A project of:

UCLA

PLUGGING INTO POWER
FAMILY AND COMMUNITY ENGAGEMENT FOR EQUITABLE CS EDUCATION

Plugging into Power

A Guide for and by Parents to Advance Computer Science Education

Why <u>ALL</u> kids should learn Computer Science and what adults can do to help

Dear Parents, Caregivers, and Community Members,

We all want the best for the young people in our lives—our children, grandchildren, nieces and nephews, cousins, neighbors, friends, and students—especially when it comes to preparing them for the future. But with all the technological changes and advancements happening today in Computer Science (CS) and Artificial Intelligence (AI), it can be challenging and overwhelming to know where to start.

We created this guide to empower parents, grandparents, educators, community members and other adults who care about the growth and well-being of young people with the knowledge and tools to help children explore and experience learning CS in and out of schools. A group of parents, caregivers, and community advocates from diverse backgrounds collaborated with researchers from the UCLA School of Education & Information Studies to codevelop this resource. The guide breaks down what CS is (and isn't!); why it's critical to learn about CS, what ethics have to do with computing; and what steps adults can take at home, school, and beyond to ensure that the young people in their life have high-quality CS learning opportunities and gain the skills needed to become informed consumers and creators with technology—whether they become programmers or not!

Thank you for your interest in learning more about CS and why it's foundational to prepare young people for college, careers, and engagement in their communities. We also want to thank all the parents, caregivers, and community members whose perspectives, voices, and experiences inspired and contributed to this guide, as well as the educators, nonprofit leaders, and community members who provided valuable feedback. Finally, we would like to thank the Siegel Family Endowment (SFE) for their generous financial support and contributions to this work.

Sincerely,

The Computer Science Equity Project Team
UCLA Center X

What's Inside

What is Computer Science and why is it important?

Computer Science touches everything in our lives...

There's no denying that computers and technology have become an integral part of how we live (think cell phones and online banking), work (think email and video conferencing), and play (think video games and online shopping), and can even influence our decisions (think digital ads, social media influencers, and recommender systems). This is why it's important that ALL people learn about Computer Science (CS) and how it affects the world and their lives. Especially, how can we learn to use computing tools to help and not harm?

...and impacts every career industry.

CS is used in every job field including medicine, music, finance, fashion, farming, sports, politics, entertainment, law, and so much more! And Artificial Intelligence (AI), a part of CS, is also impacting all fields. For example, doctors use computers and AI to diagnose patients or perform surgery, fashion designers create dresses using computer programs, and sports teams analyze game data to optimize performance. No matter what jobs are in the future, it's a pretty safe bet that CS will be an important skill to have.

And we're not talking about coding!

CS goes way beyond coding and software development. It's not only for kids who want to become programmers or work in tech. Just like learning to read a book or write an essay teaches kids important lifelong literacy skills, Computer Science can provide another way for kids to express themselves, explore new ideas, or pursue the things they care about whether it's helping those in need, making art, educating others, or inventing something entirely new!

"Computer Science is the study of computers and ALL the phenomena that arise around them."

– Herbert Simon

✓ **CS is...**	✗ **CS is NOT...**
Solving real-world problems using computers	Learning to type/keyboard or use word processing software (like Microsoft Word) -- often called "educational technology" in school
Reading and writing (designing) computer programs	Creating spreadsheets (data entry) -- often called "digital literacy" in school
Reasoning about what computers can and can't do	Using social media
Useful for every career and industry	Only for programmers or tech workers
Creating video games or apps	Playing video games or using apps
A subject anyone can (and should) learn	Only for "smart" people or "nerds"
Important learning for all grades K-12	Only for high schoolers or people "good at" math or students who want to work in tech
Foundational to how AI is created and used	Separate from AI
for everyone!	Only for people of certain races, backgrounds, abilities, or gender.

"Computer literacy is one of the major prerequisites in almost every career. And I think it's so important for everybody to have those basics in order to have a successful career and future."

– Sandra, Mother of a 6th grader and an 8th grader

PLUGGING INTO POWER
FAMILY AND COMMUNITY ENGAGEMENT FOR EQUITABLE CS EDUCATION

What is Artificial Intelligence (AI) and what does it have to do with Computer Science?

What is Artificial Intelligence (AI)?

AI is technology that involves computers performing complex decision making, data analysis, and creative tasks that are usually done by humans. AI works by following a set of instructions (algorithms) to analyze large amounts of information (data sets), look for patterns, and then make decisions or predictions based on that analysis.

Did you know that AI is CS?

