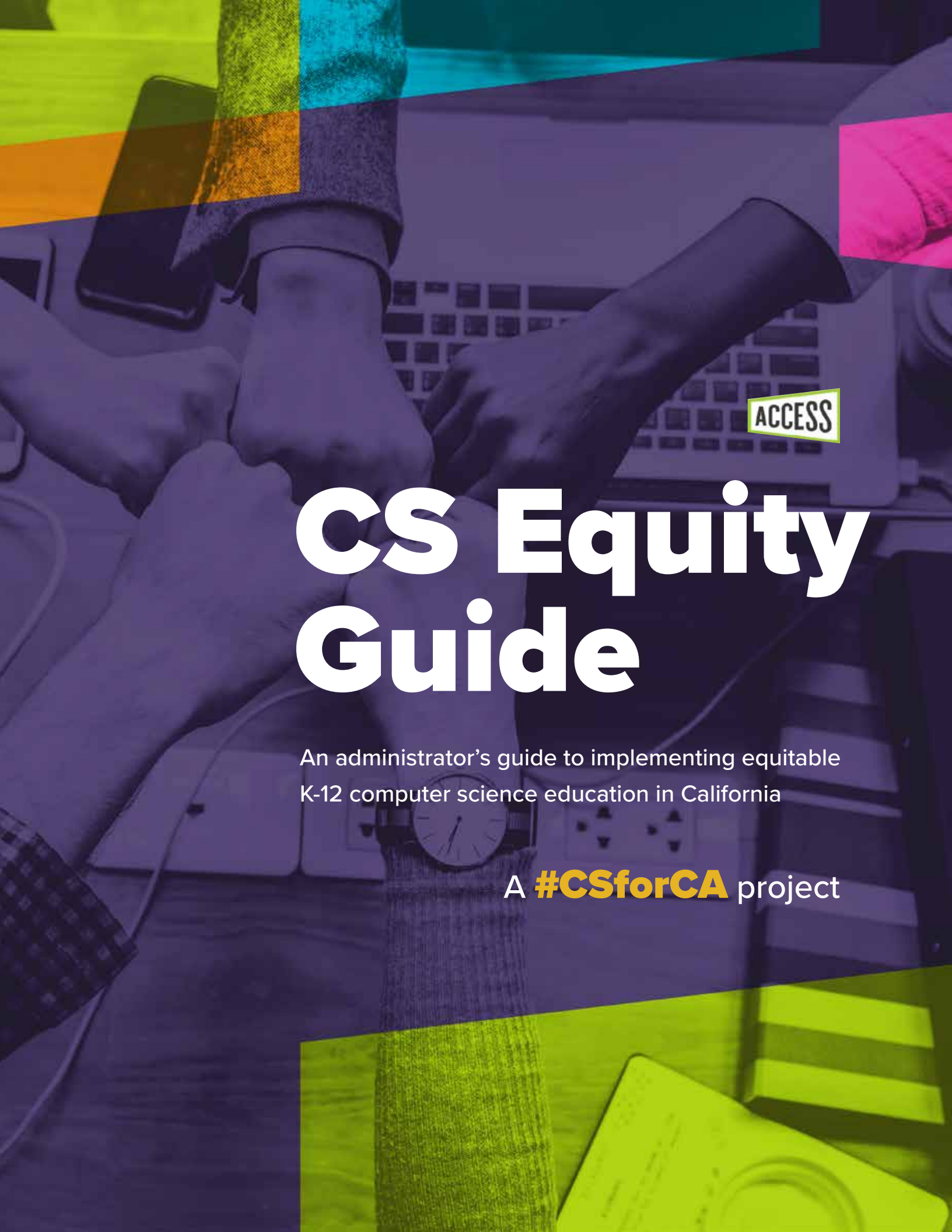


CS Equity Guide

An administrator's guide to implementing equitable K-12 computer science education in California

A #CSforCA project

by administrators, *for* administrators

This guide is designed for administrators interested in implementing **equity-minded computer science (CS)** in their schools, districts, or counties. If you are a principal, district or county leader interested in bringing CS to all your students, this guide is for you.

As part of ongoing research and implementation in CS education equity, UCLA researchers conducted interviews with leaders from early-adopter LEAs (local education agencies) who responded to the most frequently asked questions they receive from colleagues across the state. This guide includes their answers to these questions.

We hope this guide provides an important resource for school administrators 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 special needs. We would like to thank the district leaders who shared with us their successes and challenges towards making CS education available to all in California.

With support from the National Science Foundation, the **Alliance for California Computing Education for Students and Schools (ACCESS)** was formed in 2012 to broaden participation in CS education in California, especially for girls, low-income students and students of color. ACCESS launched **CSforCA** in 2016 as part of the national CSforAll movement, with the support of then-Lt. Governor Gavin Newsom, CA 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 have all been working to develop a CS Strategic Implementation Plan.

While CSforCA and the Implementation plan are statewide campaigns working 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. In an effort to build capacity at the local level, 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.

As the landscape of California K-12 CS education evolves, 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. Please contact us at **info@csforca.org**.

1 Introduction

1.1 Why computer science?

Computer science (CS) education is foundational learning for the future of California.

CS provides students the skills of problem solving, critical thinking, creativity, and collaboration needed to succeed in the 21st century. In order to become active members of modern society, students need to learn to become creators of technology, not just passive consumers. CS helps prepare students for college, careers, and community engagement. Increased access to CS can allow students to discover innovative ways to solve problems in their communities and to learn about college majors or careers they might never thought were possible.

