students utilizing a variety of tools/platforms, and culminate with final projects around the following topics: Human Computer Interaction, problem-solving, Web Design, Programming, Computing and Data Analysis and Robotics. **This course should be made available to the entire school (especially to groups traditionally underrepresented in computing).**

- **High School: Introduction to Data Science** (IDS) - The IDS curriculum teaches students to reason with, and think critically about, data in all forms. The Common Core State Standards (CCSS) for High School Statistics and Probability relevant to data science are taught along with the data demands of good citizenship in the 21st century. IDS is a "C"-approved mathematics course in the University of California A-G requirements. As a statistics course, completion of IDS validates Algebra II in California.

- **10th-11th grades: AP® CS Principles** (AP CSP) - This year-long course models what many Universities teach in their non-majors CS introductory course. It focuses on the "foundations of CS with a focus on how computing powers the world. Along with the fundamentals of computing, students will learn to analyze data, create technology that has a practical impact, and gain a broader understanding of how CS impacts people and society." **Despite its labeling as an AP® course, this course should be made available to the entire school (especially to groups traditionally underrepresented in computing).**

- **11th-12th grades: AP® CS A** (AP CSA) - **This course is the strongest predictor of majoring in CS in college.** AP CSA models what many universities teach in their first course for CS majors. It is taught in Java and introduces "object-oriented programming methodology with an emphasis on problem-solving and algorithm development. It also includes the study of data structures and abstraction." CSAwesome is a College Board endorsed AP CS A curriculum and professional development to teach Java programming.

2.05 What is an appropriate course sequence for computer science?

There is no "one-size-fits-all" CS course sequence; pathways can range from introductory exposure to more advanced classes, as demonstrated in the appendices of the California K-12 CS Standards. The goal is to provide multiple opportunities for all students to experience CS.

Elementary school can introduce computational thinking concepts, digital literacy and citizenship, and programming with blocks. Text-based programming is appropriate for middle school. Physical computing is relevant for all ages, and the same devices and platforms can be revisited multiple times throughout K-12 so students can progressively build understandings and competency over time. In high school, strategic course planning and scheduling should be done so that students in both college and career pathways can have meaningful CS learning experiences.

Consider implications for equity when deciding what is compulsory and what is elective, keeping in mind who actually has access to CS if an elective is the only opportunity to take it. For example, students whose elective time is already required by other courses (English learners, students with disabilities, etc.) would be denied access to CS if offered only as an elective, and other students who do not see CS as a part of their identity or interests will most likely choose not to take it. Compulsory CS courses should adequately prepare students to choose and engage in electives. Consider integrating an introductory CS unit into a larger non-elective course, or combining two courses in one, as in Bootstrap Algebra. In addition, enrollment and assessment of electives should be carefully monitored to avoid "preparatory privilege" (see definition in **1.06 What is equity? How is equity different from equality?**). For more ideas, see **2.02 How can I add computer science into the master schedule?**.

There are also various ways to integrate CS with other subject areas, initiatives, and resources at all levels. CS-related subject areas that might interest students include digital media, art and design, robotics, game development, app design, video production, and music. And even storytelling projects in history, writing, and language arts classes can provide creative pathways for students to encounter CS who might not have before. See section **2.03 Can computer science be integrated in other disciplines?** for more ideas.

You can find a sample K-12 CS course pathway in the K-12 Computer Science Framework and see scope and sequence examples including district-wide implementation models in **9.01 What computer science education initiatives are currently taking place in California?**.

Equity considerations:
Algebra I, as with other subjects, is a gatekeeper for secondary CS. We recommend selecting an introductory curriculum that allows for algebra as a co-requisite. For more information about math requirements and their effect on CS enrollment, refer to Just Equations, an organization devoted to ensuring math policies give all students the quantitative foundation they need to succeed.

"We encourage schools to design CS pathways that work for them now and plan for change, knowing that the demand for CS courses will increase as students gain more exposure to the standards in the lower grades, and teacher capacity will grow."
— **Emily Thomforde, Computer Science Coordinator, San Mateo County Office of Education**

2.06 What course code should computer science courses have?

Identify computer science course pathways that will ensure equitable access for all students. This includes creating pathways for diverse educators to qualify for teaching CS. How you choose a course code in CS impacts who is authorized to teach it. For example, a course coded as a Career Technical Education (CTE) course often requires a teacher with a CTE and/or Information and Communication Technologies (ICT) authorization, whereas a CS course coded in general education/non-CTE can be taught by a teacher with a single subject credential in math, business, or Instructional Technology Education (ITE). Alternatively, a teacher with multiple subject or different single subject credentials can obtain the CS supplementary authorization to teach a general education course. There are several CS courses dual coded in CTE and general education. Check with the California Department of Education (CDE) for the latest course codes. See more in **5.02 Who is authorized to teach computer science in California?**.

Coding a course as a CTE course allows access to additional Perkins funding, but this can also restrict the required teaching credential. Districts are encouraged to establish a dual course code that can be used for teachers with either credentials so that students on both college and career pathways have exposure to CS, regardless of the coding of the course. Be aware that the CS education landscape in California is evolving and credentials may change, so check with the California Commission on Teacher Credentialing.

The CSforCA website has a helpful infographic to clarify what qualifications allow one to teach which classes.

2.07 How do I know which courses are approved for A-G college and career readiness in California?

There are several courses that are already approved by UCOP for adoption at the local level. Check the UCOP A-G Course Management Portal to see if your curriculum is already approved for A-G. Getting new courses approved can be a challenge, so selecting pre-approved courses can simplify the process.

2.08 What does computer science "count" for in high schools in California? Math? Science? Elective? CTE?

At the high school level, consider CS courses that meet the rigor of University of California (UC)/ California State University (CSU) A-G college preparatory requirements to incentivize students to add it to their schedules. A CS course could potentially count as a math, science, CTE pathway course, and/or an elective, depending on the specific course and UC approval as well as your local requirements. Many CS courses in California are designated as a "G" elective and many CS courses are part of CTE pathways, while some are both. CTE pathways consist of a multi-year sequence of courses that integrates academic understanding with technical and occupational knowledge as a way to provide students with a pathway to post-secondary education and careers. Many schools choose to classify CS courses in the CTE pathway in order to claim Perkins Funding to support a teacher with CTE/ICT authorization.

To be University of California (UC) eligible, students are required to take one college prep elective. In this case, CS "counts" as a "G" elective and therefore, can be considered part of the "core" requirements for UC eligibility. UC also recognizes an approved CS course to count toward the advanced math requirement ("C" credit) or science ("D" credit). For some students, getting a CS course to count as an advanced math or science provides further incentive for them to take it. There are some advantages to maintaining CS as an elective. A "G" elective reinforces the notion that CS is for everyone, regardless of math or science interest and aptitude. Conversely, one disadvantage of maintaining CS as an elective is English Learners often don't have space in their schedules for an elective because their schedules are full with EL coursework.

Check what courses count in your region to help your LEA make strategic and equitable decisions. Review the A-G Course Management Portal and speak with your human resources department to determine what courses can be offered and what corresponding teacher certifications are required. Updated course designations are available on the ACCESS website.

2.09 There is so much to be done to add a new course in middle or high school. How can I simplify the process?

To start, we recommend that each school in your district start by doing the following in advance of the upcoming school year. You can anticipate your CS program to evolve as teachers and students gain interest and capacity over time, and keep in mind you will want to iterate as new challenges and opportunities emerge. See more tips for getting started plus example district implementations in **Chapter 9 More Information**.

- **Decide which course(s)** are appropriate for your school, based on the potential teaching staff and student interest. Some courses cost money (CodeHS, PLTW) while other courses are available for free (Exploring Computer Science, AP CSP).

- **Identify teachers** from your school who may be interested in teaching CS. Common sources include math, science, creative arts, CTE technology teachers, or anyone else interested in bringing computing to the school. Any educator can be successful in teaching introductory CS with sufficient professional learning opportunities. See more in **Chapter 5 Preparing and Supporting Teachers**.

- **Sign your teacher(s) up for summer PD** for these courses. Regardless of the curriculum or teaching background, it is essential that CS teachers participate in professional development to feel supported and prepared. Check the Computer Science Teachers Association (CSTA) website for the latest opportunities. Sometimes the professional development is free, but you may want to review your budget when taking into consideration the requirements your district has for compensating teachers going to non-contractual professional development.

- **Incorporate CS into master scheduling.**

- **Recruit students to take courses, especially students from traditionally underrepresented groups**, like African Americans, Latinx, Native Americans, Pacific Islanders, women, English learners, LGBTQIA students, and students with disabilities. There may be students who don't realize they might enjoy CS or students who love technology that counselors may not know about, so recruitment may need extra attention. Learn more in **Chapter 3 Students and Recruitment**.

- **Assure the classroom** that will be used for computing classes has a layout that allows for collaboration, lab work, and working computers or laptops. Connect with your IT team to install and test any software that will be used. Find out more about infrastructure requirements in section **4.07 What computing devices (hardware), software programs, and broadband (internet) connections are necessary?**.

- **Determine the district approval process**, including steps like getting an approval on the school board agenda.

For more information on how to begin your CS program, see **9.04 This is a lot of information. Where do I start?**.

"Both content and pedagogical knowledge are critical. Strong teachers can transfer their skills used to teach other disciplines. To support teachers with lesser content knowledge, you can connect with an organization like TEALS to see if there is potential to have a computing professional co-teach your courses with your teachers. They maintain the pedagogical flow of their class, while getting support in their classroom and with their own development."
— Bryan Twarek, Computer Science Supervisor, San Francisco Unified School District

2.10 Can I start computer science at one school or does it need to be district wide?

There are pros and cons of introducing a new discipline like computer science on a school-by-school basis or as a district-wide initiative. Some administrators pilot their CS program at one school as a way to start small and experiment, exploring what works and scaling from there. Starting small allows them to build capacity among a small group of "CS champions" who go on to empower fellow teachers by sharing their successes with CS. However, it is difficult to generalize findings from one pilot experience, so you will need to recognize individual school's unique culture and needs as you expand.

One benefit of implementing a district-wide initiative is that it ensures that all schools have access without any site being singled out. It also allows more schools to generate evidence for what is working and what is not. However, district-wide implementation can be daunting and overwhelming, especially if there are unexpected challenges or limited resources. Sometimes district-wide initiatives can be met with resistance if there isn't "buy-in" from staff who don't see evidence of the value of CS.

It is important to form a key leadership group of teachers, counselors, and administrators who will collectively decide what implementation strategy works best for your LEA. The CSforAll SCRIPT Program can help your leadership group identify your vision for CS implementation and formulate a strategy for meeting your goals.

2.11 What "level" should I start with (elementary, middle school, high school)?

It may be easiest to start at the level you have the most momentum for CS. However, if universal experience is a goal, earlier grades may be easier to implement CS than high school, where adding a new course is generally more challenging, unless integrating CS into an existing course.

When making decisions about where to start, look to where you have excited principals and teachers, buy-in from staff, the most access and capacity for professional development opportunities, and/or teachers that are already certified, and go from there.

The goal is to eventually have a coherent pathway from Kindergarten through 12th grade. Coherence will develop through the use of standards, curriculum, and coordination among elementary, middle, and high school teachers. Become familiar with the district processes and culture in order to be able to advocate for CS and have a say in district-level decisions. The CSforALL SCRIPT Program can help you determine the best path forward in keeping aligned with your LEA's culture, values, and goals.