Computer Science (CS) is a broad field of study and industry that includes the study of computers, algorithms, and much more, while AI is just one branch of CS. Learning CS provides the foundational knowledge and tools for developing AI systems, so understanding and learning CS will help you better understand and learn about AI. Even as some Generative AI—or GenAI—tools are able to code and seemingly do the work of a computer scientist, people still need to know CS to help design, guide, evaluate, and make sure that AI and other tech are created and used with the greater good in mind.

Is AI the same as GenAI?

AI refers to the broader category of technologies that analyze past information or data to mimic activities performed by human intelligence as described above. You probably use AI in your daily life, even without knowing it! For example, digital assistants such as Siri and Alexa can understand your voice commands and answer questions, recommendation systems used on Netflix or Amazon suggest shows or products based on your past searches, and social media like Instagram or TikTok filter content by what you last clicked on. On the other hand, GenAI goes beyond this to generate new text, audio, images, or videos. GenAI tools like ChatGPT are more complex and does not only search the Internet to find information. This type of GenAI called Large Language Models (LLMs) are "trained" on large amounts of data and learn patterns from that data in order to generate new text and images, or answer questions in ways that look or sound natural. So, for example, GenAI can create text for a new book based on reading and analyzing 10 novels written by humans in different languages. Or GenAI can use legal information about apartment rental contracts in one city and then develop a unique rental agreement for a different apartment in another city. However, GenAI does not always do things perfectly and thus needs to be checked for accuracy and misinformation. GenAI also requires extreme amounts of energy, which contributes significantly to global climate issues. Read more about the ethical concerns of AI on the next page.

80% of parents have concerns about harmful effects of AI, but only 35% of parents have discussed AI usage with their kids.

– National University

What does ethics have to do with AI?

As AI changes the way we engage with the world and each other, we need to think ethically and critically about AI to better understanding how to use it responsibly. Common concerns with AI include:

- **Bias and Inaccuracy**: The results provided by AI—to answer questions, create images, solve problems, etc.—are highly dependent on the data or information fed to it, which is why AI can provide incorrect and biased answers.
- **Environmental Impacts:** AI data processing centers generate a lot of heat and water is used to cool those centers. This means AI has a big negative impact on the environment, using up fresh water resources worldwide without that water returning to the water table.
- **Data Privacy**: The information people share with AI chatbots or programs may be stored and used to train AI or for other purposes without the user knowing if or how their personal data is being saved and used.
- **Ethics:** AI can be used for harmful purposes or have hurtful consequences, such as cyberbullying, generating fake images or "deepfakes" that impact elections and politics, or using artists' or authors' work without their consent, compensation, or acknowledgment.

How is AI used in schools?

There are mixed feelings on AI use in schools. Some educators are making lesson plans with AI, letting students use AI to conduct research, or using AI tutors to provide individualized lessons. Other teachers fear AI is being misused by students to write essays or do assignments for them. AI use in education offers opportunities for efficiency and personalized feedback, but there are concerns that AI could negatively impact learner agency, creative learning opportunities, and interpersonal connections. It's crucial that kids learning about AI/CS should include having discussions about the ethics of tech too. To ensure that AI is treated responsibly at your child's school:

- Ask your school how or if AI is integrated into the school curriculum.
- Join or form a parent group to push for better understanding of AI and education for parents.
- If AI is integrated in the school curriculum, make sure the content is culturally responsive and centers around issues of bias, privacy, and values.
- Find out if teachers, principals, and administrators receive training in ethical AI/technology practices.

PLUGGING INTO POWER
FAMILY AND COMMUNITY ENGAGEMENT FOR EQUITABLE CS EDUCATION

Why do I want my child to learn Computer Science?

What is the value of learning Computer Science (CS)?

People can have different reasons for wanting kids to learn CS—and there's no one right answer! For example, you may want to help your child secure a good job in the future, gain digital literacy skills, be an informed member of our technology-driven society, learn about AI and it's impact on the world, or inspire more girls and people of color to participate in STEM fields. We all want children to grow up to have a healthy, happy, and meaningful life, and to be kind people who help one another. How can your child's engagement with technology support this vision of their future or hinder it?

What do you think?

Why do you want the children in your life to learn CS? Are there other reasons beyond what we have shared so far? When people talk about CS education, they tend to draw on some combination of the seven core values* shown below that describe the impacts CS will or could have. Which of the following connect to your reasons for CS?

Equity & Social Justice

Competencies & Literacies

Citizenship & Civic Engagement

Technological, Social & Scientific Innovation

Economic & Workforce Development

School Reform & Improvement

Personal Agency, Joy & Fulfillment

*Santo, R., Vogel, S., & Ching, D. (2019). CS for What? Diverse Visions of CS Education in Practice. New York, NY: CSforALL.

