Guidance on the Future of Computer Science Education in an Age of AI

teachai.org/cs

Note to the Reader

Guidance on the Future of Computer Science Education in an Age of AI was first published in July 2024 with three briefs and survey insights.

This updated publication adds additional briefs and mini-briefs.

Partners

Guidance on the Future of Computer Science Education in an Age of AI was developed by TeachAI and the Computer Science Teachers Association (CSTA) in partnership with a community of thought leaders.

Lead Partners

Association for the Advancement of Artificial Intelligence

Digital Promise

Everyday AI

GESELLSCHAFT FÜR INFORMATIK

In partnership with Karen Brennan, Bruce Fuda, Maya Israel, and Matti Tedre

Partners

Adobe

AISES

AI Singapore

AI for Education

anpri

Atlassian Foundation

ACTFL Language Connects

aws

Black in AI

Canada Learning Code En Programmation

Center for Black Educator Development

CSET Center for Security and Emerging Technology

Cisco

CollegeBoard

COSSBA

CYBER.ORG

DataScience4everyone

Education Leaders of Color

EidoS Always Learning

EPIC Expanding Pathways in Computing

EY

GitHub

Google

ACE Institute for Advancing Computing Education

Indigitize Computer Science

Infosys Foundation USA

Latinos for Education

micro:bit

Microsoft

ncwit.org

nea National Education Association

nsba National School Boards Association

nsta National Science Teaching Association

PACE Policy Analysis for California Education

Raspberry Pi Foundation

RobinCode Hayallerini Kodla Türkiye

SCIE Sociedad Científica Informática de España

SETDA Leadership · Technology · Innovation · Learning

TACC Texas Advanced Computing Center

TECHNOVATION

Partners

TeachAI Steering Committee

In collaboration with the World Economic Forum

Past Partners

We thank the following organizations and individuals for their partnership in the 2024 publication of the first three briefs and survey insights: AI4K12, Amazon, Getting Smart, Grok Academy, IndigiGenius, Micron, Shuchi Grover, Sociedade Brasileira de Computação, and Steamlabs Africa.


Break it Down
When rock hits
Rock hits blue

Table of Contents

Executive Summary

Recent innovations in artificial intelligence (AI) are provoking a wide range of questions about the powerful technology's role in the computer science (CS) classroom. Educators seeking to understand the opportunities and risks of AI are asking:

- Why is it still necessary to learn to program?
- What do students need to learn about AI?
- How do we leverage AI to broaden access and participation in CS?

While AI offers possibilities for supporting teachers' work and enhancing student learning, there are also concerns about disparate societal impacts. Teachers are concerned about student privacy, the overreliance on AI tools, and students not developing foundational CS skills. These risks demand a cautious approach by everyone in the education community and a commitment to ensuring that AI benefits all students. **Guidance on the Future of Computer Science Education in an Age of AI** is meant to inspire the community to reflect on what it means to teach with and about AI in CS education.

When should we introduce AI coding tools in foundational CS experiences?

There's no universal sequence that answers this question—it depends on when, where, the type of tool, and how students begin learning computer science. For example, primary school students currently have opportunities to develop foundational skills before generative AI tools are age-appropriate. But in middle or high school, students may already encounter these tools in other subjects, potentially shaping how they engage with CS from the start.

Regardless of when AI is introduced, we believe two things are essential:

- Students need a strong foundation in core CS and computational thinking concepts such as algorithms, decomposition, abstraction, and debugging. These skills help students make sense of code before generating or modifying it with AI.
- We're still learning how generative AI can reinforce, not replace, conceptual understanding in CS. Research and classroom use will continue to shape what responsible implementation timelines and best practices look like.

Ultimately, the goal is to prepare students to work with AI as a tool in problem solving, rather than relying on AI as a shortcut around understanding.

The Guidance

At a time when the entire education community is grappling with how to realize the benefits of AI while mitigating the risks, the briefs that compose **Guidance on the Future of Computer Science Education in an Age of AI** serve as the beginning of a discussion rather than definitive answers. The briefs offer preliminary insights for responsibly and effectively integrating AI into primary and secondary CS education, address common misconceptions, and provide a balanced perspective on critical issues.

The guidance illustrates an education community grappling with what AI means for the future of their discipline and serves as an example for other disciplines.

The guidance can be used to support a broad audience:

- Teachers
- Administrators
- Curriculum Providers
- Professional Development Providers
- Standards Writers
- Edtech Developers
- Researchers

The Process

Developed by TeachAI and the Computer Science Teachers Association (CSTA), **Guidance on the Future of Computer Science Education in an Age of AI** was first published in 2024 with three briefs and survey highlights. Additional briefs were published in 2025. Collectively, the briefs are informed by:

- 72 TeachAI advisory committee and government agency participants,
- 46 expert interviews,
- 8 focus groups featuring over 100 teachers, researchers, and curriculum and professional development providers,
- a lead group of researchers and organizations, and
- a global working group of non-profit organizations, teacher associations, and industry leaders.