"We started with an elementary afterschool program. A single afterschool program was obviously not meeting our equity goals, so soon after, we began training large groups of teachers throughout our district."
— Steve Kong, Coordinator, Riverside Unified School District

2.12 What about data science?

Data science is a subset of CS that involves collecting, analyzing, visualizing, and communicating with data. Data science is currently accepted as a "C" math in A-G. The Introduction to Data Science course offered by the IDS project at UCLA is a great high school resource and could be considered as part of a district's CS pathway.

2.13 What about cybersecurity?

Cybersecurity is the practice of protecting devices, data, systems, networks, and programs from malicious digital attacks. There is an increased demand for those with an understanding of cybersecurity and safe technology practices as cyber attacks continue to grow and as digital devices become more and more ubiquitous.

Cybersecurity is a subconcept in California's K-12 Computer Science Standards and schools can implement a cybersecurity education component via a CTE pathway through industry sectors such as ICT and Public Services. The National Institute for Standards and Technology (NIST), which is the home of the National Institute for Cybersecurity Education (NICE), has a list of online resources for learning about cyber security. The Cyber Guild and Cal Poly SLO's CA Cybersecurity Institute are other resources for expanding cybersecurity training. The Department of Defense and CSforALL are also piloting a program to expand computer science and cybersecurity education to JROTC schools. Schools in California serving Junior Reserve Officers Training Corps (JROTC) youth may be eligible to participate.

3 Students and Recruitment

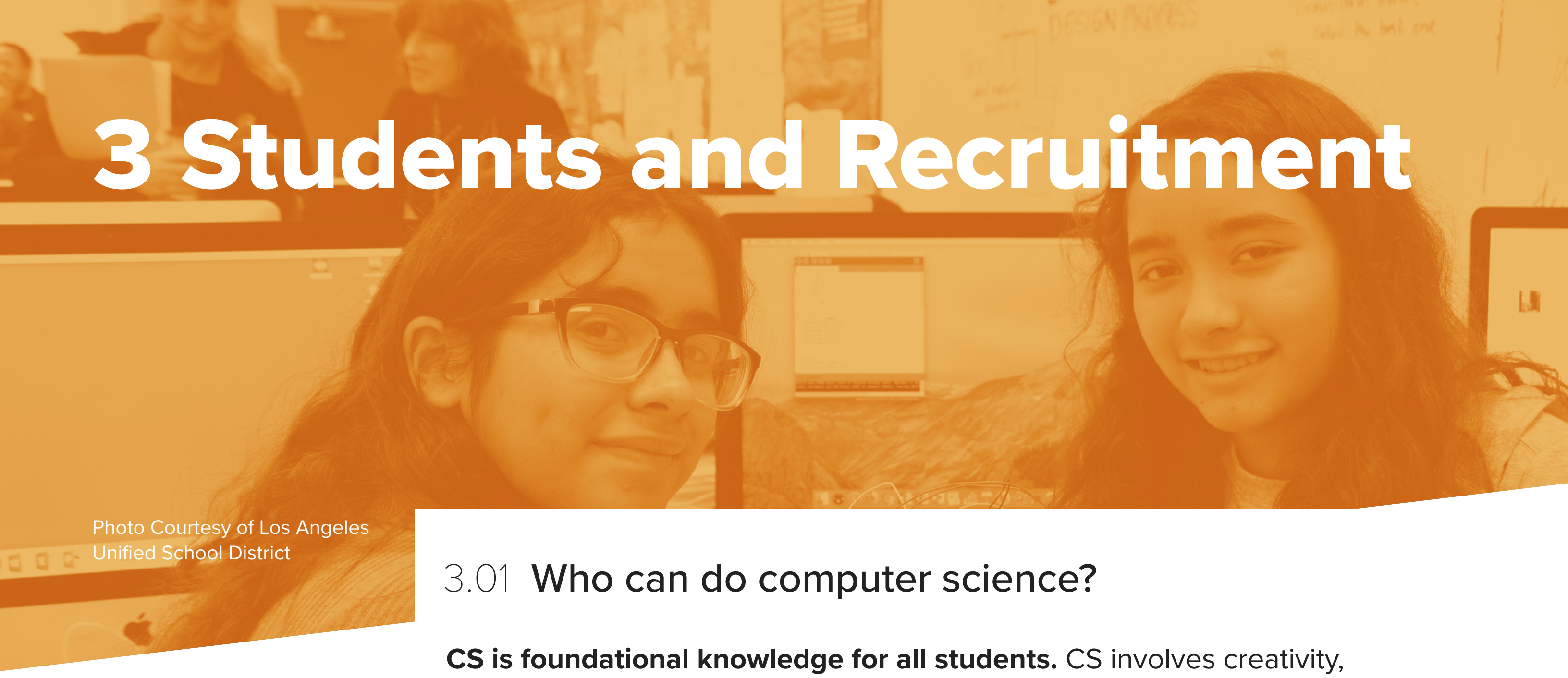


Photo Courtesy of Los Angeles Unified School District

3.01 Who can do computer science?

CS is foundational knowledge for all students. CS involves creativity, problem-solving, logic, and communication skills, and it is a driver of innovation across all fields, from the sciences to the arts. Be sure to encourage all students, especially those who have been traditionally underrepresented in the field, such as females, African Americans, Latinx, Native Americans, Pacific Islanders, women, English learners, LGBTQIA students, and students with disabilities, to enroll in CS courses. A variety of skills, experience, and backgrounds are not only useful for learning CS, but fostering computer scientists and those working with technology who are of diverse perspectives and backgrounds can lead to more creative solutions that meet the needs of more people. Because technology has become integral in everything from government, to healthcare, to our justice system, it is important to empower and be intentional about including all students in CS education. Read more about why it's important to support equitable CS in sections **1.03 Why computer science?, 1.05 Why is it important to focus on equity in computer science?, and 1.07 What does equity in computer science mean?**.

> "Principle 1: Every student and every teacher is capable of learning computer science. Access to and achievement in computer science should not be predicated on the basis of race, ethnicity, gender identity, socioeconomic status, language, religion, sexual orientation, cultural affiliation, learning differences, or special needs."
> — **California Computer Science Strategic Implementation Plan (p. 6)**

3.02 How will students know to sign up for computer science? How can we recruit them?

School counselors, educators, administrators, families, and currently enrolled CS students all play pivotal roles in student recruitment and retention.

Two of the most effective ways to get underrepresented students into classes are:

1. **Individual advocacy** (teachers and counselors individually reach out to underrepresented students); and

2. **Recruiting in groups**, and having students encourage other friends to enroll, so they are not isolated or the token representative.

Counselors can be powerful advocates and may benefit from learning more about the value and purpose of equitable CS education in order to be better informed about why and how to encourage more youth to enroll in CS courses. Counselors may hold assumptions based on a student's race/ethnicity, gender, sexual orientation, socioeconomic class, or disability, or they may not know about a student's personal experiences or interests with technology and computing without delving deeper. These misunderstandings can inadvertently lead to overlooking or denying youth certain CS opportunities. We have found that students often do not know to share their interests or experiences unless explicitly asked. Resources and professional development opportunities are available to school counselors through NCWIT's Counselors for Computing.

We suggest counselors make efforts to ensure all students experience CS at least once along their middle/high school pathways to either connect to prior interests or spark new ones. This creates a reference point for counselors to ask about and advise students on future CS enrollment. At schools like Mendez High in Los Angeles, Exploring Computer Science has become a 9th grade course requirement, ensuring that every student has an introductory experience and understanding of what CS is before moving on in their high school pathways.

Teachers and administrators also play a crucial role in recruiting underrepresented students. By implementing courses that are engaging, challenging, and provide beneficial outcomes for students (e.g. interesting and relevant projects, connections with the community, AP credit, job opportunities, etc.) they will learn about CS courses from their peers and want to take part.

Give currently enrolled CS students ways to share their projects and excitement for CS with their peers. School-wide fairs can showcase the creative work of young women and students of color who can serve as role models or mentors for other curious students. Participating in CS pathway classes with friends can also support student retainment.

Families are also powerful influences on youth and their course selections. When informed about the positive educational and career impacts CS can have, parents and caretakers can help motivate students to take CS courses. For more information on involving families, see **7.01 How can I get support from families and the local community for computer science education?**.

The following are other useful resources for recruitment:

- CS Recruitment Toolkit - Microsoft
- Recruitment Toolkit - College Board
- Tips for HS Student Recruitment – CSTeachingTips

Photo Courtesy of Los Angeles Unified School District

4 In the Classroom

Photo Courtesy of Riverside Unified School District

4.01 What are students supposed to know and be able to do in a computer science class?

The California K-12 Computer Science standards prepare students for college, career, and community engagement. The standards include five core concept areas of which CS is comprised, along with seven core practices that demonstrate ways in which students actively engage in CS learning experiences in order to develop conceptual knowledge.

The computer science **core concepts** include:

- Computing Systems
- Networks and the Internet
- Data and Analysis
- Algorithms and Programming
- Impacts of Computing

The computer science **core practices** include:

- Fostering an Inclusive Computing Culture
- Collaborating Around Computing
- Recognizing and Defining Computational Problems
- Developing and Using Abstractions
- Creating Computational Artifacts
- Testing and Refining Computational Artifacts
- Communicating About Computing

For more information and examples of what these concepts look like in the classroom, please visit the CDE website K-12 Computer Science standards. Other helpful resources that unpack standards in computer science include the Computer Science Teachers Association (CSTA), the K-12 Computer Science Framework, and the International Society for Technology in Education (ISTE).

4.02 How should a computer science class be taught?

While computers, programming, or coding can play a significant role in CS, they are not the entirety of the subject area. Be mindful of students' screen time, and make sure to take time to explore concepts through "unplugged" activities, encouraging rich discussion on concepts like data, networking, and the ethical implications of the structure of certain computing platforms, in order to foster the development of critical consumers and creators of technology. Students should be able to express their learning and explore the different facets of computational thinking through the creation of artifacts. Exploratory, hands-on project development allows educators to focus on students' processes rather than the end product. Collaboration and communication are integral to CS, so include group projects and teamwork throughout computing courses along with open-ended individual activities that connect to students' interests and passions. Educators should maintain a beginner's mindset to model problem-solving, finding multiple solutions, and debugging as core practices in programming. See more about developing pathways and curricula resources in **Chapter 2 Program Design and Development**.

4.03 How can I engage diverse learners in computer science?

The best teaching practices from other disciplines can apply to CS classrooms, as well. For example, a teacher with pedagogical expertise in facilitating collaborative work can employ those same strategies to CS lesson plans. Classes should be student-centered, differentiated, asset-based, and culturally responsive. A list of instructional strategies from the Code.org CS Discoveries curriculum guide (starting on p. 4) may be helpful. See more resources for supporting teachers and the needs of diverse learners in **Chapter 5 Preparing and Supporting Teachers**.

Microsoft and Microsoft Philanthropies TEALS (Technology Education and Literacy in Schools) partnered with the National Center for Women & Information Technology (NCWIT) to produce the Guide to Inclusive Computer Science Education, which provides educators with context and concrete steps to build and expand inclusivity in CS education.

4.04 Why do we need to examine biases? What does that have to do with computer science?

It is important for administrators, program developers, educators, and school counselors to examine their own biases and challenge stereotypes about "who looks like a computer scientist." Research documents how we unconsciously hold different expectations for students based on our upbringing, identity, or ways our schools are organized (e.g., Delamont, 1996; Tenenbaum & Ruck, 2007).