1.2 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*** is one tool in the CS toolbox, but CS teaches more than just how to code. CS is about solving problems using computers. Coding (or programming) is using a language in order to carry out these solutions.

- ***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 (NSF). 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 Science and Engineering Practices in the Next Generation Science Standards.

1.3 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 its effect on communities. Special attention must be paid to provide access to all students, especially those who have been historically underrepresented in CS, including African Americans, Latinx, Native Americans, Pacific Islanders, women, English language learners, LGBTQIA, and students with special needs. 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 understanding and demystifying computing technology and being able to think critically of its potential and impacts.

California, home to Silicon Valley, is the 5th 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). Yet students of color, low-income students, females, and English language learners are extremely underrepresented in CS (See Appendix).

These very patterns are then reflected in the workforce. The computing industry is currently dominated by a narrow subset of the population. When the creators of technologies broadens, those technologies will reach more people and more effectively. This effectiveness encompasses the reality that when CS designs are made by a narrow section of our population, those designs are not made with other perspectives, needs, or experiences in mind.

Equitable access to the foundational learning CS provides will give every student the opportunity to thrive in the 21st century. Equitable access has to begin in K-12, and the earlier students are presented with quality opportunities in CS, the better.

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).

1.4 What does equity in computer science education mean?

Historically, students have been denied access to CS due to a variety of factors: counselors and teachers 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 educational opportunities 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. While making CS classes available to all populations, it's important that all students feel included and engaged in the content. 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 computer science 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).

"Our superintendent made a clear call to action: create an Equity Task Force of different stakeholder groups to address opportunity and achievement gaps that existed within our district. We discussed what equity looked like and then went through an iterative process to agree upon our district's definition of 'equity'. The process of defining 'equity' took several weeks, but ensured that all of our stakeholders across our schools and community were all on the same page. The definition served as an anchor for objectives for our district. We then tied in action items to those broader objectives."

1.5 Who should lead a computer science initiative?

For your CS program to be successful, it is imperative to designate a district/school/county lead whose sole or primary responsibility is to lead the implementation. This person 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. This person will also work with schools to add CS to the master calendar.

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. They contain a lot of the necessary skills and requirements that a good leader for CS education would need. Consider what department this person will be located in, 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.

Some necessary qualifications:

- Knowledge of CS
- Experience as a teacher and instructional leader
- Ability to motivate and inspire principals
- Ability to effectively develop and leverage relationships

2 Developing Pathways

2.1 Does computer science have its own standards?

Computer science standards were adopted in California 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.

2.2 What is an appropriate course sequence for computer science?

There is no “one size fits all” for CS pathways; 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 learn CS, so careful considerations should be made so that students in both college and career pathways can have meaningful experiences with CS.

A common pathway at the middle/high school level is (1) Computer Science Discoveries; (2) Exploring Computer Science; (3) AP Computer Science Principles; and (4) CS A. Of course, there are many courses that an interested student could take such as digital media art and design, robotics, game development, app design, music technology, or other CS-related courses after having first experienced it through an introductory course. CS can be a stand-alone course (most likely in middle and high school), but there are various opportunities to integrate CS with other subject areas, initiatives, and resources at all levels.

2.3 What course code should our CS courses have?

Identify computer science course pathways that will ensure equitable access for all students. How you choose to provide a course code in CS impacts who is authorized to teach it. For example, a course that is coded as a Career Technical Education (CTE) course can only be taught by a teacher with a CTE and/or Information and Communication Technologies (ICT) authorization, whereas a CS course coded in general education/non-CTE can only be taught by a teacher with a single subject credential in math, business, or Instructional Technology Education (ITE). Or, a teacher with a multiple subjects or different single subject credentials may 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 Section 5.2: Who is authorized to teach computer science in California?)

Equity considerations:
Always consider the implications for equity when deciding what is compulsory and what is elective. While it may be easier to add CS as an elective, be mindful of who actually has access to CS if an elective course is their only access to it. For example, students who have their elective time taken up by other required courses (English learners, students with disabilities, etc.) may be automatically denied access to CS if it is offered only as an elective.

CTE allows access to additional Perkins funding but restricts who can teach the course. 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 access 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 California Commission on Teacher Credentialing.

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

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

When creating your master schedule, CS should be included for all students. This inclusion may look different at different levels, but universal exposure at each grade band is important.

Although it is more likely that CS is integrated into existing content in younger grades and exists as its own course in older grades, there are various exciting opportunities to integrate with other courses and resources at the middle and high school level. True integration requires extensive planning and instructional time, so setting aside time for teacher collaboration and planning is essential.

Unlike educational technology, CS does have content, so implemention must address standards of at least two (2) 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 appendix to the California K-12 CS standards has alignment maps to all state standards.

In elementary and middle school, CS is often integrated into existing curricula. For example, students can collect and analyze data in science class, program interactive stories in language arts, or create visualizations in math class.
CS can also be offered as a "special" alongside physical education, art, and music. In the middle grades, consider placing CS in a compulsory exposure wheel, where all students get access to CS, as well as the arts, health, foreign languages, etc., as seen in San Francisco Unified School District's middle grades model. Be mindful of schedule overlaps and the students who could be unintentionally excluded, for example, English learners, students in drama, resource class, or orchestra, etc. If there is already a full slate of graduation requirements, consider options for course integrations. The University of California Curriculum Integration (UCCI) provides examples of integrated courses that meet both CTE and General Education requirements.