Define your priorities

As you help shape the educational journey of your child, it may be useful to define what you think priorities in their Computer Science (CS) education should be. To support this process, below are value statements adapted from the "CS Visions" activity* created by Rafi Santo, Sara Vogel, and Dixie Ching to help jumpstart your priorities list. This way, you will be better positioned to advocate for the CS education you believe will most benefit the children in your life. You can find the original activity cards here, an online quiz here, and an article explaining the purpose of the activity here.

Kids should learn CS because...

- Knowing how to code is a new form of literacy.
- Computing provides youth with the ability to express themselves creatively and have voice.
- Collaboration on CS projects can lead to meaningful relationships between students/educators.
- Creating new technologies like apps, websites, or robots is fun!
- There are major disparities in minorities', young women's, and rural youth's engagement in STEM fields, and universal CS education is part of addressing that.
- It will level the playing field and help close the "digital divide" and "participation gap" around tech for lower-income youth.
- Computing may provide our youth with more and better career opportunities to choose from.
- Our technology is largely designed by economically, racially and socially privileged groups, and their biases and blind spots get embedded in tech. Increased access to CS education can help.
- Not all tech will be in the best interest of our students—they'll need to be able to think critically about technology platforms. It's a "program or be programmed" world out there!
- Computational thinking will be key no matter what career youth end up in.
- Practices from CS can enhance student learning of traditional academic subjects (introducing computer modeling to learn ecology concepts, or using CS concepts to learn algebra).
- CS education often uses project-based learning approaches that can enhance school pedagogy and move away from "sage on the stage" approaches.
- Political and cultural participation are increasingly shaped by computing and our students need to understand the social impacts of tech.
- Youth shouldn't just be consumers but also producers of technology.
- Informed citizens need to understand the basics of how the technological world works in order to contribute productively to society as a whole.
- The more people that understand CS, the more innovations and new knowledge we can produce as a society that solve "wicked" problems such as climate change or cybersecurity.

*Santo, R., Vogel, S., & Ching, D. (2019). CS for What? Diverse Visions of CS Education in Practice. New York, NY: CSforALL.

PLUGGING INTO POWER
FAMILY AND COMMUNITY ENGAGEMENT FOR EQUITABLE CS EDUCATION

Why should ALL kids learn Computer Science?

Computer Science teaches important life skills...

Learning Computer Science (CS) helps young people develop "computational thinking" skills that are also important life skills such as critical thinking, problem solving, decision making, and perseverance. Computational thinking skills are also useful for other academic subjects. For example, CS involves solving problems by looking for patterns or breaking problems into smaller, more manageable parts. These same skills of recognizing patterns and systems thinking are used in Math, Social Studies, Science, and Language Arts.

...that prepare kids for college, career, and beyond.

Computer science is for everyone, not just programmers or "techies." People have used their CS education to pursue passions outside of tech or to address key problems in society. CS learning provides opportunities for project-based learning, real-world problem solving, and entrepreneurship, while helping students develop critical reasoning, communication, and collaboration skills beneficial to all children's personal development.

All kids should know about the good AND bad of tech.

Learning CS demystifies technology and prepares kids to become well-informed consumers and creators, able to think critically about how computers can both help and harm. Without knowing about the potential dangers, kids may become susceptible to misinformation, scams, cybersecurity, etc.

Can computers be biased?

Technology, and specifically computer technology, can be incredibly helpful. But did you know that it can also be hurtful? For example, in 2017 a viral video showed an automatic soap dispenser that failed to dispense any product onto his dark skin, while a white coworker who used the same dispenser had no problems. This example shows how technology can have negative consequences depending on how it is designed and who is designing it. The same is true for algorithms designed for computer apps used in hospitals or banks, or in the AI we use today. Depending on what dataset is used to train AI and who is designing it, we may see very different results (see, for example, Joy Buolamwini's research on current AI facial recognition tools that had less than 1% error rate for light-skinned men but above 30% error rate for dark-skinned women). With technology rapidly evolving, it is crucial that future tech creators reflect the gender and race of all its users. Yet, a 2024 survey shows that only 12% and 16% of all tech workers are Black or Latine and only 18% of CS degrees are held by women. This is why it's essential that all kids learn CS, and not just a select few.

In the US, only 6% of high school students ever take a Computer Science class.

Why isn't Computer Science (CS) required learning in school?