This can exacerbate inequities instead of ameliorate them. MIT has a helpful site with interactive case studies for understanding and addressing biases in science, technology, engineering, and math (STEM), as well as resources for specific strategies. We encourage taking advantage of professional learning resources and workshops that support inclusive educational models such as growth mindset and asset-based approaches, challenge preheld biases, and explore ways to engage learners in a culturally responsive curriculum. CSforCA's workshop for administrators includes activities that encourage self-reflection, interactive dialogue, and analysis of how perceptions of students can influence the way we teach in order to expand access and inclusion to computer science (Goode, 2008). By joining CSforCA, you can learn about upcoming workshops. See more in **1.05 Why is it important to focus on equity in computer science?** and **3.01 Who can do computer science?**.

Culturally Responsive Teaching

Making a course culturally responsive makes the content inviting and inclusive. Culturally responsive educators:

- Have high academic expectations
- Offer students appropriate support (e.g., scaffolding, tutoring) as determined through formative assessment
- Shape curriculum to value and build on the experiences, knowledge, and cultures students bring to the classroom
- Establish relationships with students and their home communities
- Examine biases and cultivate students' awareness of existing dominant power structures in society (Ladson-Billings, 1995; Sleeter, 2012)

Learn more at the Center for Culturally Responsive Teaching and Learning and Culturally Responsive Teaching and the Brain.

Other equity-minded professional learning resources include:

- **Teaching Tolerance**
- **Anti-Defamation League Anti-Bias Resources**
- **Rethinking Schools**

4.05 What does assessment for a computer science class look like?

Portfolio-based and performance-task assignments allow for embedded and authentic evaluation, offering opportunities for students to be motivated by assessment. The CS Principles Performance Tasks are a good example. The Exploring Computer Science (ECS) curriculum also has assessments built in through the Principled Assessment of Computational Thinking.

CS assessments should include more than just coding or programming tasks. Instead, they should include problem-solving debugging challenges, formative feedback on drafts, and reflections on process. Introductory CS courses or interdisciplinary classes geared towards digital literacy can be designed with the purpose of inciting a curiosity for computing and expanding notions of who engages with CS. Harvard's Creative Computing Lab offers key principles in their Assessing Creativity in Computing Classrooms to guide the assessment of creative programming activities.

Equity considerations:
Formative assessment refers to the evaluation of student understanding and learning needs that are integrated into a lesson. These assessments involve collecting data on student learning that can be used to improve instruction while it is happening.

Extensive formative assessment can help guide the course facilitation to meet students "where they are". Using assessments that avoid a single correct answer to a problem allows for different perspectives and understandings.
– **National Research Council (2014)**

As CS is a relatively new and evolving K-12 discipline, standards-based exams are still being developed. Allow educators to take the lead in evaluating student learning and adjust coursework accordingly. Utilize best practices from other disciplines -- CS courses can use similar ways of monitoring growth and achievement during instruction. While CS is not formally part of the California School Dashboard, it could potentially impact other indicators, as described in **6.02 How can we best utilize state-funded Local Control Accountability Plans (LCAP) and Local Control Funding Formula (LCFF)?**.

4.06 How do I know if my computer science plan is working?

Develop a program evaluation plan to measure your success and identify areas for growth. Collaborate with existing research teams and teacher input to determine data collection methods and tools. Begin by tracking progress on access (where are classes offered?), participation (who takes CS?), achievement (what are grades and pass rates on AP exams?), and retention (do students continue to enroll in CS courses?). Once those are established, other forms of evidence may include: performance/ assessment measures, instructional quality, teacher retention, student engagement, and feedback from educators, parents, and students. Whatever program you develop, it should include plans for continual monitoring and be available for stakeholders to view and refine.

The CSforCA data tool can help you find out who has access to CS education in your school, district, county, or zip code. This tool aggregates publicly available data from the California Department of Education and other sources to report who has access to and enrolls in CS by gender, race/ethnicity, and income. While this tool doesn't measure growth over time, it can provide helpful comparison data for defining benchmarks and goals for your program. For more data and information, contact the data department in your district.

"Be sure to leverage the curricular frameworks you have in place and find entry points where you can leverage new tools or instruction. This is not about devices. This is about the thinking and the strategy."
— **Sophia Mendoza, Director, Instructional Technology Initiative, Los Angeles Unified School District**

4.07 What computing devices (hardware), software programs, and broadband (internet) connections are necessary?

Hardware and software considerations will rely on both your budget and needs. Introductory level courses can be delivered on Chromebooks and other basic devices while some of the higher-level courses (e.g., AP CS A and some CTE courses) may require greater additional computing power. Many resources are browser-based, so be sure to have sufficient broadband at school sites. Keyboards are helpful for upper grades, while touchscreens and tablets can be preferable for younger students or students with certain disabilities. A hand-held mouse is often important for younger students, too, as some can struggle with the dexterity required of trackpads. In addition, many lessons on problem-solving using computing are better served "unplugged", needing no equipment at all.

It is recommended to use a regular classroom with a modular seating arrangement, since a traditional lab environment may hinder collaboration. Pair programming is a strategy for students to collaborate on projects, so it's helpful to have large table space for group work or physical computing projects.

Beyond costs for the physical machinery, it is highly recommended that an LEA includes funding for tech support so there is someone on site to troubleshoot technical issues as they arise. As the tech support responds to machinery and internet access, the educator can focus on teaching CS classes, as well as planning, prepping, and implementing curriculum. You may also want to build in a budget for system updates and maintenance.

Consider coordinating these decisions with your district's technology department and your county office's technology unit. For example, determine the ratio of how many students to machines, functionality of technology hardware and software, ensuring ongoing IT support, and whether or not to lease or buy machines.

"Unplugged' activities are a great supplement for 'plugged' work, but are a terrible substitute for learning to write code, which requires execution by a computational agent. Use unplugged activities to illustrate concepts and provide foundational computer science and computational thinking experiences. Students without devices are better served by a focus on the Data and Analysis and Impacts of Computing standards, leaving Algorithms and Programming and Computing Systems to subsequent years."
— Emily Thomforde, Computer Science Coordinator, San Mateo County Office of Education

For more information on what considerations to make when developing a district computer system, see Digital Promise's Digital Learning Playbook and/or California IT in Education.

Equity considerations:
Consider the device and internet needs of students at home as well as what is provided from school. For example, one device per family may not be enough when multiple family members or siblings require a computer to complete work and/or school obligations, and the appropriate amount of "screentime" can be different in different homes. For distance learning, coordinate across the school or district as well as with parents and caregivers to assess comprehensive computing, bearing in mind those who may not have internet access, technical support, or multiple digital devices at home. Broadband and connectivity are essential elements for providing internet access to CS students. Not all students have access to the internet from their homes and some rural communities struggle with reliable internet connections. For more information about the inequities in broadband access and resources for support to help close the digital divide in California:

- **California's Digital Divide (PPIC)**
- **The Digital Divide in Southern California**
- **California Emerging Technology Fund**

4.08 What about distance learning?

In general, it could be considered easier to put a CS class online than any other class, since a lot of the creation and communication that happen in CS are already computer-based. However, keep in mind the challenges distance learning poses to equitable implementation:

- Instructors need professional development on how to best deliver content online.
- Students need access to devices and internet service to access online content (see **4.07 What computing devices (hardware), software programs, and broadband (internet) connections are necessary?**).
- Engaged learning is dependent on the human connection between the instructor and the student, as they adjust the learning in response to informal formative assessments, and make the lesson more relevant to the student based on what they know about them. This connection is more difficult to make when the interaction is not in person.

For more information, see New York City CS4All's Resources for Remote Computer Science Instruction.

Photo Courtesy of San Francisco Unified School District

5 Preparing and Supporting Teachers

Photo Courtesy of Santa Barbara County Office of Education

5.01 Who should I hire to teach computer science?

There is no typical résumé for a CS teacher. While having a background in CS is useful, be sure to consider non-traditional candidates with past experience in technical or creative fields, like math or art. Hiring managers should prioritize pedagogical knowledge over CS content knowledge, or someone currently on staff could be the right person, which removes the additional challenge of onboarding a new hire. A strong teacher of any subject can become a successful CS teacher with the right mindset and support for ongoing professional learning. If hiring a school-based instructor is impossible, consider hiring an educator for the district to teach computing across multiple sites. Although there are statewide certification requirements (see section **5.02 Who is authorized to teach CS in California?**) for who can teach CS, there is also local authority to make exceptions. Permits for these exemptions are for one year and are determined locally. If the exempted instructor needs more time to earn a credential, they will need to demonstrate that they are making progress. Check with your human resources department to explore what's available. Once you have a program in place, establish a pipeline and professional development, including fast-track programs from local colleges.

Equity considerations:
Teachers of color can provide unique connections to students who would benefit from seeing themselves represented in a CS classroom. CSforCA supports efforts to recruit, build capacity, retain, and sustain teachers of color in CS education. Since the overall teacher workforce doesn't reflect California's diverse student body, we need to make a concerted effort to recruit, train, and retain teachers of color in CS education. The Edtrust-West has a report documenting the lack of teachers of color and provides important information on why hiring teachers of color matters and how they can best be supported.

"There are most likely some teachers that are teaching CS in some capacity in your district. Be sure to recognize these early-adopters and give them pats on the back."
– Sophia Mendoza, Director, Instructional Technology Initiative, Los Angeles Unified School District

5.02 Who is authorized to teach computer science in California?

In California, teachers are authorized to teach CS with any one of the following:

- single-subject credential in mathematics (or in foundational-level mathematics for grades K-9) (More information on p. 30 of the Administrator's Assignment Manual)
- single-subject credential in business
- single-subject credential in industrial and technology education (ITE)
- Career Technical Education (CTE) credential in Information Communication Technologies (ICT) or Arts, Media, and Entertainment (AME)
- any single-subject or multiple-subjects credential AND
    - supplemental authorization in computer concepts and applications (CCA)
    - supplemental authorization in CS
    - supplemental authorization in mathematics (or in foundational mathematics for grades K-9) (More information on p. 30 of the Administrator's Assignment Manual)

When CS is part of a CTE pathway, teachers with a CTE credential in ICT (Information Communication Technology) or Arts Media and Entertainment (AME) are eligible to teach CS, depending on the course. There is a CS supplementary authorization available for teachers who hold other multiple or single subject credentials. UC Irvine, UC Riverside, SFSU, and UCSD are some of the institutions that offer a supplementary authorization program. Teachers with other credentials may be eligible to teach CS with a temporary waiver or emergency permit. Teachers can also get a math supplementary authorization (CSETS + methods course). Additional information is available at California's Commission on Teaching Credentialing website. Regardless of teaching authorization, it is strongly suggested that ALL teachers participate in CS professional development opportunities as well as have professional learning resources to learn what equity in CS means in your program.

"A teacher can get industry hours for a CTE credential, but these don't necessarily have to be hours working at a tech company. For example, teachers can use IT lead hours at a school site towards industry hours."
— Claire Shorall, Manager, Oakland Unified School District

It is important to consider what if any expertise is required for the classes you are offering. For example, CS-A and other advanced CS courses at the secondary level may require familiarity with teaching technical programming languages such as Python or Java. Other introductory level courses, such as CS Principles, Exploring Computer Science, or CS Discoveries are written with the assumption that a teacher has little to no experience with computer science. Professional learning opportunities are crucial to providing some of the necessary support to teach these introductory classes.