2.5 What curricula is recommended?

There are various excellent curricula available that demonstrate the powerful and multifaceted nature of CS. While various curricula can meet your needs, be sure to consider the implications for equity when selecting a course. Below are some popular choices:

- **3rd-7th grades: Creative Computing** - This seven-unit curriculum introduces young people to the creative possibilities of programming using **Scratch**, a free online programming software. Students learn to express themselves through creating and sharing projects.

- **7th/8th/9th 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 variety of contexts and aims to show all students that CS is for them.

- **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. Assignments and instruction 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).**

- **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 (CSA)** - This year-long course 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." **This course is usually taken by students who know they want to major in CS.**

- **6th-12th grades: Bootstrap** - 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.

- There are lots of other excellent providers of curricula such Project Lead the Way, C-STEM, Google's CS First, Beauty and Joy of Computing, 9 Dots, Mobile CSP, Intro to Data Science, and Code.org. A list of other curricula is provided here.

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), in which the curricula is designed to meet the needs of special populations can help everyone learn. See more in Chapter 5: Preparing and Supporting Teachers.

2.6 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 UC/CSU A-G college preparatory requirements to provide a greater incentive for students to add it to their schedules.

The vast majority of CS courses in California are designated a "G" elective and many CS courses are part of CTE pathways. 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. 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 they are full with EL coursework.

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 students to take it. Therefore, a CS course could potentially count as a math, science, CTE pathway course, and/or an elective, depending on the specific course and its approval from UC. Many schools like to classify their CS courses in the CTE pathway because they claim Perkins Funding to support a teacher with CTE/ICT authorization. Each LEA will make its own decision on what courses count in your specific locale.

Check with the AG Course Management Portal and speak with your Human Resources to determine what courses can be offered and what corresponding teacher certifications are required. For the latest information about these changes, see the FAQ on the ACCESS website.

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

From our experience, we recommend that each school in your district do the following in advance of the upcoming school year:

- 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). Regardless of the curriculum, it is essential that teachers participate in Professional Development for CS.

- Identify teachers from your school who may be interested in teaching CS. Common sources include math, science, creative arts, CTE technology teachers, and anyone else who wants to bring computing to the school. A teacher 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. Computer Science Teachers Association (CSTA) helps coordinate summer professional development. Check their 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 the 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 special needs. 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 the next year has appropriate layout for collaboration, lab work, and working computers. Connect with your IT team to install and test any software that will be used next year. Find out more about hardware and software requirements in Section 4.5 of this guide.

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

When piloting a program, it can be difficult to generalize from the experiences of one school, so recognize each school's varying culture and needs as you expand. Starting a district-wide initiative ensures that all schools have access and that no one school is singled out. It also allows more schools to discover what is working and what is not.

"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. Teachers with lesser content knowledge can get support from those with greater content knowledge, while maintaining the pedagogical flow of their class."

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

Start at the level you have the most momentum for CS. For example, consider where you have excited principals and teachers, best access to professional development opportunities, etc. and go from there. The goal is to have a coherent pathway from Kindergarten through 12th grade. Coherence will develop through the use of standards and curriculum and then coordination among elementary, middle, and high school teachers.

"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."

2.10 How do I decide what courses to offer?

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 the curriculum you would like to use is already approved for A-G. Getting new courses approved can be a challenge, so pre-approved course descriptions can simplify the process.

2.11 What about data science?

Data science is a burgeoning subset of CS that involves collecting, analyzing, visualizing, and communicating with data. The Introduction to Data Science course offered by the IDS project at UCLA is a great K-12 resource. Data science is currently accepted as a "C" math in A-G.

3 Students and Recruitment

3.1 Who can do computer science?

In today's world, 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 special needs, to enroll in the CS courses. A variety of skills, experience, and backgrounds are not only useful for learning CS, but different perspectives and approaches can help solve problems in new ways and meet the needs of more people, so it is important to be intentional about including all students.

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

School counselors and teachers play a pivotal role in encouraging students to take CS courses.

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)

2. **Recruit underrepresented students in groups**, and have them encourage their friends to sign up, so they are not isolated

Counselors can sometimes inadvertently deny opportunity based on assumptions or misunderstandings. Counselors are powerful advocates, but some may need to learn more about the value and purpose of CS

education in order to be better informed about how to encourage more youth to enroll in CS courses. They may not know their students' personal experiences or interests with technology and computing without delving deeper. We have found that students often do not share about these interests or experiences unless explicitly asked.

Since counselors rarely have the time to be able to find out from every student if they are interested or have experience with digital technolgoy, and since *all* students can benefit from taking at least one CS course in the middle/high school level, we suggest counselors make efforts to ensure all students experience CS at least once along their pathways to either connect to prior interests or spark new ones. At schools like Mendez High in Los Angeles, ECS has become a 9th grade course requirement, ensuring that every student has an experience with and understanding of what CS is before moving on in their high school pathways, and making it easier for counselors to advise students.

Counselors can also help with recruitment. Resources and professional development opportunities are available to school counselors through NCWIT's Counselors for Computing. Make sure to engage with your counselors to help recruit students into courses.

Teachers and administrators play an important role as well in recruiting underrepresented students into CS classes. School-wide fairs can highlight the work of young women and students of color already successful in CS and be role models for other curious students.

Useful resources for assisting with recruitment:

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

4 In the Classroom

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

In September 2018, California adopted K-12 CS standards that 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

4.2 How should a computer science class be taught?

While computers 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. Students should be encouraged to explore the different facets of computing through the hands-on creation of artifacts. As collaboration and communication are integral to computing, prioritize group projects and teamwork throughout courses.