Unlike core subjects like math or writing, CS is not a required K-12 course in most states, which means that many kids don't learn CS and many schools don't offer a CS class taught by a well-prepared teacher. And even if schools have computers or iPads, that doesn't mean kids are learning CS. For example, some schools only use computers to teach typing, do math drills, or to take computer-based tests. Even worse, in schools where CS courses are available, there are racial and gender disparities in who takes those CS classes. Sometimes we have pre-conceived notions about who should take CS, and then only certain students are encouraged to study CS, so it's important to challenge those stereotypes about what a computer scientist looks like. Other reasons might have more to do with students' course schedules, or some schools don't even offer CS courses at all. CS and AI are emerging career fields that lead to high-paying and high-demand jobs that can address socio-economic inequality and empower youth. More importantly, it is imperative that all students learn CS to become critical users and thoughtful creators of technology. Learn more about these issues shared above and why so few girls, African Americans, and Latine are learning CS by reading the book *Stuck in the Shallow End* and graphic novel *Power On*! (links below).

Learn more:

- **Computational Thinking for Parents and Families (handout)**
  csforca.org/computational-thinking-for-parents-and-families
- **In the Age of AI, California Students Urgently Need Access to CS (EdSource article)**
  edsource.org/2025/age-of-ai-california-students-urgently-need-access-to-computer-science/727237
- **State of Computer Science Education (report)**
  advocacy.code.org/stateofcs
- **Stuck in the Shallow End (book)**
  mitpress.mit.edu/9780262533461/stuck-in-the-shallow-end
- **Power On! (graphic novel available in English, en Español (¡Conectados!), 한국어, & audiobook)**
  poweronbook.com

Sparking conversations about Computer Science and AI

It's important to talk to kids about computers.

What does your child like to do with technology? Do they have a phone or computer at home, or use devices at school? What exactly are they doing with that technology, and more importantly, do they understand important ethical concerns behind the screen? For example, how is their technology use affecting other people around them? How is their personal information being recorded and used against them? Use the questions provided on the next page to spark your child's deeper thinking about Computer Science (CS) and the role of tech in their lives.

Let them hear it from you!

Not everyone relates to the images and stereotypes of who does CS and STEM. You can play an important role in breaking stereotypes and clarifying misunderstandings by talking to your child about tech starting early on and encouraging them to take CS courses. If you learned CS, share your experiences – the good and the bad – and how you value that learning today. If you didn't get to learn CS, share why you want your children to learn what you didn't have the chance to.

Talk to your school and other parents.

Ask your teacher or school leader to find out what your child is learning about CS at school already, or ask how they can offer more CS learning opportunities if there are none. Talk to other parents or community members about how they approach CS conversations with their children, get tips on finding high-quality CS classes outside of school, or plan a meeting in your community to discuss how to support kids learning CS in your area.

"I think that we as parents also have to prepare ourselves more to better understand computing and to be able to talk to our children, having a conversation with them about computers, so that they are also interested."
– Iris, Mother of a 9-year-old

Sparking Conversations about Computer Science (CS)

Talk to your child to better understand what they know about CS, even if they take CS classes. You can clear up misconceptions, discuss CS career pathways, and ask critical questions about the good (and bad) of tech. And if they don't take CS already, encourage them to try it out!

Using Computers

- What do you like to do with computers (e.g., computer, phone, smartwatch, etc.)?
- Where do you see technology in our everyday life?
- If you could do anything with technology, what would that be?
- What does a computer do? What is a code/coding?
- What kinds of career industries use Computer Science? (HINT: ALL OF THEM!)
- What kinds of careers are you interested in pursuing? How will CS be useful for that job field?
- How does technology help people? How can technology be harmful?
- When you use computers, your cell phone, AI, etc., how are your actions impacting others? And how might your private data and information be recorded and used as well?

Ethics of Computing

- What do you envision computers doing in the future?
- What should we be careful about computers doing in the future?
- What is artificial intelligence (AI)?
- Do you think robots can be biased? How?
- How do you think human values, beliefs, and biases get built into the computing tools we use today? (for example: AI, cell phone apps, internet search engines, etc.)
- What does it mean to be inclusive of all people in technology/CS? Why is this important?
- Should technologists be responsible for their creations? Why? What happens if they are not?

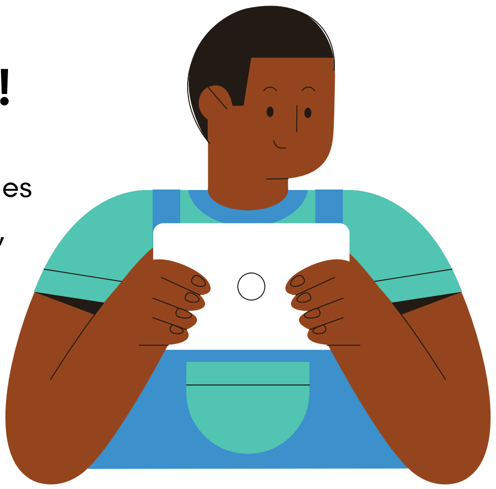
Technology in School

- How do you use technology/computers at school?*
- What CS courses are available at school? Who's taking them or not? Why do you think that is?
- If your child is taking CS: What are you learning about CS? What do you like/dislike about your CS class? What projects do you enjoy making? What do you want to learn more about?
- If your child is not taking CS: Does your school offer CS? Have you thought about taking it? Why or why not? [If your child is in high school in California, for example, you could suggest: Let's see if it counts toward A-G, graduation and toward your college eligibility/application.]
- If not at school, what are other ways or places to learn about CS that you're interested in?