5.03 How can teachers learn to teach computer science?

Select professional development providers that will offer foundational learning as well as ongoing support for your teachers. It is important to identify teachers for professional development who are authorized to teach CS. Microsoft and NCWIT developed a useful CS professional development guide focusing on equitable CS education. See more in **2.04 What curricula and professional development are recommended?** and **4.03 How can I engage diverse learners in computer science?**.

"It's important to plan for necessary resources to support teachers participating in ongoing professional learning. Working with your County Office of Education and/or professional learning provider, you may need to budget for substitute and travel costs for workshops offered on weekdays, travel and staff time associated with trainings offered on weekends, as well as materials and staff time for virtual professional learning offered outside of contractual time."
— Jared Amalong, Computer Science Coordinator, Sacramento County Office of Education

5.04 How can an industry professional become a computer science teacher?

The Commission on Teacher Credentialing provides information about pathways an industry person with IT experience can take to start teaching. To teach CS in a CTE pathway, a candidate would need a Designated Subjects ICT credential (cl888) and a yearlong authorization to teach CTE (cl895). The Designated Subjects credential requires a high school diploma and can be obtained by anyone who has the appropriate industry experience through a sponsoring LEA or school district. Approximately 50% of the CTE profession hold this credential. See more in **7.02 How can I involve local industry?** and **7.03 How can I get industry professionals and volunteers in the classroom?**.

5.05 I have a teacher who is passionate about bringing computer science to more students. How can their enthusiasm and interest best be supported?

Educators who are excited about bringing CS to more schools and students can play a significant role in supporting equitable CS education. They can teach CS classes, volunteer in computing clubs, help with program development, and serve as important allies for recruitment and advocacy. There are many ways to foster an enthusiastic teacher into a CS leader for your district. Some districts have found success in having a Teacher on Special Assignment (TOSA) be responsible for helping expand professional learning opportunities. In addition to working in the central office, they also can become a professional development facilitator of their curriculum of choice and support other schools in the district.

5.06 How can I get teachers, counselors, and administrators excited about and engaged in professional development sessions?

To keep teachers, counselors, and administrators excited and engaged, it's important for them to feel they are part of a larger movement to challenge systemic inequities and provide equitable CS learning opportunities, within their school, throughout the state, and across the country. Professional development experiences like Summer of CS take a multi-stakeholder and systems-based approach to equitable CS implementation to help various school staff see they are in this effort together with others in their district, across all of California, and beyond.

It is important for teachers to feel supported among a strong community with shared values in improving their practice through professional learning. Be sure to design a continuum of professional development opportunities for a variety of teacher skills and ability, from novice to expert. Offer participatory workshops so that educators can get hands-on experiences with the programs they are expected to teach, and allow teachers opportunities to provide feedback on what professional learning needs or problems of practice they want support for. Design the professional development opportunities so that administrators do not have to do a lot of the logistical work; they just need to provide a list of teachers. The CSforCA coalition invites teachers to join its community of practice and additional networks are described in **9.03 What other resources should I take a look at?** See more about engaging educators, counselors, and administrators in **3.02 How will students know to sign up for computer science? How can we recruit them?** and **5.08 What kind of training do administrators need?**.

"We had really high interest in our professional development offerings because we not only broadcast them on our district-wide learning management system, but because teachers would leave our trainings with the curriculum and materials they needed. They could hit the ground running the next day."
— Sophia Mendoza, Director, Instructional Technology Initiative, Los Angeles Unified School District

"It's so important that the district leaders have an understanding of the professional learning that the teachers are participating in, so that they too can understand its value."
— Karen Larson, Computer Science Coordinator, Santa Clara County Office of Education

5.07 What's next after teachers go through their initial professional development?

Teachers are at their best when they feel supported in the classroom and are part of an ongoing professional learning community with other teachers. Many times CS educators feel isolated as the only person covering that content at their site. In-classroom coaching is a useful strategy to support teacher development in real time in the classroom (Margolis, Ryoo, & Goode, 2017). In addition to coaching, professional learning communities (PLCs) can provide support through collaboration across sites. A PLC can allow teachers to troubleshoot common challenges, as well as share best practices. They should find a PLC that meets regularly (once a month or two), either in-person, virtually, or even through asynchronous means (e.g., online forums). These PLCs should be structured in a way that is participatory in nature, where teachers learn from one another, instead of having a facilitator that leads the discussion. The discussions at the meetings should balance what could be immediately applicable for teachers (e.g., activities, programs) with more abstract development (e.g., equity considerations).

Be sure that your teachers are compensated for their time and structure their school day to facilitate their involvement.

There are ample opportunities for teachers to participate in professional learning communities, both online and in-person, including:

- **The Computer Science Teachers Association (CSTA)** is a membership organization that supports and promotes the teaching of CS. Their membership consists of more than 25,000 members from more than 145 countries and includes elementary, middle, and high school teachers; college and university faculty; industry and government; school administrators; non-profits; and parents. With regional chapters, grant-writing support, regular publications, and conferences, CSTA offers many avenues for teachers to connect and develop their CS teaching skills.
- **CS for All Teachers** is a virtual community of practice, welcoming all teachers from Pre-K through high school who are interested in teaching computer science.
- **CSTeachingTips** is a collection of CS teaching tips to help teachers anticipate students' difficulties and build upon students' strengths. Teachers contribute and promote tips on this user-friendly website.
- **ScratchEd** is an online professional learning community for educators using Scratch. Educators can share stories, exchange resources, ask questions, and find people and events on the website.

"Administrators should understand that CS can't be learned in one day of PD, and that teachers need to be able to learn along with their students."
— Emily Thomforde, Computer Science Coordinator, San Mateo County Office of Education

5.08 What kind of training do administrators need?

Principals don't need the same level of detail that teachers do, but they do need an understanding of why equitable CS education is important. CSforCA and the UCLA CS Equity Project developed a workshop for administrators to accompany this guide and better understand the foundations of CS, dispel misconceptions, and demonstrate the versatility and power of CS. For more information, join CSforCA and see sections **1.05 Why is it important to focus on equity in computer science?**, **3.01 Who can do computer science?**, and **4.04 Why do we need to examine biases? What does that have to do with computer science?**.

The Strategic CSforALL Resource and Implementation Planning Tool (SCRIPT) guides teams of district administrators, school leaders, and educators through a series of collaborative visioning, self-assessment, and goal-setting exercises to create or expand upon a CS education implementation plan. Check the SCRIPT website to search for workshops or trained facilitators in your area.

5.09 How can teachers best support students with disabilities in computer science classes?

Preliminary research has shown that not only can students with disabilities learn CS, but that CS can serve as a means by which they can thrive in the classroom (Gribble, et al., 2017; Hansen, et al., 2017).

Project TACTIC at the University of Illinois has various resources to support students with disabilities. They recommend:

- Teach using the Universal Design for Learning (UDL) framework so that students have access to flexible instructional delivery, materials, and assessment;
- Balance explicit instruction to teach fundamental computing concepts alongside opportunities for open inquiry so that students can engage in creative problem-solving; and
- Facilitate collaborative problem-solving so that students can learn how to co-construct understanding, support each other, and share their expertise with their peers.

The Alliance for Access to Computing Careers (AccessComputing) at the University of Washington has resources for students with disabilities and their instructors. AccessComputing helps students with disabilities successfully pursue undergraduate and graduate degrees in computing fields, and works to increase the capacity of postsecondary institutions and other organizations to fully include students with disabilities in computing courses and programs.

5.10 How can teachers best support students who are English Language Learners?

Consult with your English Language Learner department, as they are invaluable resources for supporting English Language learner students. Keep in mind that the master schedule is a huge component to whether or not English Language Learners have equitable access to CS because they are required to take English language development courses and will have less room in their schedules for electives. You can learn more about supporting English Language Learner students in STEM subjects in the report by the National Academies of Science, Engineering, and Medicine.

PiLa-CS, a National Science Foundation-funded project based out of New York City that supports emergent bilingual students, provides example units and strategies. They utilize Scratch, which is available in more than 50 languages and facilitates

Before beginning to plan courses and pathways for CS, first review the landscape and history of CS education in your district. A baseline of what is currently happening and an overview of your district or county will help your team determine where there is the most need. Ask questions like:

- Which courses are being offered?
- Where are the courses being offered (and where not)?
- Who takes the courses (and who does not)?
- Who succeeds in the courses (and what is contributes to the challenges others are facing?)
- What funding for CS education is in place?
- Who is teaching CS? How long have they been teaching CS and what professional development have they completed?
- How are community partners involved?
- What has already been tried?
- What were the challenges faced?
- How can you overcome them?

See an Example District Overview Tool that you can use for planning purposes.

participation in creative computing for students with diverse language backgrounds.

5.11 How do we support paraeducators/ paraprofessionals in computer science classrooms?

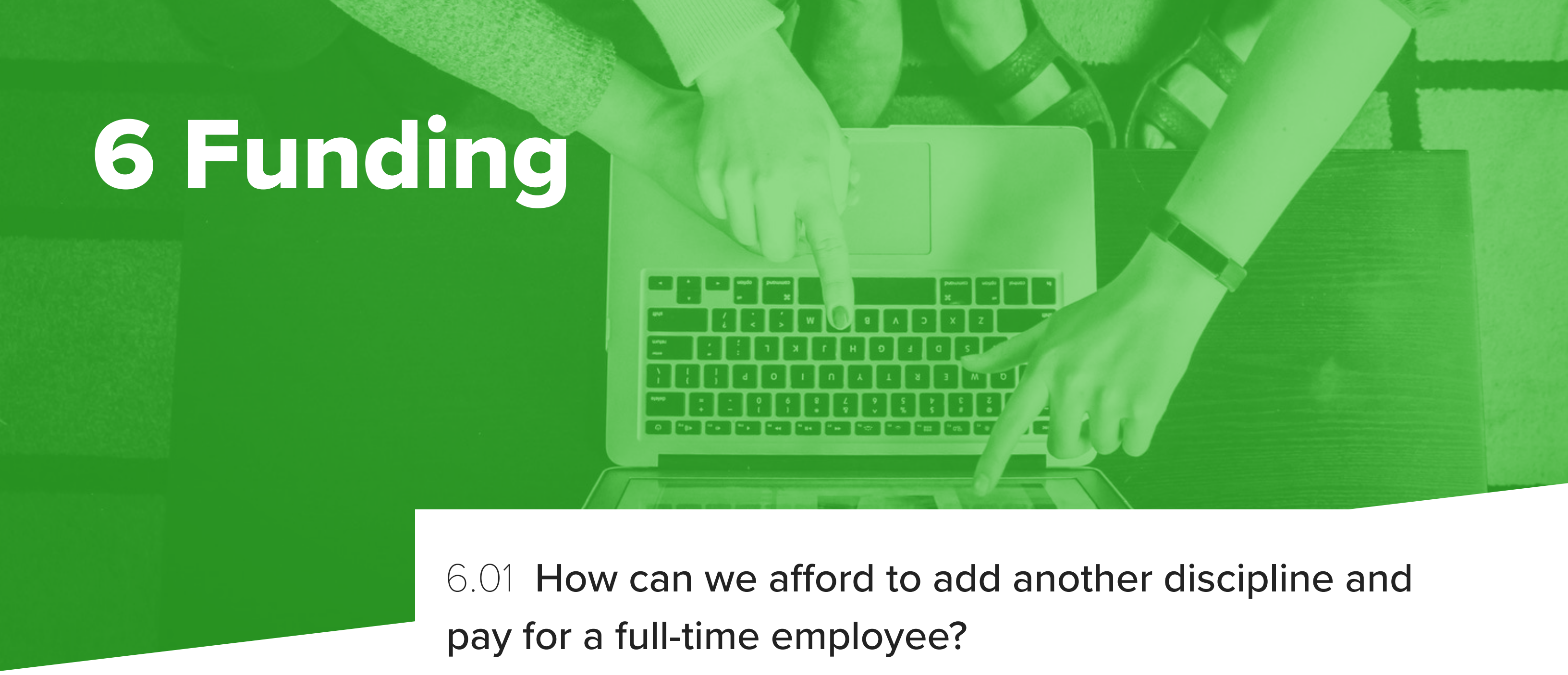
Paraeducators, as well as mainstream-supporting credentialed teachers in the classroom, will need professional development and ongoing support. For more information, see the TACTICal teaching brief on paraeducators that lists common challenges and strategies for K-12 CS paraeducators.