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 constant 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
- Cultivate students' awareness of existing dominant power structures in society (Ladson-Billings, 1995; Sleeter, 2012)

More resources for culturally responsive education can be found on the National Education Association website.

The best teaching practices identified in other subject areas and education more broadly apply to CS classrooms as well. Instruction should be student-centered, differentiated, and culturally responsive. A teacher who already has expertise in facilitating collaborative work can use the same strategies in a CS classroom. The focus on Teacher Practices research on the ECS site, as well as the list of instructional strategies from the Code.org CS Discoveries curriculum guide (starting on p. 4) may be helpful.

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

Develop a program plan and evaluation to measure your success and learn from your mistakes. Once you have begun your CS program, establish program monitoring with your existing research teams and teacher input. Begin by measuring access (where are classes offered?),

participation (who takes CS?), and achievement (grades and pass rates on AP exams). Once that is in place, move onto better performance/assessment measures, instructional quality, teacher retention, etc. and monitor continually. Whatever program plan you come up with should be available for stakeholders to view and refine.

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

As in other content areas, assessment should serve as a guide for how to structure your courses. Portfolio-based and performance-task work allows for embedded and authentic assessment, that offers opportunities for students to be motivated by assessment. The CS Principles Performance Tasks are a good example. The Principled Assessment of Computational Thinking and Exploring CS both have assessments built into the curriculum. Because CS is not just programming, assessment should reflect that. Not all assessments should be programming tasks, and instead should include problem-solving tasks that encourage student skills around data, modeling, or algorithm design.

Equity considerations:
Extensive formative assessment can help guide the course facilitation to meet students "where they are". The use of assessments that avoid a single correct answer to a problem allow for different perspectives and understandings.

As this is a nascent area, standards-based exams are still being developed, but teachers should take the lead in evaluating student learning and adjusting their instruction accordingly. Transfer best practices from other disciplines -- CS courses can use similar ways of monitoring achievement during instruction. While CS is not formally part of the California School Dashboard, it could potentially impact other indicators.

Equity considerations:
Do not assume that all students have access to devices or the internet at home. Therefore, make hardware purchases and/or homework decisions accordingly.

4.5 What hardware, software, and broadband connection are necessary?

It is recommended that all K-10 classes get lightweight devices (e.g., Chromebooks) because they are both cheaper and easier to manage. Higher level courses (e.g., AP CS A and some CTE courses) may require greater computing resources. Because so many resources are browser-based, 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. Mice are often important for younger students, too, as some can struggle with the dexterity required of trackpads. It is recommended to use a regular classroom with a modular seating arrangement, since in many cases, traditional lab environments hinder collaboration. It is also helpful to have large table space for collaborative projects. In addition, many lessons on problem-solving using computing are better served "unplugged", needing no equipment at all.

Beyond costs for the physical machinery, consider budgeting for tech support so there is someone on site to troubleshoot technology issues as they arise. It is unfair to expect your CS teacher to also function as the tech support. Issues that come up may require a greater level of tech know-how than some teachers have capacity for in addition to the responsibility of learning CS content, CS pedagogy, and planning, prepping, and implementing lessons. You may also need to build in a budget for software updates, if necessary.

"Be sure to leverage the instructional 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."

5 Preparing and Supporting Teachers

5.1 Who should I hire to teach computer science?

There is no typical resume for a CS teacher. While having a background in CS could be useful, be sure to take a look at the resumes of non-traditional candidates, with past experience in technical or create fields, like math or art. Someone from your current staff could be the right person to take on the challenge, without having to acclimate to a new school culture. Although there are statewide certification requirements (see section 5.2: Who is authorized to teach CS in California?) for who can teach CS, there is also local authority to make temporary exceptions. Permits for these exemptions are for one year and are determined locally. If the exempted instructor needs more time to get a credential, they need to demonstrate that they are making progress to get an extension on the exemption. Check with your Human Resources department to explore what options are available in your district. Once you have a program in place, establish a pipeline that can engage potentially interested teachers in professional development programs, including fast-track programs from local colleges.

5.2 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*** **mathematics** for grades K-9)
- single-subject credential in **business**
- single-subject credential in **industrial and technology education (ITE)**
- **Career Technical Education (CTE)** credential in information technology (IT)

- 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)

Teachers with other credentials may be eligible to teach CS with a temporary waiver or emergency permit. Additional information is available at California's Commission on Teaching Credentialing website.

When CS is part of a CTE pathway, teachers with a CTE with ICT or AME (game design) 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. Teachers can also get a math supplementary authorization (CSETS + methods course). Regardless of teaching authorization, it is strongly suggested that ALL teachers participate in professional development.

"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."

5.3 How can teachers learn to teach computer science?

Identify professional development providers to provide sustained support for your teachers. It is important to identify teachers for professional development that are authorized to teach CS. Microsoft and NCWIT developed a useful computer science professional development guide focusing on equitable computer science education, that lists different providers and what they can offer.

"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."

5.4 I have a teacher who is passionate about bringing computer science to more students. How can s/he effectively advocate?

There are many ways an enthusiastic teacher can be supported. 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.

5.5 How can I get teachers and administrators excited about and engaged in professional development sessions?

When teachers find a strong community and value from professional development sessions, they will continue coming. Be sure to design a continuum of professional development opportunities for a variety of teacher skills and ability, from novice to expert. Design the professional development so that administrators do not have to do a lot of the logistical work; they just need to provide a list of teachers.