*Keep in mind that even if your child has access to computers in school or takes CS classes, it doesn't necessarily mean they are learning "CS." Access to and quality of CS learning can vary greatly depending on the school, grade, curriculum, or staff available. For example, students have computers but only to learn typing; or a class uses unplugged activities to teach computational thinking skills without any devices; or in a school with no CS classes a history teacher teaches coding and students create interactive timelines or video games about historical events. This is why it's valuable to talk to your child or school directly.

PLUGGING INTO POWER
FAMILY AND COMMUNITY ENGAGEMENT FOR EQUITABLE CS EDUCATION

Helping your child learn Computer Science



YOU can help kids learn Computer Science!

You don't need to be a tech expert to support your child's Computer Science (CS) learning journey. There are many "unplugged" CS activities that introduce CS concepts and practices through hands-on activities, including ones that don't even need a computer. Learn more on the next page which provides a list of **Getting Started with Computer Science** resources organized by age range.

Put on your student hat.

The best teachers are lifelong learners, so put on your student hat! The most important thing to remember is that you don't need to know all the answers. You will be helping your child learn key CS practices such as thinking creatively, problem solving, and working collaboratively, by modeling for your child that it's okay to make mistakes and not have all the answers.

Let's talk about Computer Science.

Sometimes simply talking with your child about computing can make all the difference: it can be the first step in helping your child learn about CS. Better yet, you don't need a device or computer to start conversations with your student about what CS is and why it's important to learn, or to encourage them to try a CS course at school. Students sometimes don't see themselves doing CS based on who they've seen doing it—and those images might not look like them, especially for girls and students of color. In spite of this lack of representation, it's important to ask why that might be the case. By talking to your child, you can help them see they can do anything and be anything. Also, having critical discussions about the ethics of computing can support your child learning about both the good and potential harms of technology. See the **Sparking Conversations about Computer Science** section to find example questions to ask yourself, your child, and your school about technology, what is (and isn't) CS, and why it's important for all kids to learn.

Getting Started with Computer Science (CS)

There are many resources out there to help you support your child learning CS, and here are a few of our favorites to get you started. Keep in mind that the best learning experiences aren't necessarily the most flashy ones, but rather, ones that will connect to your child's interests or are personally relevant to their community and lives. Consider how your child can take what they learn with CS and apply it to what they're passionate about.

Young Children (Ages 4-7)

Early exposure can teach computational thinking (or how to think like a computer) and help young children have a healthy relationship with technology, so they see it as a creative medium rather than just for watching videos or playing games. Learning can start at home with books, coding apps designed for younger learners, or "unplugged" activities that don't require any devices.

- **Computational Thinking for Parents and Families handout:** csforca.org/computational-thinking-for-parents-and-families
- **CS Unplugged website:** csunplugged.org
- **How to Code a Sandcastle book:** joshfunkbooks.com/how-to-code-a-sandcastle
- **Hello Ruby book:** helloruby.com
- **Scratch Jr. iPad app:** scratchjr.org

Elementary and Middle School (Ages 8-12)

This is a popular age to introduce kids to coding. Establish your goals: Do you want to expose them to computing for their general education? Do they already have an interest in computing or STEM? Or see the **Why do I want my child to learn Computer Science?** section for more. Consider what offerings in school or out of school would best fit your child's interests, learning style, and comfort level: designing a video game, joining an all-girls coding club, or doing online activities at home.

- **Six Computer Science Games and Activities to Entertain Your Kids:** csforca.org/six-computer-science-games-and-activities-to-entertain-your-kids
- **Scratch programming for kids**: scratch.mit.edu
- **Micro:bit programmable device**: microbit.org
- **The Clubhouse Network out-of-school clubs**: theclubhousenetwork.org
- **Girls Who Code clubs:** girlswhocode.com

Teens and High School (Ages 13-18)

Talk to your teen about CS, even if they are already taking classes. Learn about what interests them, discuss CS careers or Career and Technical Education (CTE) pathways, and have important discussions about the promises and perils of tech and AI. How can they apply they learning to their interests, futures, and community?