5.12 How do I evaluate computer science teachers?

Look for effective, engaging, and equitable teaching practices, similar to those you might find indifferent subject areas. Certain practices that are more typical to programming classes that you may notice include students working on different projects, using online resources and references, asking peers how to do things, and teachers not having all the answers. This is all expected.

"At SFUSD we had a teaching rubric to help us evaluate the quality of our instruction, letting us see where we were strong, and where our teaching needed more support. Coaches (teachers on special assignment) supported teachers in critical reflection and ongoing professional growth."
— **Bryan Twarek, Computer Science Supervisor, San Francisco Unified School District**

Photo Courtesy of Riverside Unified School District



6 Funding

6.01 How can we afford to add another discipline and pay for a full-time employee?

A primary source of funding for CS is through a school or a district's general fund and/or other statewide allocations. You may be able to leverage existing funding sources, such as grants in STEM, career pathways, CTE/Perkins, professional learning, etc., and direct them toward CS education. Other times, local PTAs or other parent organizations can help access funds, though parent engagement strategies will look different in different communities. Asking local industry to support is another way to garner additional funds. Programs like Microsoft's TEALS program are another option for bringing additional content expertise in the classroom to facilitate alongside a CS teacher. Learn more in sections **7.02 How can I involve local industry?** and **7.03 How can I get industry professionals and volunteers in the classroom?**.

"Students who were in the alternative education independent study school I taught at always showed up to my CS classes. Many of the students who attended this school were there because they had previous attendance issues. CS kept them coming back."
— Ed Campos, Consultant, Kings County Office of Education

6.02 How can we best utilize state-funded Local Control Accountability Plans (LCAP) and Local Control Funding Formula (LCFF)?

CS education can be added as a college and career readiness indicator on Local Control Accountability Plans (LCAPs) under Priority 7: Course Access. Districts can create four-year implementation and evaluation plans to help all students achieve the K-12 CS standards. In addition, you may have already set aside funds for professional development in your LCAP. These funds can be found in several priority areas depending on the plan. Local Control Funding Formula (LCFF) includes targeted funding for schools with low income, English

Learners, and foster youth. This funding is specifically designed for schools to implement new and improved instructional services for their most struggling students. Many districts already implementing CS have been allocating LCFF funding for this purpose (California Department of Education, 2015).

"When the Los Angeles Board of Education unanimously passed a resolution to ensure CS education and digital tools for all students, it signaled to all of us that the families and communities of LAUSD were invested in the access and equity of CS education."
— Sophia Mendoza, Director, Instructional Technology Initiative, Los Angeles Unified School District

In order to make sure CS gets included in future funding plans, it is strategic to determine and discuss CS's potential to meet other district goals. There are preliminary studies that point to positive effects from CS course participation on student math scores and attendance. Local industry representatives may be interested in speaking to a superintendent in order to highlight the need for CS professionals to meet the growing demand.

In California, there is funding available from the Career Technical Education Incentive Grant Program (CTEIG) and K12 Strong Workforce Program (SWP) for the purpose of strengthening industry transfer pathways particularly in the sector of Information Community Technologies. See Workforce & Economic Development and contact your regional K12 SWP Pathway Coordinator for more.

Consider using E-Rate funding for CS education infrastructure, like networking equipment for computer labs. See California IT in Education for more.

Another opportunity is grant funding and partnerships with local universities. For example, the US Department of Education and the National Science Foundation offer funding opportunities, especially prioritizing special populations and cross-sector or research-practice partnerships.

Photo Courtesy of Santa Clara County Office of Education

7 Family, Community, and Industry

Photo Courtesy of Riverside Unified School District

7.01 How can I get support from families and the local community?

Families recognize the value of a quality computer science education. Nine in ten parents say they want their children to learn CS or believe that schools should offer CS classes. But only one in four school principals say their schools offer computing programming/coding classes.

Engage families and community members in recruitment. Let families and community members know about CS course offerings and highlight the importance of all students accessing CS. Encourage families to share success stories of how their children were positively influenced by CS learning. Family members are trusted messengers and can encourage other families and community members to support and expand CS education.

Consider organizing a Family Code Night to give parents/families/caretakers and potential CS students a fun, hands-on CS experience. Approach local technology companies for support and volunteers. Other resources include:

- EDC's CS toolkit for families provides information on the need for incorporating CS education both in and out of the school day.
- STEM Family has ideas for how families can incorporate CS concepts into household activities.
- CS4All NYC has developed Resources for Remote Computer Science Instruction for Families that are translated and include unplugged activities.
- Common Sense Media reviews the various coding apps and games to help students learn to code.

Families and community members are advocates for CS. Equipping family and community members with data to demonstrate the lack of access and opportunities is an effective way to communicate the need for more CS education. CSforCA has developed a data tool for users to search schools by name, district, county, and zip code to reveal opportunity gaps. Code.org has resources and data that can be included with sample letters to school leaders and policymakers to advocate for equity in CS education.

Your local schools board's support and commitment to CS is also an important strategy for expanding access to CS. School board support and school site councils are important for allocating resources in the Local Control Accountability Plans (learn more in **Chapter 6 Funding**). The California School Boards Association details promising practices to expand access to STEAM courses and experiences including CS. The California State PTA School Smarts Parent Engagement Program helps parents improve understanding of the K–12 school system and what they can do to get involved and advocate to support their children.

"Building community partnerships and community-facing events is just a huge win. We've hosted events and afterschool activities that are tied in with community stakeholders. They don't always understand CS or what it looks like, but they do understand the importance of giving students access to it."
— Steve Kong, Coordinator, Riverside Unified School District

7.02 How can I involve local industry?

Many local businesses recognize the value of CS education and are willing to provide internships to students. Work with local technology employers to communicate the importance of equitable CS opportunities and develop an appropriate program for students of various skill levels. Aside from internships, there are less involved opportunities to support work-based learning, like workplace tours, mentorships, guest speakers, or job shadowing, as explained in this toolkit from SFUSD's Career Pathways department.

"We approached local technology companies with a robotics camp we wanted to sponsor for our students with free and reduced lunch. One organization let us use their technology center for free so students could experience working in the space for a week. Another provided laptops for each student to use during the program and at the end of the week, they were able to take the laptops home. So coordinating with these different community partners, we were able to get a venue and laptops for the kids. On our end, we managed transportation, signups, and logistics. Then, word gets around in the community, because these community partners are sharing that story with others, who in turn approach you and say, 'Hey, we heard you did this thing, can we help you do something similar?'"
— Steve Kong, Coordinator, Riverside Unified School District

If your district has one, speak to the Linked Learning department in order to leverage existing connections. In addition, you can inquire with the human resources department of your local businesses, contact your local Chamber of Commerce (many have STEM-focused programs for this purpose) or reach out to intermediary organizations who can help build partnerships such as Families in Schools.

7.03 How can I get industry professionals and volunteers in the classroom?

TEALS and EnCorps are programs that support schools by pairing CS classroom teachers with full-time employees who have CS expertise. These professionals are not to replace the full-time teacher, but rather, to serve in partnership with them. Volunteers can share technical expertise and insight from the workplace, while full-time teachers can provide pedagogical expertise and classroom management. Invite past alumni, recent graduates, or students from local colleges and universities majoring in CS to volunteer or present projects. Check if college students can receive course or internship credit for working as teaching assistants.

Make an effort to match volunteers to the demographics/backgrounds/language abilities of students. Students are often inspired to hear from and see CS professionals they can identify with. For more, see CSTeachingTips for volunteers.

Other programs include K12 SWP Pathway Coordinators that facilitate industry connections with K–14 career technical education programs. See the K12 Strong Workforce Program for more information. Regional Workforce Intermediaries, like the Central Valley's Valley Vision, coordinate and align students and teachers with employers, civic partners, and industry. This is often accomplished through regional advisory committee meetings. Another regional entity that may help with connecting professionals to classrooms is your local Workforce Development Board. Finally, your CSTA chapter may also have useful connections that already have experience working with local schools.

Schools that are isolated in rural communities may have greater challenges connecting with volunteers and local industry. Working with other rural districts in partnership could help develop a critical mass of support for students. Consider exploring online options so that students can have a video view of technology in action. Resources like Nepris can help connect classrooms with industry professionals. See the Small School Districts Association for additional ideas.

7.04 How do you utilize community colleges or other institutions of higher education?

Contact local colleges to understand what is possible in terms of dual or concurrent enrollment, allowing CS courses to count for both high school and college credit and determine what will be offered at the college, high school, or online. Dual enrollment refers to programs that allow high school students to take courses at a community college. Concurrent enrollment allows high school students to enroll as a non-degree-seeking student at a university and take courses for full credit. Both dual and concurrent enrollment award the high school credit needed for graduation, along with credit from a college or university, which oftentimes can be transferred to another institution. Connecting with a college can lessen the burden of finding the right staff to teach a CS class and can offer more CS course variety to interested students. These kinds of arrangements also have the incentive to students of reducing the economic burden of postsecondary education.

"We wanted to provide meaningful experiences for our students, so we hosted an 8th Grade Hands-on Career Day. We started small, reaching out to people we knew, and they connected us to others. We emailed and called local industries. Our colleges were a huge resource because they connected us to alumni. We visited other career events to get ideas and network. We had 125 professionals, 6 colleges and served 400 students."

— Linda Riggle, CTE Coordinator, Glenn County Office of Education

8 Expanded Learning Time Opportunities

Photo Courtesy of Riverside Unified School District

8.01 What about providing computer science after-school?

Prioritizing CS during the school day provides more access to all students. It is difficult to make out-of-school program equitable, as not all students can participate due to competing responsibilities (e.g., child care, job), obligations/experiences (e.g., sports, drama, band), transportation, and other factors which disproportionately affect students that have been historically disenfranchised. In addition, if students don't already see CS as part of their identity or interests, they will not choose to engage with it.

However, creating extended exposure opportunities and time for deeper exploration can help students develop a clearer understanding of CS concepts and design projects that connect to personal interests, and out-of-school programs can help with this.