"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."

5.6 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 (PLC) with other teachers. Many times CS teachers feel isolated as the lone 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 in-class, there is also a need for collaboration across sites with a PLC. Therefore, there is a greater need for collaboration across sites and 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 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 teachers are compensated for the time they spend learning how to better teach CS, and structure their school day to facilitate their involvement.

"If you can, budget for substitutes for weekday professional development days. Saturdays are easier to implement, but more difficult to get participation and engagement."

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

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 CS.

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.

"We have video vignettes of exemplar models of how teachers address certain standards and how they work in the classroom. We then share them on a learning management system so teachers can see what others do to address these standards."

5.7 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 what their teachers will be teaching. A session for administrators could provide an overview of the foundations of CS, dispel any misconceptions, and demonstrate the versatility and power of CS.

Look for resources like the Strategic CSforALL Resource & Implementation Planning Tool (SCRIPT). 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.

5.8 How can teachers best support students with special needs in computer science classes? What about English Language Learners?

Project TACTIC at the University of Illinois has resources to support students with special needs. Their tips can work with ELL students. They recommend:

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

Work closely with your ELL department, as they are invaluable resources to supporting your English language learner students.

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

- What funding for CS education is in place?
- How are community partners involved?
- What has already been tried?
- What were the challenges faced?
- How can you overcome them?
- How long have they been teaching CS?
- 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?)

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

"Find a time when principals are already meeting and find a way to sneak in some CS into the meeting. This is much easier than trying to add something else to their schedule that is just designed for CS."

5.9 How do we support paraeducators in computer science classrooms?

Paraeducators likely also need professional development and guidance from the CS teacher. Paraeducators sometimes don't have basic tech skills and find CS courses intimidating, so there can be a need for both basic technology professional development, in addition to the CS content professional development. For more information, see the TACTICal teaching brief on paraeducators.

5.10 How do I evaluate computer science teachers?

You should look for similar transferable, good teaching practices that you would see in different subject areas. But, you may notice students working on different projects, using online resources, asking peers how to do things, and teachers not having 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."

6 Funding

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

You may be able to leverage existing funding sources, such as grants in STEM, career pathways, CTE, professional learning, etc., and direct them toward CS education. Other times, local PTA's can help access funds, though this will vary according to the income level of the neighborhood. Asking local industry to support programming is another way to garner additional funds. Programs like TEALS is also another way to get content expertise in the classroom, alongside a CS teacher. See more in Section 7.2: How can I involve local industry?

Some schools find that coding a course as a CTE will help access Perkins funds to pay for the teacher and classroom resources. However, 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 access to CS, regardless of the coding of the course.

"We set up schools with the curriculum and professional development they initially needed and if they wanted more, we provided them with other resources, encouraging them to utilize their own school site funds to explore further."

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 4-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 especially 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 using their LCFF funding for this purpose (California Department of Education, 2019).

In California, there is substantial funding available from the Career Technical Education Incentive Grant Program and K12 Strong Workforce Program for the purpose of strengthening transfer pathways and in particular in the industry sector of Information Community Technologies. See Doing What Matters for Jobs and the Economy and contact your regional Deputy Sector Navigator.

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 partnerships.

"Building community partnerships and community-facing events is just a huge win. We've hosted some 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."

7 Family, Community, and Industry

7.1 How can I get support from the local community for computer science education?

Parents recognize the value of a quality CS 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 parents and community in CS opportunities. Let parents know about the course offerings and encourage them to highlight the importance of all students learning CS. Use these sample emails and announcements to let your community know about your new CS programs. Provide guidance to school counselors and schedulers on why CS is important for all students. Email and recruit administrators and parents to advocate for expanding CS in your school district.

Consider organizing a Family Code Night to engage parents and students in the value of CS education. Approach local technology companies for support and volunteers, including the community in your efforts.

"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."

It is helpful to get the support of the local school board when implementing CS education. The board's commitment is also a way to get CS on the radar for forthcoming Local Control Accountability Plans (see Section 6: Funding). California School Board Association has resources for advocating for CS.

7.2 How can I involve local industry?

While internships for high schoolers can be difficult to come by, they can also be incredibly valuable. Work with local technology employers to develop an appropriate program for students of various skill levels.

Many local businesses recognize the value of CS education and are willing to provide internships to students. 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.

Speak to your district's Linked Learning department in order to work with the connections that already exist. In addition, you can inquire with the Human Resources department of 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 make those connections such as Families in Schools.

"We approached local technology companies to support us with a robotics camp we wanted to sponsor for our students on 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 actually were able to take the laptops home. So coordinating with these different community partners, we were able to get a venue donated and laptops for the kids. On our end, it was just making sure we managed the transportation, the signups, and the logistics of the event. Word gets around then within the community, because you have these community partners that are sharing that story with others, who in turn approach you and say, 'Hey, we've heard you did this thing, how can we help you do something similar?'"

7.3 How can I get volunteers in the classroom?

There are other programs like TEALS and EnCorps that partner with schools to support CS teaching in the classroom with full-time employees who have CS expertise. College students from local institutions can also be useful teaching assistants. Don't rely on volunteers for teaching – the teacher needs to teach and reinforce pedagogy, while volunteers support with CS content knowledge. Attempt to match volunteers to the demographics/backgrounds/language abilities of students. For more, see CSTeachingTips for volunteers.

7.4 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. Dual or concurrent enrollment could be offered at the college, high school, or online. 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 have the incentive of reducing the economic burden of postsecondary education. If students pass, they may be eligible to receive college credit.