- **Python programming:** python.org
- **First Robotics club and competition**: firstinspires.org
- **MakerFaire events**: makerfaire.com
- **Power On! graphic novel:** poweronbook.com
- **Coded Bias documentary about biased AI:** codedbias.com

How to advocate for Computer Science and AI education

90% of parents want their children to learn CS or agree that schools should offer CS. But only 25% of principals say their schools offer programming/coding classes.

- Gallup, 2016.

Awaken your inner advocate.

Many kids don't learn Computer Science (CS) and many schools don't offer classes because in many states CS is not a required course in K-12 public education. You can help change that and make CS available to all kids! Parents' opinions matter to schools, so know that you hold a lot of power when it comes to advocating for a better education for your children. The following page provides ideas for what you can do to advocate for more CS and AI learning opportunities at your school, in your community, and at the policy level.

You are not alone!

Fellow parents, caregivers, and community members can become champions for CS too! We all want the best education for the young people in our lives and agree that schools should ensure all children receive a quality CS and AI education to be prepared for college, careers, and beyond. Some parents are already active volunteers at school or serve a special cause in the community as a parent leader, PTA member, community advocate, promotora/o, etc. Learning about the need for greater access to CS and AI education can inform and contribute to these ongoing advocacy efforts. Parents like to hear and learn from other parents, so as a trusted messenger, you can do your part to share what you know.

Advocate for Computer Science (CS) & AI Education

First, find out what CS/AI learning experiences are offered at your school using the **CSforCA Equity Data Dashboard** or by inquiring with your teacher or principal.

at your school

- If there are CS classes, find out how accessible are they to all students:
  - How many classes are there? Do they fulfill any graduation or college requirements?
  - Who typically takes them and why? How do students find out about them and are they encouraged (or not) to take them? How can the school work to ensure all students learn CS?
  - Does the curriculum include real-world examples, AI, and ethics in computing?
  - What projects are done and how do they connect to students' lives and interests?
  - How are teachers prepared and supported?
- If there are no classes, consider what is needed to ensure all students learn CS:
  - What ways can CS be incorporated in the curriculum or offered outside of class time?
  - What resources are lacking: teachers, technology, time, funding, district support?
  - How can teachers be equipped and comfortable teaching about technology?
- Schedule time to talk with your principal or district leader to make your case for more CS education. Visit **csforca.org/take-action/#make-the-case** for advocacy tools such as a sample letter template, slidedeck, social media cards, **K-12 CS Standards**, **K-12 CS Framework**, **TeachAI**, and **CS Equity Guide** for implementing CS learning in schools. See a sample letter in the **Appendix**.
- If a CS course can't be offered, brainstorm ideas to incorporate CS aside from classes, such as: host a **Family Code Night** for families to learn about CS together, invite CS professionals or relevant alumni to speak at Career Week, ask a local high school robotics club to host an afterschool club for elementary students, ask a teacher to host a lunchtime or afterschool **Girls Who Code** club.
- Attend your local school board, School Site Council (SSC), English Learner Advisory Committee (ELAC), and District English Learner Advisory Committee (DELAC) meetings, and provide the data on your school compared to others to bring attention to the need for more access to CS.
- Ask about funding for classes or resources through your Local Control and Accountability Plan (LCAP) program, or research grants/sponsorships from local companies and nonprofit organizations.

in your community

- Contact local industry, nonprofits, out-of-school/informal learning organizations, NGOs, and colleges/universities to inquire if they can offer CS learning programs for kids, organize field trips, provide funding, donate prizes for fundraising efforts, sponsor or provide space for an event/family code night/hackathon, offer internships/mentorships/job shadowing, host a computing club, mentor a class, or provide professional development for teachers.

at the policy level

- Look for local- or state-level initiatives that support CS education at the policy level. For example, CSforCA is a statewide coalition of parents, teachers, school leaders, administrators, researchers, and policymakers all working towards bringing CS to all students in California. Sign up for the newsletter at **csforca.org** to get the latest CS education news and opportunities.

Key Statistics about Computer Science (CS) Education

The following key statistics can help increase awareness about the critical need for all children to learn about CS and the ethics of AI. Share these with your child, parents/caregivers, school leaders, and community advocates. The statistics are from the State of CS Education Report 2025 and code.org.

In the U.S., only **6%** of high school students ever take a Computer Science class.

U.S. Hispanic/Latine students are **1.7 times** less likely than their white and Asian peers to take CS, even if their school offers it.

CS education elevates students' career opportunities and their lifetime earning potential, allowing them to uplift their families. But only **38%** of students enrolled in CS classes are economically disadvantaged.