Here are strategies for connecting in-school with out-of-school CS programming while keeping an eye to equity:

- Take advantage of existing out-of-school infrastructure and programming and integrate in-school curriculum or project extensions.
- Partner with nonprofits and NGOs that specifically target underrepresented youth (Girls Who Code, Black Girls Code, SMASH Academy, Code Nation, Girl Scouts) or reach out to local informal learning programs that currently offer CS activities (e.g., libraries, makerspaces).
- Offer extra time for students to continue work on class assignments or passion projects by establishing an afterschool or lunch time computing club
- Investigate robotics programs and competitions (e.g., FIRST, VEX, Wonder League, PiE). These are part of large communities that can help students experience and connect with other youth excited about CS, but consider

how they address equity (i.e. how do they make their program inviting and inclusive, etc.)

- Explore extracurricular programs that have published curricula ready to implement or adapt (e.g., CS First).
- Create opportunities for mentorship or internship programs by reaching out to parents, community members, and local technology and design firms. See more in **Chapter 7 Family, Community, and Industry**.
- See more ideas in **8.02 What else can our students do outside of the school building to build on their computer science experience?**.

For more strategies on how to make informal education experiences more equitable, see this toolkit for Broadening Perspectives on Broadening Participation in STEM from the Center for Advancement of Informal Science Education (CAISE).

8.02 What else can our students do outside of the school building to build on their computer science experiences?

Online courses, Massive Online Open Courses (MOOCs), and summer programs can expose students to new programming languages and expand their concepts of what are the creative possibilities of CS. Here are some other ideas, including those found in EDC's CS toolkit:

- **The Connectory:** This website identifies out-of-school STEM learning opportunities, including CS. Search by age, opportunity, and topic.
- **Maker Faire:** Maker Faires are a good place for students to experience the creative potential of CS, connect with makers, and get ideas for what they can develop.
- **MIT App Inventor:** The tutorials can get students started on developing apps on their own.
- **The Clubhouse Network:** This out-of-school program for kids 10- to 18-years-old encourages students to explore web design, programming, video game design, and 3D modeling.

- **Technovation:** Teams of girls from around the world learn how to code and create apps that address problems in their own communities.

- **Level the Playing Field SMASH Academy**: A STEM-intensive summer residential college prep program for underrepresented students.

- **Coderdojo**: A global network of free, volunteer-led, community-based programming clubs for young people aged 7 to 17.

Photo Courtesy of Riverside Unified School District

9 More Information

9.01 Are there examples of computer science implementation I can look at?

San Francisco, Oakland, and Los Angeles Unified School District have documented and shared their CS implementation efforts. While these examples are all large urban districts, all have elements about their programs that could be useful to any district.

- **San Francisco Unified School District** provides a lot of helpful information on their site on how they have expanded CS education in their district.
- **Oakland Unified School District'**s Claire Shorall describes practical strategies from how OUSD expanded CS access to thousands of students in this Medium article.
- **Los Angeles Unified School District** shares support models for teachers interested in CS education.

"This work is incredible, but sometimes it's scary. I say embrace it. Risk is great — but so is opportunity — when people buy into the process and know there's a method to the madness."
— Claire Shorall, Manager, Oakland Unified School District

9.02 What other computer science education initiatives are currently taking place in California?

California is committed to continuing its leadership by backing K-12 initiatives that ensure students are exposed to and supported in high quality CS education.

- **CSforCA** is a campaign to ensure all schools have access to meaningful and sustainable teaching and learning opportunities in CS. CSforCA is a project that grew out of **Alliance for California Computing Education for Students and Schools (ACCESS)**. CSforCA is a multi-stakeholder coalition of K-12 CS educators, parents, industry leaders, nonprofit organizations, and college and university professors who have worked for CS education equity since 2012. With evidence-based research, CSforCA supports systemic changes needed to provide rigorous and inclusive CS education for all.

- The California State Board of Education approved the Instructional Quality Commission recommendation and adopted the **K-12 CS standards** on September 6, 2018

  "The California Computer Science Standards ... are based on computer science core concepts and core practices, aligned to the K-12 Computer Science Framework ... The standards were developed by educators (members of the State Board of Education-appointed Computer Science Standards Advisory Committee), utilizing work done by the Computer Science Teachers Association. The standards are designed to be accessible to each and every student in California." – California Computer Science Standards (Introduction, p. 1)

- The **Next Generation Science Standards (NGSS)** explicitly placed Computational Thinking as one of the eight practices necessary for science and engineering.

- The **CS Strategic Implementation Plan Panel (CSSIPP)** aims to expand and improve CS education statewide in grades K-12 by developing policy recommendations for the state.

9.03 What other resources should I take a look at?

The Links in the Appendix of this guide has an exhaustive list of all the resources shared in this guide, organized by topic. The following is a list of major national initiatives working towards equitable CS education:

- **Expanding Computing Education Pathways** Alliance seeks to increase the number and diversity of students in the pipeline to computing and computing-intensive degrees by supporting state-level computing education reforms. Through interventions, pathways, partnerships and models that drive state-level computing education change, ECEP supports states as they work to align their state efforts with the national vision for CS for all.

- **CSforALL** is a central resource for individuals and organizations interested in K-12 CS (CS) education. They connect providers, schools and districts, funders, and researchers working toward the goal of providing quality CS education to every child in the United States.

- **Code.org®** is a nonprofit dedicated to expanding access to CS in schools and increasing participation by women and underrepresented minorities.

- The **K-12 CS Framework** is a set of conceptual guidelines for CS education developed by the Association for Computing Machinery, Code.org, CS Teachers Association, Cyber Innovation Center, and National Math and Science in collaboration with states, districts, and the CS community.

- **Outlier** provides research and resources for district implementation to those looking to support CS education efforts and for understanding a variety of perspectives—including teachers, students, and school leaders—on the key supports and barriers that impact introductory CS in high schools.

- **National Center for Women and Information Technology (NCWIT)** equips change leaders with resources for taking action in recruiting, retaining, and advancing women from K-12 and higher education through industry and entrepreneurial careers.

- **National Girls Collaborative Project** brings together organizations throughout the United States that are committed to informing and encouraging girls to pursue careers in science, technology, engineering, and math (STEM).

- **CS Education Acronyms** can help make meaning of all the acronyms in CS education.

9.04 This is a lot of information. Where do I start?

Equitable implementation of CS is a systemic endeavor, and addressing so many factors can feel overwhelming. Below is a checklist of ways to get started. Refer to the GROW Action Plan for more guidance.

- [ ] Organize a team to sign up for the CSforALL SCRIPT workshop.
- [ ] Present to the school board and other elected officials on the need to expand CS offerings in an equitable way.
- [ ] Engage families through Family Code Nights.
- [ ] Explore professional development opportunities for you and your teachers.
- [ ] Engage school leaders, counselors, teachers and other important stakeholders who can help with planning and/or recruitment.
- [ ] Identify data points of where your district is and where you'd like to be in five years, using the Example District Overview to Determine Need.

In addition, more guidance on where to begin can be found in sections **2.09 There is so much to be done to add a new course in middle or high school. How can I simplify the process?** and **9.01 Are there examples of computer science implementation I can look at?**.

9.05 Is it possible to create a version of this guide for my region?

Yes! This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. This license lets you duplicate, edit, build upon, and distribute the guide, as long as you give credit to CSforCA for the original creation, do not use the material for commercial purposes, and distribute your contributions under the same license.

Appendix

Links

FOREWORD

Alliance for California Computing Education for Students and Schools (ACCESS) http://access-ca.org

CSforCA https://csforca.org

CS Strategic Implementation Plan
https://www.cde.ca.gov/pd/ca/cs/cssip.asp

1 INTRODUCTION

National Equity Project: What is Equity?
https://www.youtube.com/watch?v=mIWOtQzsoZg&-feature=youtu.be

CSforALL SCRIPT Program https://www.csforall.org/projects_and_programs/script

Why Computer Science Education Is Essential To The Success Of Our Students And State https://csforca.org/learn-california-computer-science/#future-of-california

edx.org
https://www.edx.org/learn/educational-technology

ala.org https://literacy.ala.org/digital-literacy

Wing (2010) https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Next Generation Science Standards -
Using Mathematics and Computational Thinking
https://ngss.nsta.org/Practices.aspx?id=5

K-12 Computer Science Framework https://k12cs.org

California K-12 Computer Science Standards https://www.cde.ca.gov/be/st/ss/computerscicontentstds.asp

California “majority-minority” state statistics
https://www.census.gov/programs-surveys/acs/technical-documentation/table-and-geography-changes/2017/5-year.html

Computer Science In California Schools: An Analysis Of Access, Enrollment And Equity
https://www.kaporcenter.org/csinschools

Harrison, 2019 https://www.wired.com/story/five-years-tech-diversity-reports-little-progress

Google and Gallup, 2016
http://services.google.com/fh/files/misc/diversity-gaps-in-computer-science-report.pdf

EdSource-Berkeley IGS Survey, 2017
https://edsource.org/documents/Educating-Californias-Children-Survey-Report-2017.pdf

CSforALL, 2020 https://vimeo.com/420069660

Margolis, Estrella, Goode, Jellison-Holme, & Nao, 2017 https://mitpress.mit.edu/books/stuck-shallow-end

Goode, Flapan, & Margolis, 2018 https://jhupbooks.press.jhu.edu/title/diversifying-digital-learning

Ladson-Billings (1995) https://journals.sagepub.com/doi/10.3102/00028312032003465

Sleeter (2012) https://journals.sagepub.com/doi/10.1177/0042085911431472

CSforCA data tool
https://csforca.org/the-data/#community-comparison

SFUSD Supervisor for Computer Science, Curriculum & Instruction https://www.edjoin.org/JobDescriptions/688/2018%20-19%20SY%20Supervisor%20Computer%20Science-20180928095228.pdf

Chicago Public Schools Computer Science Curriculum Instruction Manager https://www.topschooljobs.org/job/703921/computer-science-curriculum-instruction-manager

2 PROGRAM DESIGN AND DEVELOPMENT

California K-12 Computer Science Standards https://www.cde.ca.gov/be/st/ss/computerscicontentstds.asp

SFUSD's middle grades model https://mgredesign.sfusd.edu/redesign-model

SFUSD's pacing guides https://sites.google.com/sfusd.edu/3-5cs#h.p_hSIDSWnvGWa-

Sample scheduled of itinerant and site-based CS staff https://docs.google.com/document/d/1EtW8ImQ4rSFCuCQC8V8Aqhn-drQnuOQ4_qxmWRXMXkY/edit

Next Generation Science Standards (NGSS) https://www.nextgenscience.org

NGSS Practice: Using Mathematics and Computational Thinking https://ngss.nsta.org/Practices.aspx?id=5

California CTE/ICT Standards https://www.cde.ca.gov/ci/ct/sf/ctemcstandards.asp

University of California Curriculum Integration (UCCI) https://ucci.ucop.edu/courses/course-catalog.html

California K-12 CS Standards - Appendix https://www.cde.ca.gov/be/st/ss/documents/compsciappendix.docx

California CS Content Standards Database https://www2.cde.ca.gov/cacs/cs

California's Quality Professional Learning Standards https://www.cde.ca.gov/pd/ps/qpls.asp

Santa Barbara County Office of Education Standards Venn Diagram Exercise https://drive.google.com/file/d/16kJ-M11RT2Z6hxmNW_xJVkyREUmT6Auv/view

Universal Design for Learning (UDL) http://www.cast.org/our-work/about-udl.html#.XJRdjRNKjOQ

Cline & Necochea (2003) https://www.tandfonline.com/doi/abs/10.1207/S15327892MCP0501_4