"You cannot do this work alone. We have found a lot of value by partnering with university partners (like UCLA), Code.org, ISTE, and similar organizations. We pull from the best they have to offer us, and then we make sure to make it our own, to reflect our schools and our communities."

8 Out-of-School Learning

8.1 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.

Extended exposure and time for deep exploration helps students develop a clear understanding of CS and how it fits into their lives, and out-of-school programs can help with this.

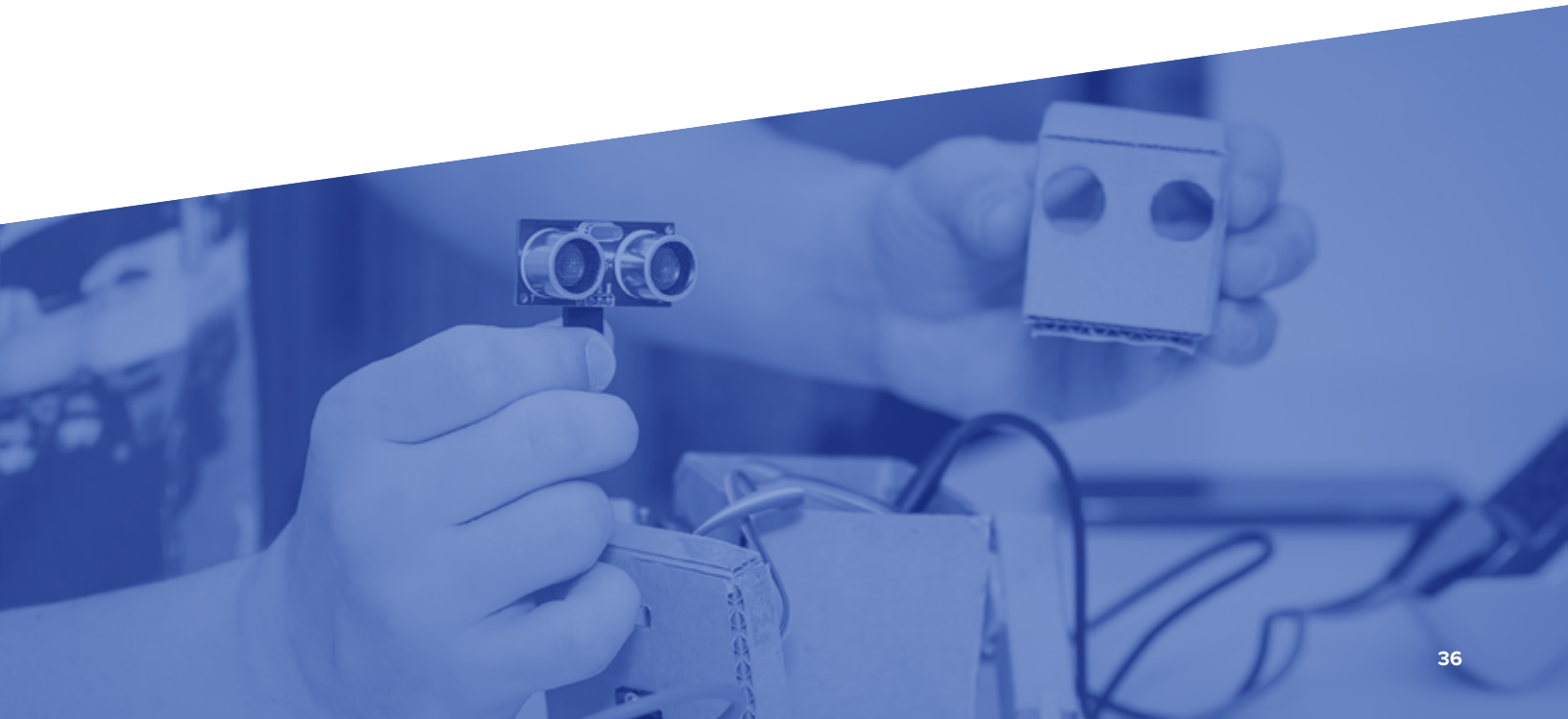
Here are strategies for connecting in-school with out-of-school CS programs:

- Consider how to use the existing infrastructure of your out-of-school time programming and have them work off of in-school learning opportunities.
- Work with nonprofits and NGOs that specifically target underrepresented students (Girls Who Code, Black Girls Code, SMASH Academy, Code Nation, Girl Scouts).
- Provide opportunities for students to work longer on their passion projects (that hopefully they have worked on in school, as well).
- There are various popular robotics programs that are part of large competitive communities available (e.g., FIRST, VEX, Wonder League, PiE), but consider how they address equity before selecting a program e.g, how do they make their program inviting and inclusive, etc.).
- There are several extracurricular programs that have the curriculum written and ready to implement (e.g., CS First).
- Some out-of-school program providers can create connections to mentors.

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

There are various online courses, Massive Online Open Courses (MOOCs), and summer programs that expose students to new programming languages or exciting ways they can use CS. Here are some ideas to explore (Brooks & Elliott, 2017):

- **The Connectory:** This website identifies out-of-school STEM learning opportunities, including computer science. Search by age, opportunity, and topic.
- **Maker Faire:** Maker Faires are a good place for students to experience the creative potential of computer science, 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 creating 3D models.
- **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:** This program is a three-year STEM-intensive residential college prep program for underrepresented students.