Students who take an AP Computer Science course are **17%** more likely to go to college.

90% of parents want their children to learn CS or agree that schools should offer CS. But only **25%** of principals say their schools offer programming/coding classes.

Only **60%** of U.S. high schools teach CS and it is least likely to be taught in rural schools or schools with a high number of students of color.

High schools with more than **50%** of their students qualifying for free or reduced lunch are less likely to offer foundational Computer Science classes.

Appendix

Resources provided in this guide:

Plugging into Power: pluggingintopower.org
UCLA Center X Computer Science Equity Project:
centerx.gseis.ucla.edu/computer-science-equity-project
Computational Thinking for Parents and Families (handout)
csforca.org/computational-thinking-for-parents-and-families
In the Age of AI, California Students Urgently Need Access to CS (EdSource article)
edsource.org/2025/age-of-ai-california-students-urgently-need-access-to-computer-science/727237
Stuck in the Shallow End (book)
mitpress.mit.edu/9780262533461/stuck-in-the-shallow-end
Power On! (graphic novel also available en Español (*¡Conectados!*), 한국어, and audiobook:
poweronbook.com
CS Unplugged website: csunplugged.org
How to Code a Sandcastle book: joshfunkbooks.com/how-to-code-a-sandcastle
Hello Ruby book: helloruby.com
Scratch Jr. iPad app: scratchjr.org
Six Computer Science Games and Activities to Entertain Your Kids:
csforca.org/six-computer-science-games-and-activities-to-entertain-your-kids
Scratch programming for kids: scratch.mit.edu
Micro:bit programmable device: microbit.org
The Clubhouse Network out-of-school clubs: theclubhousenetwork.org
Girls Who Code clubs: girlswhocode.com
Python programming: python.org
First Robotics club and competition: firstinspires.org
MakerFaire maker events: makerfaire.com
Coded Bias documentary about biased AI: codedbias.com
CSforCA Data Dashboard: csforca.org/the-data
CSforCA Advocacy tools: csforca.org/take-action/#make-the-case
K-12 Computer Science Standards: csteachers.org/k12standards
K-12 Computer Science Framework: k12cs.org
CS Equity Guide: csforca.org/csequityguide
TeachAI: teachai.org
Family Code Night: familycreativelearning.org
Seasons of CS: seasonsofcs.org
CSforCA: csforca.org
State of CS Education Report 2025: advocacy.code.org/stateofaics
code.org: code.org/en-US/promote-computer-science

Appendix

Key Terms:

Algorithm is a set of math rules that can help calculate an answer to a problem.
Artificial Intelligence (AI) is the science and engineering of making intelligent machines, through computer systems. AI works by absorbing information and training data for correlations and patterns to make predictions about future states.
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.
Computer Science (**CS**) is the study of computers and computing, including their theoretical and algorithmic foundations, hardware and software, and their uses for processing information. The discipline of CS includes the study of algorithms and data structures, computer and network design, modeling data and information processes, and artificial intelligence.
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 and is applied in all types of learning, such as math and science. Skills that are often listed as part of computational thinking include: recognizing and defining computational problems (decomposition), developing and using abstractions, creating computational artifacts (constructing algorithms), testing and refining computational artifacts (debugging and evaluation) (K–12 CS Framework, 2016)
Cybersecurity is the art of protecting networks, devices, and data from unauthorized access or criminal use and the practice of ensuring confidentiality, integrity, and availability of information. It seems that everything relies on computers and the internet now—communication (e.g., email, smartphones, tablets), entertainment (e.g., interactive video games, social media, apps), transportation (e.g., navigation systems), shopping (e.g., online shopping, credit cards), medicine (e.g., medical equipment, medical records), and the list goes on.
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). This is helpful for CS but is not CS.
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). This is not CS.
Generative AI (**GenAI**) is a form of AI that can generate new text, audio, images, or videos.
Hardware is the physical components of a computer, which include the motherboard, processor, memory, storage, and drives. The hardware can also include the keyboard, mouse, monitors, printers, and speakers of a computer system.
Misinformation is false or inaccurate information—getting the facts wrong. Disinformation is false information which is deliberately intended to mislead—intentionally misstating the facts.
Programming (or "**coding**") is the collaborative process between computer systems and individuals.
Software is a collection of instructions, data, or computer programs used to run machines, such as running programs, scripts, and applications.