University of Florida Creative Technology Research Lab's tool to align UDL guidelines with CS https://docs.google.com/document/d/1V7qsDUgrE6_L7xGF3CuPLs7BhuBwp4P-EIGqio27rxo/edit

Linda Darling-Hammond & Gardner (2017) https://learningpolicyinstitute.org/product/effective-teacher-professional-development-report

CSTA PD opportunities database https://csteachers.org/page/quality-pd

CSforAll Member Directory https://www.csforall.org/projects_and_programs/member_directory

Scratch Jr Curricula https://www.scratchjr.org/teach/curricula

Creative Computing Curriculum https://creativecomputing.gse.harvard.edu/guide

CSinSF's adaptation of Creative Computing https://creativecomputing.gse.harvard.edu/guide

CS Discoveries https://code.org/educate/csd

Bootstrap Algebra https://www.bootstrapworld.org/materials/algebra

Bootstrap Physics https://www.bootstrapworld.org/materials/physics

Bootstrap Data Science https://www.bootstrapworld.org/materials/data-science

Exploring CS http://www.exploringcs.org

Introduction to Data Science https://www.introdatascience.org

AP® CS Principles https://apcentral.collegeboard.org/courses/ap-computer-science-principles

AP® CS A https://apstudent.collegeboard.org/apcourse/ap-computer-science-a

CSAwesome https://www.csawesome.org

California K-12 CS Standards - Appendix https://www.cde.ca.gov/be/st/ss/documents/compsciappendix.docx

Bootstrap Algebra https://www.bootstrapworld.org/materials/algebra

K–12 Computer Science Framework https://k12cs.org/curriculum-assessment-pathways

Just Equations https://justequations.org

CDE CALPADS course codes https://www.cde.ca.gov/ds/sp/cl/systemdocs.asp

California Commission on Teacher Credentialing https://www.ctc.ca.gov

CSforCA FAQ on Who can Teach CS https://csforca.org/wp-content/uploads/2020/08/v2-teaching-com_47895427.pdf

UCOP A-G Course Management Portal https://hs-articulation.ucop.edu/agcmp#/login

UC/CSU A-G college preparatory requirements http://admission.universityofcalifornia.edu/freshman/requirements/a-g-requirements/index.html

ACCESS website FAQ http://access-ca.org/issues-solutions/faq-about-computer-science

CSTA - Professional Development Opportunities https://www.csteachers.org/page/Learning

Microsoft TEALS Program https://www.tealsk12.org

CSforAll SCRIPT Program https://www.csforall.org/projects_and_programs/script

Intro to Data Science https://www.mobilizingcs.org

California K-12 CS Standards https://www.cde.ca.gov/be/st/ss/computerscicontentstds.asp

CTE pathway through industry sectors such as ICT and Public Services https://www.cde.ca.gov/ci/ct/sf/documents/pubservices.pdf

National Institute for Standards and Technology (NIST) https://www.nist.gov

National Institute for Cybersecurity Education (NICE) https://www.nist.gov/itl/applied-cybersecurity/nice

NICE's list of online resources for learning about cyber security https://www.nist.gov/itl/applied-cybersecurity/nice/resources/online-learning-content

Cyber Guild https://cyber-guild.org

Cal Poly SLO's CA Cybersecurity Institute https://cci.calpoly.edu

DoD/CSforALL JROTC CS program https://www.csforall.org/projects_and_programs/jrotc

3 STUDENTS AND RECRUITMENT

California CS Strategic Implementation Plan https://www.cde.ca.gov/pd/ca/cs/cssip.asp

NCWIT's Counselors for Computing https://www.ncwit.org/project/counselors-computing-c4c

CS Recruitment Toolkit - Microsoft https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE2FztP

Recruitment Toolkit - College Board https://apcentral.collegeboard.org/courses/ap-computer-science-principles/recruitment-toolkit

Tips for HS Student Recruitment – CSTeachingTips http://csteachingtips.org/tips-for-recruitment-in-HS

4 IN THE CLASSROOM

California K-12 CS Standards https://www.cde.ca.gov/be/st/ss/computerscicontentstds.asp

Computer Science Teachers Association (CSTA) CS Standards https://www.csteachers.org/page/standards

K-12 CS Framework https://k12cs.org

International Society for Technology in Education (ISTE) https://www.iste.org

CS Unplugged https://csunplugged.org/en

Code.org instructional strategies (on page 4) https://curriculum.code.org/csd-18

NCWIT Guide to inclusive computer science education https://www.ncwit.org/resources/guide-inclusive-computer-science-education-how-educators-can-encourage-and-engage-all

Delamont (1996) https://www.taylorfrancis.com/books/9781351201476

Tenenbaum & Ruck (2007) https://doi.org/10.1037/0022-0663.99.2.253

MIT's interactive case studies for understanding and addressing bias in STEM https://mit-teaching-systems-lab.github.io/unconscious-bias

CSforCA http://csforca.org

Goode (2008) https://dl.acm.org/doi/10.1145/1352322.1352259

Join CSforCA https://csforca.org/sign-up

Ladson-Billings (1995) https://journals.sagepub.com/doi/10.3102/00028312032003465

Sleeter (2012) https://journals.sagepub.com/doi/10.1177/0042085911431472

Center for Culturally Responsive Teaching and Learning https://www.culturallyresponsive.org

Culturally Responsive Teaching and the Brain https://crtandthebrain.com

Teaching Tolerance https://www.tolerance.org

Anti-Defamation League Anti-Bias Resources https://www.adl.org/education/resources/tools-and-strategies/anti-bias-tools-strategies

Rethinking Schools https://www.rethinkingschools.org

CS Principles Performance Tasks https://apcentral.collegeboard.org/pdf/ap-csp-student-task-directions.pdf?course=ap-computer-science-principles

ECS curriculum http://www.exploringcs.org/curriculum/curriculum-scope-sequence

Principled Assessment of Computational Thinking https://pact.sri.com/ecs-assessments.html

Assessing Creativity in Computing Classrooms https://creativecomputing.gse.harvard.edu/assessment

National Research Council (2014) https://www.nap.edu/read/18409/chapter/6

CSforCA Data Tool https://csforca.org/the-data/#community-comparison

CS Unplugged https://csunplugged.org/en

CSforALL Teachers Blog - Pair Programming https://www.csforallteachers.org/blog/pair-programming

Digital Promise's Digital Learning Playbook https://digitalpromise.org/online-learning/technology

CA IT in Education https://cite.org/default.aspx

California's Digital Divide (PPIC) https://www.ppic.org/publication/californias-digital-divide

The Digital Divide in Southern California https://broadbandnow.com/research/maps/digital-divide-southern-california

California Emerging Technology Fund (CETF) http://www.cetfund.org

CSforCA: Online Learning Is Ok For Now But Not Forever: Inequality And Schooling In The Age Of Covid-19 https://csforca.org/online-learning-is-ok-for-now-but-not-forever-inequality-and-schooling-in-the-age-of-covid-19

NYC CS4ALL Resources for Remote CS Instruction http://cs4all.nyc/2020/04/02/resources-for-remote-computer-science-instruction

5 PREPARING AND SUPPORTING TEACHERS

Edtrust West report: California's Teacher of Color Shortage https://west.edtrust.org/resource/seen-heard-reflected-a-look-at-californias-teacher-of-color-shortage

Who is authorized to teach CS (ACCESS) http://access-ca.org/infographic-building-a-robust-cs-education-teaching-force

Administrator's Assignment Manual https://www.ctc.ca.gov/docs/default-source/credentials/manuals-handbooks/administrator-assignment-manual.pdf?sfvrsn=8

California supplemental authorization in CS https://www.csinsf.org/supplemental-authorization.html

CS supplementary authorization programs:

- UC Irvine https://sites.uci.edu/cs1c/cs1catoc-teacher-certificate-program
- UC Riverside https://www.extension.ucr.edu/certificates/19013519/educationandcredentials/subjectmatterspecialization/computerscienceeducation
- SFSU http://bulletin.sfsu.edu/colleges/education/credentials
- UCSD http://csteachers.ucsd.edu

California's Commission on Teaching Credentialing https://www.ctc.ca.gov

Microsoft and NCWIT CS professional development https://www.ncwit.org/sites/default/files/resources/cs_professional_development_guide.pdf

The Commission on Teacher Credentialing https://www.ctc.ca.gov/commission/default

Summer of CS https://www.summerofcs.org

Join CSforCA https://csforca.org/sign-up

Margolis, Ryoo, & Goode, 2017 http://www.exploringcs.org/wp-content/uploads/2014/04/SeeingMyselfArticle.pdf

Computer Science Teachers Association (CSTA) https://www.csteachers.org

CS for All Teachers https://csforallteachers.org

CS Teaching Tips http://csteachingtips.org

ScratchEd https://scratched.gse.harvard.edu

CSforCA http://www.csforca.org

UCLA CS Equity Project https://centerx.gseis.ucla.edu/computer-science-equity-project

Join CSforCA https://csforca.org/sign-up

Strategic CSforALL Resource & Implementation Planning Tool (SCRIPT) https://www.csforall.org/projects_and_programs/script

Project TACTIC https://ctrl.education.illinois.edu/TACTICal

Universal Design for Learning (UDL) framework http://udlguidelines.cast.org

The Alliance for Access to Computing Careers (AccessComputing) https://www.washington.edu/accesscomputing

AccessComputing resources https://www.washington.edu/accesscomputing/resources-home

Example District Overview Tool https://docs.google.com/document/d/1zYhuA1IquEV_L08NvPEqsq3QTxNQWAPoly1Nc4kR4Fc

National Academies of Sciences, Engineering and Medicine - English Learners in STEM Subjects https://www.nap.edu/catalog/25182/english-learners-in-stem-subjects-transforming-classrooms-schools-and-lives

PiLa-CS https://www.pila-cs.org

PiLa-CS examples
https://www.pila-cs.org/educator-resources

Scratch https://scratch.mit.edu

Scratch in many languages https://en.scratch-wiki.info/wiki/Scratch_in_Many_Languages

Project TACTIC TACTICal brief on Paraeducators during K-12 Computer Science Instruction https://ctrl.education.illinois.edu/TACTICal/paraeducator

SFUSD teaching rubric https://sites.google.com/sfusd.edu/csplc/resources/teaching-rubric

6 FUNDING

TEALS https://www.microsoft.com/en-us/teals

Local Control Accountability Plan (LCAP)
https://www.cde.ca.gov/re/lc

California K-12 CS Standards https://www.cde.ca.gov/be/st/ss/computerscicontentstds.asp

Local Control Funding Formula (LCFF)
https://www.cde.ca.gov/fg/aa/lc/lcffoverview.asp

LA School Board passes CS Education resolution
https://home.lausd.net/apps/news/article/866803

TimeforCS findings
http://outlier.uchicago.edu/TimeforCS/findings

Occupational Outlook Handbooks for Computer and IT Occupations https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Career Technical Education Incentive Grant Program
https://www.cde.ca.gov/ci/ct/ig

K12 Strong Workforce Program https://www.cccco.edu/About-Us/Chancellors-Office/Divisions/Workforce-and-Economic-Development/K12-Strong-Workforce

Workforce & Economic Development
https://www.cccco.edu/About-Us/Chancellors-Office/Divisions/Workforce-and-Economic-Development

K12 SWP Pathway Coordinator https://www.cccco.edu/About-Us/Chancellors-Office/Divisions/Workforce-and-Economic-Development/K12-Strong-Workforce/K12-SWP-Pathway-Coordinator-Timeline-and-SOW