The guidance also draws on a literature review of dozens of research articles and the insights of more than 360 educators worldwide who responded to a May 2024 CSTA/TeachAI survey.

Each Brief

The briefs address some of the most pressing questions in CS education today:

1. **Why is it Still Important to Learn to Program?**
   Despite their coding abilities, AI tools are far from perfect, and learning to program lays the foundation for students to develop the conceptual understanding, agency, and dispositions needed to understand, use, and evaluate these tools and their outputs.

2. **How Are Computer Science Educators Teaching With and About AI?**
   Learning CS is a natural opportunity for students

to explore the benefits, limitations, and societal impacts of emerging technologies. Early initiatives suggest that teaching with and about AI in CS education has the potential to promote computational thinking, demystify AI, and equip students to use AI tools responsibly. A mindful approach to implementing AI tools is critical to realizing the potential benefits of AI while mitigating the risks.

3 **How Can Students Become Critical Consumers and Responsible Creators of AI?**
Many teachers have long advocated for teaching about technology's ethical and societal impacts as a core component of CS education. Amid the rise of Generative AI (GenAI), CS teachers can help students develop the practices needed to become critical consumers and responsible creators of AI.

4 **How Might Learning Computer Science Evolve in an AI-Driven World?**
AI is changing how students engage with computer science by offering faster feedback and new ways to explore ideas. With strong programming foundations in place, students can learn to direct intelligent systems rather than just use them.

5 **In What Ways Can AI Enhance Creative Thinking in Computer Science?**
Creative thinking thrives when students drive the ideas, and AI supports their exploration. Teachers can use AI to accelerate iteration, personalize projects, and expand what's possible without replacing student originality.

6 **What AI Experiences Are Foundational for Every CS Student?**
This brief outlines four essential experiences that help students recognize, understand, and shape AI systems. It shows how CS teachers can integrate these experiences into existing lessons to build students' technical insight and critical judgment, with example activities and resources for each experience.

7 **How Can AI Support Universal Design for Learning?**
Universal Design for Learning offers a structure for reaching all learners, and AI can help bring that vision to life in CS classrooms. With thoughtful use, teachers can use AI tools to support access, reduce barriers, and give every student meaningful ways to engage with computer science.

8 **How Might AI Be Used to Broaden Participation in Computer Science?**
AI can help make CS more inclusive when used intentionally. In this brief, five thought leaders share powerful ways AI can support student access, engagement, and creativity across contexts: through music and language tools, culturally-grounded projects, personalized supports, and co-designed curriculum models. Their insights highlight both the opportunities and the care required to ensure AI opens doors in CS without reinforcing existing barriers.

The mini-briefs provide background on AI and highlight age-appropriate experiences for young students.

9 **What is AI?**
This quick reference introduces key AI concepts, provides historical and classroom context, and explains the role of CS in helping students build foundational knowledge and skills for understanding AI.

10 **What AI Experiences Are Essential for Primary School Students?**
This resource outlines age-appropriate activities and key goals for introducing AI in the early grades, supporting responsible exploration and digital curiosity.

11 **The Role of AI in Computer Science Education: Results from a Teacher Survey**
CSTA and TeachAI surveyed 364 computer science teachers between March and July 2024. While survey insights are included in the briefs, survey highlights and an infographic are also available.

These resources are designed to evolve alongside the field and help ensure that AI literacy is built on a strong foundation of CS knowledge, responsible practice, and inclusive learning.

⟶ Resource Selection Criteria

The instructional resources featured in these briefs were selected to highlight valuable and accessible AI learning experiences for all students.

Consider these criteria as you develop and evaluate AI resources:

- **Objective Alignment:** Ensure that lesson objectives align with the specific AI learning goals outlined in the curriculum or guidelines, such as standards.
- **Flexible Time Commitment:** Select activities that fit instructional needs and classroom schedules, from 15-minute introductions to comprehensive, multi-week lessons.
- **Beginner-Friendliness:** Focus on activities that are accessible to people new to AI or computer science, ensuring the content is engaging and easy to understand for beginners.
- **Balanced Resources:** Aim for a mix of primary and secondary activities, including unplugged lessons.
- **Credible Sources**: Use resources developed by reputable and recognized curriculum developers in AI and computer science education to ensure quality and reliability.

For more guidance, see the Teacher Accessibility, Equity, and Content (TEC) Rubric for evaluating computing curricula.