9 More Information

9.1 What 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 of **ACCESS**, the state's leading advocate for CS education in California. ACCESS is a statewide network 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, ACCESS 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 **Next Generation Science Standards (NGSS)** explicitly placed Computational Thinking as one of the eight practices necessary for science and engineering.

- The CS community supported the Olson/Buchanan bill (AB1764) in 2014-2015 allowing CS to count toward a 3rd or 4th year math. As of February 1, 2019, CS can also count toward a 3rd or 4th year science.

- 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.

- Explore CS supplementary authorization programs like **San Francisco State University**, **UC Irvine**, **UC Riverside**, and **UC San Diego**.

- Examples of district implementations:

    - **Los Angeles Unified School District** links to the supports they have for teachers interested in CS education.

- **Oakland Unified School District**'s Claire Shorall describes in a piece for Medium how OUSD expanded access to CS for their students.
- **San Francisco Unified School District** provides a lot of helpful information on their site on how they have expanded CS education in their district.

9.2 What other resources should I take a look at?

- **Alliance for Access to Computing Careers** is devoted to increasing the participation of people with disabilities in computing fields.
- **Code.org®** is a nonprofit dedicated to expanding access to CS in schools and increasing participation by women and underrepresented minorities.
- **CS Education Acronyms** can help make meaning of all the acronyms in CS education.
- **CSEdWeek Hour of Code** is an annual program dedicated to inspiring K-12 students to try computer science for one hour. The virtual event is hosted every second week of December in recognition of Grace Hopper's birthday.
- **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.
- **CS Teachers Association (CSTA)** is a membership organization that supports and promotes the teaching of CS. CSTA provides opportunities for K–12 teachers and students to better understand CS and to more successfully prepare themselves to teach and learn.
- **CSTeachingTips** includes a set of CS teaching tips to help teachers anticipate students' difficulties and build upon students' strengths.
- **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 to align efforts with the national vision for CS for all.
- **K-12 CS Frameworks** are 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.
- **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).
- **Scratch** is a free programming software and online community that offers an easy introduction to creative computing. **ScratchJr** is a tablet-based version of Scratch designed for younger children (ages 5-7).
- **University of Chicago's Outliers** provide research and resources for CS implementation 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.

Appendix

Supporting Data

According to 2017 College Board data analyzed by the Kapor Center for Social Impact, total AP CS enrollment was 70% male while females represented only 30% of students, even though they represent half of the student population. Also, while California's students are 60% Latinx and African American, they make up only 24% of AP CS test takers.

Expanding access to CS is important, but so is ensuring high quality engagement and pass rates. The average pass rate for last year's AP CS Principles exams is about 73% but there are still huge racial/ethnic disparities. For example, White/Asian student pass at 85% and Black/Latinx students average 51%.

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 (Google and Gallup, 2016; EdSource-Berkeley IGS Survey, 2017). Nationally, nine in ten parents want their children to learn CS in school, yet only one in four administrators believe parents are demanding it.

Listed Links

FOREWORD

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

CSforCA
http://access-ca.org/csforca

1 INTRODUCTION

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

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

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

Goode, Flapan, and Margolis (2018)
https://books.google.com/books?id=F9lLDwAAQBAJ&lpg=PP1&pg=PA45#v=onepage&q&f=false

Ladson-Billings (1995)
https://www.jstor.org/stable/1163320?seq=1#metadata_info_tab_contents

Sleeter (2012)
https://s3.amazonaws.com/academia.edu.documents/35568511/2012__Confront...CRP.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1553960188&Signature=GdUDzeatNEnb2d9jvoEkslfAhO4%3D&response-content-disposition=inline%3B%20filename%3DConfronting_the_marginalization_of_cultu.pdf

San Francisco Unified School District
https://www.edjoin.org/JobDescriptions/688/2018%20-19%20SY%20Supervisor%20Computer%20Science-20180928095228.pdf

Chicago Public Schools
https://www.topschooljobs.org/job/703921/computer-science-curriculum-instruction-manager

2 DEVELOPING PATHWAYS

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

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

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

ACCESS infographic on teacher credentialing
http://access-ca.org/infographic-building-a-robust-cs-education-teaching-force/

Next Generation Science Standards (NGSS)
https://ngss.nsta.org

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

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

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

SFUSD's current draft for a redesigned middle grades learning experience
https://mgredesign.sfusd.edu/redesign-model

University of California Curriculum Integration (UCCI)
https://ucci.ucop.edu

Creative Computing Curriculum Guide
http://scratched.gse.harvard.edu/guide

Scratch
https://scratch.mit.edu

Computer Science Discoveries (CSD)
https://code.org/educate/csd

Exploring Computer Science (ECS)
http://www.exploringcs.org

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

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

Bootstrap
https://www.bootstrapworld.org

Project Lead the Way
https://www.pltw.org

C-STEM
http://c-stem.ucdavis.edu

Google's CS First
https://csfirst.withgoogle.com/s/en/home

Beauty and Joy of Computing
https://bjc.berkeley.edu

9 Dots
https://www.9dots.org

Mobile CSP
https://course.mobilecsp.org/mobilecsp/course?use_last_location=true

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

Code.org
https://code.org

Code.org's list of other curricula
https://code.org/educate/curriculum/3rd-party

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

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

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

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

CSTA professional development opportunities
https://www.csteachers.org

TEALS
https://www.tealsk12.org

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

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

3 STUDENTS AND RECRUITMENT

NCWIT's Counselors for Computing
www.ncwit.org

CS Recruitment Toolkit - Microsoft
https://www.microsoft.com/en-us/digital-skills/learn-how/schools