Acknowledgements

We want to thank the following people for their time and contributions to this project:

UCLA Team

Julie Flapan
Jean Ryoo
Michelle Choi
Paula Nazario
Sharisa Chan

Parent Partners

Daisy Amezcua
Kelly Bedford
Irene Blancas
Shelby Hood
Megan Kelly
Judith Moscoso
Paul Robak
Armanda Ruiz
Enrique Vega

Advisory Board

Nakeia Alsup
Yolande Beckles
Shari Dickstein-Staub
Rudy Escobar
Philietz Liu
Ricarose Roque

Suggested citation: Flapan, J., Ryoo, J., Choi, M., Nazario, P., Chan, S. (2025). Plugging into Power: A Guide for and by Parents to Advance Computer Science Education. Los Angeles, California: University of California, Los Angeles.

Sample Letter

Dear ___________________,

I am writing to discuss the importance of expanding access to computer science (CS) education in ___________________. I would like to see more CS classes offered and taught by well-prepared teachers in classrooms full of students representative of our student body, including girls, Latine students, Black students, and low-income students.

Computer science is a critical subject; the skills learned from computational thinking, creativity, and problem-solving, will prepare our students to engage fully in society and thrive in the evolving job market. We must take bold steps to ensure that all of our students are sufficiently prepared for college, careers, and community participation.

Shockingly, the majority of California's students still do not have access to CS education. Students of color, girls, and students from low-income families are less likely to have access to and participation in CS classes. Here in our city, only ______ of students are enrolled in computer science, and only ______ have access.

Our students deserve better. We need to take bold action to ensure they have access to a rigorous computer science education that prepares them for their future—regardless of race, gender, geography, or economic background.

- Not sure where to begin? The CS Equity Guide has everything you need to know to get started, from integrating CS into existing classes to adding CS as a new subject in your school or district.
- Need to know how to prepare teachers to teach CS in K12? Mark the Summer of CS on your professional development calendar. Each year in June, teachers of all grades come together to learn about equity-minded curriculums currently used across California through a series of workshops.
- Wondering how CS fits into the curriculum? You can use the CS Strategic Implementation Plan as a roadmap.
- Not sure how to bring parents and students on board? Host a Family Code Night to give parents/families/caretakers and potential CS students hands-on experience.

For more information about bringing CS to your school, and to see how our community compares to others in California, please visit CSforCA.org

Sincerely,

____

Modelo de Carta

Estimado/a ___________________,

Como padre/madre, me dirijo a usted para discutir la importancia de ampliar el acceso a la educación en ciencias de la computación (CS) en ___________________. Me gustaría ver más clases de CS ofrecidas e impartidas por docentes bien preparados en aulas que representen a nuestra comunidad estudiantil, incluyendo a niñas, estudiantes Latine, estudiantes afrodescendientes y estudiantes de familias de bajos recursos.

La ciencia de la computación es una asignatura fundamental; desde el pensamiento computacional, la creatividad y la resolución de problemas, las habilidades que se aprenden prepararán a nuestros estudiantes para participar plenamente en la sociedad y prosperar en un mercado laboral en constante evolución. Debemos tomar medidas audaces para garantizar que todos nuestros estudiantes estén adecuadamente preparados para la universidad, las carreras profesionales y la participación comunitaria.

Lamentablemente, la mayoría de los estudiantes de California aún no tienen acceso a la educación en ciencias de la computación. Los estudiantes de color, las niñas y los estudiantes provenientes de familias de bajos ingresos tienen menos probabilidades de tener acceso y participar en clases de CS. Aquí en esta ciudad, solo ____________ estudiantes están matriculados en ciencias de la computación, y solo ___________ tienen acceso.

Nuestros estudiantes merecen algo mejor. Necesitamos tomar medidas audaces para asegurarnos de que tengan acceso a una educación rigurosa en ciencias de la computación que los prepare para su futuro, sin importar su raza, género, ubicación geográfica o contexto económico.

- ¿No sabe por dónde empezar? Guía de Equidad en CS tiene todo lo que necesita saber para comenzar, desde la integración de CS en clases existentes hasta la adición de CS como una nueva asignatura en su escuela o distrito.
- ¿Necesita saber cómo preparar a los docentes para enseñar CS en K-12? Marque el "Verano de CS" en su calendario de desarrollo profesional. Cada año, en junio, docentes de todos los niveles se reúnen para aprender sobre currículos enfocados en la equidad que se utilizan actualmente en California a través de una serie de talleres.
- ¿Se pregunta cómo encaja CS en el currículo? Puede usar el Plan Estratégico de Implementación de CS como una hoja de ruta.
- ¿No sabe cómo involucrar a los padres y estudiantes? Organice una Noche de Código en Familia para dar a los padres/familias/cuidadores y estudiantes potenciales de CS una experiencia práctica.

Para obtener más información sobre cómo llevar CS a su escuela y ver cómo se compara nuestra comunidad con otras en el estado, visite CSforCA.org.

Atentamente,

____