E-Rate https://www2.ed.gov/about/inits/ed/non-public-education/other-federal-programs/fcc.html

California IT in Education https://cite.org/default.aspx

7 FAMILY, COMMUNITY, AND INDUSTRY

Google and Gallup (2016)
http://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf

Family Code Night http://www.familycodenight.org

EDC's CS toolkit for families https://www.edc.org/why-computer-science-education-toolkit-parents

STEM Family https://www.stem.family

CS Remote Learning Guide for Families https://docs.google.com/document/d/1H2MmnDBTam-ivXjP46-Ur8Pk5km30y40GmDn_f-1d_0/edit

CS Remote Learning Guide for Families (Translated) https://drive.google.com/drive/folders/1R6YFzIBqQ7pZhCcsx97h0IWeztb6evP3

Common Sense Media
https://www.commonsensemedia.org

CSforCA Data Tool
https://csforca.org/the-data/#community-comparison

Code.org Promote CS https://code.org/promote

California School Boards Association http://csba.org

CA School Boards Association Governance to expand STEAM access https://www.csba.org/-/media/CSBA/Files/GovernanceResources/GovernanceBriefs/2018GovBriefSTEM2.ashx?la=en&rev=9ed1a99a8bd74a55a8cfc2538cf51138

California PTA School Smarts Parent Program https://capta.org/programs-events/school-smarts

SFUSD Work Based Learning Options https://drive.google.com/file/d/1NVb3PPe5-wbSNvfcQOKIRlvLRObZJ3p0/view

Families in Schools https://www.familiesinschools.org

TEALS https://www.tealsk12.org

EnCorps https://encorps.org

CSTeachingTips for volunteers http://csteachingtips.org/Tips-for-classroom-volunteers

K12 SWP Pathway Coordinators https://www.cccco.edu/About-Us/Chancellors-Office/Divisions/Workforce-and-Economic-Development/K12-Strong-Workforce/K12-SWP-Pathway-Coordinator-Timeline-and-SOW

K12 Strong Workforce Program https://www.cccco.edu/About-Us/Chancellors-Office/Divisions/Workforce-and-Economic-Development/Strong-Workforce-Program

Regional Workforce Intermediaries https://cwdb.ca.gov

Valley Vision https://www.valleyvision.org

Workforce Development Board https://cwdb.ca.gov/local_boards/local_workforce_investment_associations

CSTA chapters https://csteachers.org/305564/Page/Show?ClassCode=Page&Slug=about-csta-chapters

Nepris https://www.nepris.com

Small School Districts Association http://www.ssda.org

8 EXPANDED LEARNING TIME OPPORTUNITIES

CAISE Broadening Perspectives on Broadening Participation in STEM Toolkit https://informalscience.org/broadening-perspectives

EDC's CS toolkit for families https://www.edc.org/why-computer-science-education-toolkit-parents

The Connectory https://theconnectory.org

Maker Faire https://makerfaire.com

MIT App Inventor http://appinventor.mit.edu

The Clubhouse Network https://theclubhousenetwork.org

Technovation https://technovationchallenge.org

SMASH Academy https://www.smash.org/programs/smash-academy

Coderdojo https://coderdojo.com

9 MORE INFORMATION

District Implementations Examples:

- San Francisco Unified School District https://www.csinsf.org
- Oakland Unified School District https://link.medium.com/PL6AyRKBXT
- Los Angeles Unified School District https://achieve.lausd.net/page/10023

CSforCA https://csforca.org

ACCESS http://access-ca.org

California K-12 CS Standards https://www.cde.ca.gov/be/st/ss/computerscicontentstds.asp

K-12 CS Framework https://k12cs.org

NGSS Using Mathematical and Computational Thinking https://ngss.nsta.org/Practices.aspx?id=5

CS Strategic Implementation Plan Panel (CSSIPP) https://www.cde.ca.gov/pd/ca/sc/cssip.asp

Expanding Computing Education Pathways Alliance ecepalliance.org

CSforALL https://www.csforall.org

Code.org code.org

K-12 CS Framework https://k12cs.org

Outlier http://outlier.uchicago.edu/basics

National Center for Women and Information Technology (NCWIT) https://www.ncwit.org

National Girls Collaborative Project https://ngcproject.org

LeadCS CS Education Acronyms (Acronym Dictionary) https://s3.amazonaws.com/leadcs/downloads/communication/Computer_Science_Acronyms.pdf

GROW Action Plan https://docs.google.com/document/d/1CI7aT4dLTBF3BOH9KLSV3DH5AF4k9RaRXib2XTTIg-k/edit?usp=sharing

CSforALL SCRIPT https://www.csforall.org/projects_and_programs/script

Example District Overview to Determine Need https://docs.google.com/document/d/1zYhuA1lquEV_L08NvPEqsq3QTxNQWAPoly1Nc4kR4Fc/edit

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License https://creativecommons.org/licenses/by-nc-sa/4.0

References

California Department of Education. (2015, December 4). LCFF Frequently Asked Questions. Retrieved from https://www.cde.ca.gov/fg/aa/lc/lcfffaq.asp.

Cline, Z., & Necochea, J. (2003). Specially designed academic instruction in English (SDAIE): More than just good instruction. Multicultural Perspectives, 5(1), 18-24.

CSforALL. [CSforALL]. (2020, May 18). Supporting Equitable CSed in Schools During COVID [Video file]. Retrieved from https://vimeo.com/420069660.

Darling-Hammond, L., Hyler, M. E., & Gardner, M. (2017). Effective teacher professional development. Retrieved from https://learningpolicyinstitute.org/product/effective-teacher-professional-development-report.

Delamont, S. (1990). A woman's place in education: myths, monster and misapprehensions. British Educational Research Journal, 16(1).

EdSource. (2017). Educating California's Children and Youth: A Summary of the Findings from a Survey of Voters about K-12 Schools. Retrieved from https://edsource.org/documents/Educating-Californias-Children-Survey-Report-2017.pdf.

Goode, J. (2008, March). Increasing Diversity in K-12 computer science: Strategies from the field. In Proceedings of the 39th SIGCSE technical symposium on Computer science education (pp. 362-366).

Goode, J., Flapan, J., & Margolis, J. (2018). Computer Science for All: A School Reform Framework for Broadening Participation in Computing. In W. G. Tierney, Z. B. Corwin, & A. Ochsner (Eds.), Diversifying Digital Learning: Online Literacy and Educational Opportunity (pp. 45-65). Baltimore, MD: Johns Hopkins University Press.

Google Inc. & Gallup Inc. (2016). Diversity Gaps in CS: Exploring the Underrepresentation of Girls, Blacks and Hispanics. Retrieved from http://goo.gl/PG34aH.

Gribble, J., Hansen, A., Harlow, D., & Franklin, D. (2017, June). Cracking the code: the impact of computer coding on the interactions of a child with autism. In Proceedings of the 2017 Conference on Interaction Design and Children (pp. 445-450).

Hansen, A.K., Hansen, E., Iveland, A., Gribble, J., Moran, A., Harlow, D.B., Franklin, D. (2017, April). Understanding the challenges and potential of computer science education for elementary school students with disabilities. Paper presented at the annual meeting of the American Educational Research Association (AERA 2017). San Antonio, TX.

Harrison, S. (2019, October 1). Five Years of Tech Diversity Reports—and Little Progress. Wired. Retrieved from https://www.wired.com/story/five-years-tech-diversity-reports-little-progress.

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org.

Ladson-Billings, G. (1995). Toward a theory of culturally relevant pedagogy. American educational research journal, 32(3), 465-491.

Margolis, J., Estrella, R., Goode, J., Jellison-Holme, J., & Nao, K. (2017). Stuck in the Shallow End: Education, Race, & Computing (2nd ed.). MIT Press: Cambridge, MA

Margolis, J., Ryoo, J., & Goode, J. (2017). Seeing myself through someone else's eyes: The value of in-classroom coaching for computer science teaching and learning. Transactions on Computing Education, 17(2), 1-18. doi: 10.1145/2967616.

National Equity Project. (2017, May 31). Intro to Equity Webinar [Video File]. Retrieved from https://www.youtube.com/watch?v=mIWOtQzsoZg&feature=youtu.be.

Sleeter, C. E. (2012). Confronting the marginalization of culturally responsive pedagogy. Urban Education, 47(3), 562-584.

National Research Council. (2014). Developing assessments for the next generation science standards. National Academies Press.

Scott, A. Koshy, S. Rao, M., Hinton, L., Flapan, J., Martin, A., McAlear. F. 2019. Computer Science in California's Schools: An Analysis of Access, Enrollment, and Equity, Kapor Center, Oakland, CA.

Tenenbaum, H. R., & Ruck, M. D. (2007). Are teachers' expectations different for racial minority than for European American students? A meta-analysis. Journal of educational psychology, 99(2), 253.

Tucker, A., McCowan, D., Deek, F., Stephenson, C., Jones, J., & Verno, A. (2006). A model curriculum for K–12 computer science: Report of the ACM K–12 task force curriculum committee (2nd ed.). New York, NY: Association for Computing Machinery.

U.S. Census Bureau, 2013-2017. (2017). American Community Survey 5-Year Estimates. Retrieved from https://www.census.gov/programs-surveys/acs/technical-documentation/table-and-geography-changes/2017/5-year.html.

Wing, J. (2011). Research notebook: Computational thinking—What and why. The link magazine, 6.

Acknowledgments

Written by:

Julie Flapan, UCLA Center X
Roxana Hadad, UCLA Center X
Claire Shorall, formerly Oakland Unified School District
Bryan Twarek, Computer Science Teachers Association

With

Jared Amalong, Sacramento County of Education
Lauren Aranguren, Santa Barbara County Office of Education
Mark Archon, Fresno County Office of Education
Edward Campos, Kings County Office of Education
Shelley Fox, Math Jeans
Dawn Guest-Johnson, Los Angeles Unified School District
Jessie Gurbada, Riverside Unified School District
Philip James, Glenn County Office of Education
Steve Kong, Riverside Unified School District
Mark Lantsberger, San Diego County Office of Education
Karen Larson, Santa Clara County Office of Education
Michelle Lee, San Francisco Unified School District
Dylan Lira, Compton Unified School District
Jane Margolis, University of California, Los Angeles
Bill Marsland, San Francisco Unified School District
Sophia Mendoza, Los Angeles Unified School District
Sonal Patel, San Bernardino County Superintendent of Schools
Anthony Quan, Los Angeles County Office of Education
Debra Richardson, University of California, Irvine
Linda Riggle, Glenn County Office of Education
Andrew Rothman, San Francisco Unified School District
Jean Ryoo, University of California, Los Angeles
Raelynn Sharp, Sacramento County Office of Education
Beth Simon, University of California San Diego
Emily Thomforde, San Mateo County Office of Education
Matt Zuchowicz, Santa Barbara County Office of Education

Design, editing, and layout by:

Michelle Choi

Contact us at info@csforca.org. This material is based upon work supported by the National Science Foundation under Grant #1837780. Thank you to Code.org and Expanding Computing Education Pathways for additional support.

Suggested citation: Flapan, J., Hadad, R., Shorall, C., & Twarek, B. (2020). CS Equity Guide: A K-12 education leader's guide to designing, scaling, and sustaining equitable computer science in California. Los Angeles, California: University of California, Los Angeles.