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

CS Unplugged
https://csunplugged.org

Ladson-Billings (1995)
https://www.jstor.org/stable/1163320?seq=1#metadata_info_tab_contents

Sleeter (2012)
https://s3.amazonaws.com/academia.edu.documents/35568511/2012__Confront...CRP.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1553960188&Signature=GdUDzeatNEnb2d9jvoEkslfAhO4%3D&response-content-disposition=inline%3B%20filename%3DConfronting_the_marginalization_of_cultu.pdf

NEA Online Resources for Culturally Responsive Teachers
http://www.nea.org/archive/16723.htm

ECS Teacher Practices
http://www.exploringcs.org/for-researchers-policymakers/reports/ecs-teacher-practices-research

CS Discoveries Curriculum Guide
https://curriculum.code.org/csd-18

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

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

Exploring Computer Science (ECS)
http://www.exploringcs.org

CS Unplugged
https://csunplugged.org

5 PREPARING AND SUPPORTING TEACHERS

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

California Supplemental authorization in CS (via SFUSD)
https://www.csinsf.org/supplemental-authorization.html

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

Microsoft and NCWIT's computer science professional development guide
https://www.ncwit.org/sites/default/files/resources/cs_professional_development_guide.pdf

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
http://scratched.gse.harvard.edu

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

Project TACTIC TACTICal briefs for students with disabilities
https://ctrl.education.illinois.edu/TACTICal/images/librariesprovider17/default-album/0-7

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

Example District Overview Tool
https://docs.google.com/document/d/1zYhuA1lquEV_L08NvPEqsq3QTxNQWAPoly1Nc4kR4Fc

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.tealsk12.org

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

California Department of Education, 2019
https://www.cde.ca.gov/fg/aa/lc/lcffoverview.asp

Career Technical Education Incentive Grant Program
https://www.cde.ca.gov/fg/fo/r17/cteig15ins.asp

K12 Strong Workforce Program
http://doingwhatmatters.cccco.edu/StrongWorkforce/K12SWPOverview.aspx

Doing What Matters for Jobs and the Economy
http://doingwhatmatters.cccco.edu

7 FAMILY, COMMUNITY, AND INDUSTRY

Nine in ten parents want their children to learn CS - Google and Gallup, 2016
http://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf

Code.org's sample emails and announcements to let your community know about your new CS programs
https://code.org/educate/resources/recruit#blurbs

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

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

California School Board Association
http://csba.org

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

8 OUT-OF-SCHOOL LEARNING

Brooks & Elliott, 2017
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/explore

The Clubhouse Network
https://theclubhousenetwork.org

Technovation
https://technovationchallenge.org

SMASH Academy
https://www.smash.org

9 MORE INFORMATION

CSforCA
http://access-ca.org/csforca

ACCESS
http://access-ca.org

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

Next Generation Science Standards 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

CS Supplemental Authorization Programs:
San Francisco State University
http://www.sfsu.edu

UC Irvine
https://sites.uci.edu/cs1c/cs1catoc-teacher-certificate-program

UC Riverside
https://www.extension.ucr.edu/certificates/19013519/educationandcredentials/subjectmatterspecialization/computerscienceeducation

UC San Diego
http://csteachers.ucsd.edu

Examples of District Implementations:
Los Angeles Unified School District
https://achieve.lausd.net/page/10023

Oakland Unified School District
https://link.medium.com/PL6AyRKBXT

San Francisco Unified School District
https://www.csinsf.org

The Alliance for Access to Computing Careers
https://doit-prod.s.uw.edu/accesscomputing

Code.org
https://code.org

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

CS Ed Week
https://csedweek.org

CSforALL
https://www.csforall.org

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

CSTeachingTips
http://csteachingtips.org

Expanding Computing Education Pathways Alliance
https://ecepalliance.org

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

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

National Girls Collaborative Project
https://ngcproject.org

Scratch
https://scratch.mit.edu

ScratchJr
https://www.scratchjr.org

University of Chicago's Outliers
http://outlier.uchicago.edu/basics

APPENDIX

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

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

References

Brooks, C. & Elliott, K. (2017). Why Computer Science Education? A Toolkit for Parents. Retrieved from https://www.edc.org/why-computer-science-education-toolkit-parents.

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

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., 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.

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. Retrieved from http://www.exploringcs.org/wp-content/uploads/2014/04/SeeingMyselfArticle.pdf

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.

U.S. Census Bureau, 2013-2017. (2017). American Community Survey 5-Year Estimates. Retrieved from https://factfinder.census.gov.

Acknowledgments

This guide was written by:

Julie Flapan, EdD, UCLA Center X
Roxana Hadad, UCLA Center X
Claire Shorall, formerly Oakland Unified School District
Bryan Twarek, San Francisco Unified School District

With

Jared Amalong, Sacramento County of Education
Dawn Guest-Johnson, Los Angeles Unified School District
Jessie Gurbada, Riverside Unified School District
Steve Kong, Riverside Unified School District
Dylan Lira, EdD, Compton Unified School District
Sophia Mendoza, Los Angeles Unified School District

Design and layout by:

Michelle Choi

For more information:

info@csforca.org

Julie Flapan
Executive Director, ACCESS and CSforCA
Computer Science Equity Project
UCLA Center X
1320 Moore Hall, Box 951521
Los Angeles, CA 90095-1521

This material is based upon work supported by the National Science Foundation under Grant # 1837780. Thank you to Code.org for additional financial support for this guide.