Additional AI in Education Resources

For school systems ready to develop guidance on the responsible use of AI, the **AI Guidance for Schools Toolkit** offers seven principles for realizing the benefits of AI while mitigating its risks.

For those just starting to learn about AI, see **Foundational Policy Ideas for AI in Education** for resources designed for education leaders, including **"What Is AI?", "AI in Education and the Workforce,"** and **"Classroom Perspectives on AI**."

The **draft AILit Framework** clarifies AI literacy outcomes for learners, empowering them for the age of AI. Organized into four domains—engaging with, creating with, managing, and designing AI—the framework identifies competences and offers sample primary and secondary classroom use cases.

Finally, see CSTA and AI4K12's Identifying Priorities for All K-12 Students.

Acknowledgments

We thank those who contributed their time and expertise to inform this guidance and look forward to continuing to learn together what it means to teach with and about AI in CS education.

This guidance was developed by TeachAI and CSTA in partnership with Karen Brennan (Harvard University), Quinn Burke (Digital Promise), Marie desJardins (AAAI), Bruce Fuda, Maya Israel (University of Florida), Irene Lee (Everyday AI), Matti Tedre (University of Eastern Finland), and Jan Vahrenhold (German Informatics Society). We thank Christina Gardner-McCune (AI4K12 and University of Florida), Shuchi Grover (Looking Glass Ventures and Edfinity), and Grok Academy for their engagement in the 2024 publication.

We thank the TeachAI Future of CS Workgroup, which includes Adobe, Amazon, Black in AI, College Board, Data Science 4 Everyone, Digital Promise, Education Above All, ETS, Texas Advanced Computing Center, GitHub, Google, Grok Academy, Infosys Foundation USA, Institute for Advancing Computing Education, ISTE, Microsoft, National Council of Teachers of Mathematics, NCWIT, RobinCode, and participating U.S. and global government agencies.

We'd also like to recognize Allen Antoine, Owen Astrachan, Elaine Atherton, Brianna Blaser, Carla Brodley, Center for Computational Thinking and Design at Aarhus University, Marie Casao, Edie Cheng, Tara Chklovski, Jamila Cocchiola, CS Access, Sofía De Jesús, Leigh Ann DeLyser, Charlotte Dungan, Chamisa Edmo, Rudy Escobar, Ben Forta, Crystal Furman, Dan Garcia, Kinnis Gosha, Mark Guzdial, William G. Harris, Sallie Holloway, Lori Jacques, Antti Kiviniemi, John Kleeman, Jill Kowalchuk, Michael Littman, Yolanda Lozano, Don Miller, John Mitchell, Rusty Nye, Jamie Payton, Chris Piech, Jennifer Rosato, Michael Sakowicz, Mehran Sahami, Dan Schneider, Vicky Sedgwick, Ben Shapiro, Sepehr Vakil, Sara Vogel, Hannah Walden, Shira Wein, Benji Xie, and Aman Yadav for their input.

TeachAI is led by a steering committee of Code.org, ETS, ISTE, Khan Academy, and the World Economic Forum.

Authors:
Pat Yongpradit, Code.org
Nikki Meijer, Code.org
Caron Carlson, Code.org
Veronica Ellis, Code.org
Bryan Twarek, CSTA
Jacob Koressel, CSTA

Designer: Abiyasa Adiguna

Suggested Citation
TeachAI & CSTA (2025). Guidance on the Future of Computer Science Education in an Age of AI. Retrieved from teachai.org/cs. [date].

Why is it Still Important to Learn to Program?

Despite the ability of Generative AI (Gen AI) tools to write code, learning to program remains crucial for developing the conceptual understanding, agency, and dispositions necessary to use AI tools effectively, evaluate and modify AI-generated code, and understand the broader context and implications of programming. Programming produces more than code; it encourages creativity, critical thinking, and collaboration and lays a foundation for learning the discipline of computer science (CS). AI might augment learning, but it cannot replace the understanding and skills gained through learning to program.

Recent advances in Gen AI's ability to write code have led some industry leaders to claim that learning to program is obsolete. Even as professionals use AI tools to generate a growing proportion of code, learning to program continues to be a context for developing the problem-solving and computational thinking skills to use these tools appropriately, evaluate and modify their output, and place the results in context (Salehi et al., 2020). Rather than diminishing the importance of learning to program, Gen AI tools highlight the need for a foundational understanding of programming to comprehend and appreciate these tools.

In the same way that students must learn "number sense" – a conceptual understanding of numbers and how they are related and connected – before they can move on to algebra or calculus, they need to develop "code sense" to be successful in this new era of AI-assisted programming (Johnson, 2024).

⟶ Code Sense

We define **code sense** as the conceptual understanding of a computer program's underlying design, processes, and system relationships, as well as the mental capacity to analyze, simulate, and predict a program's behavior.

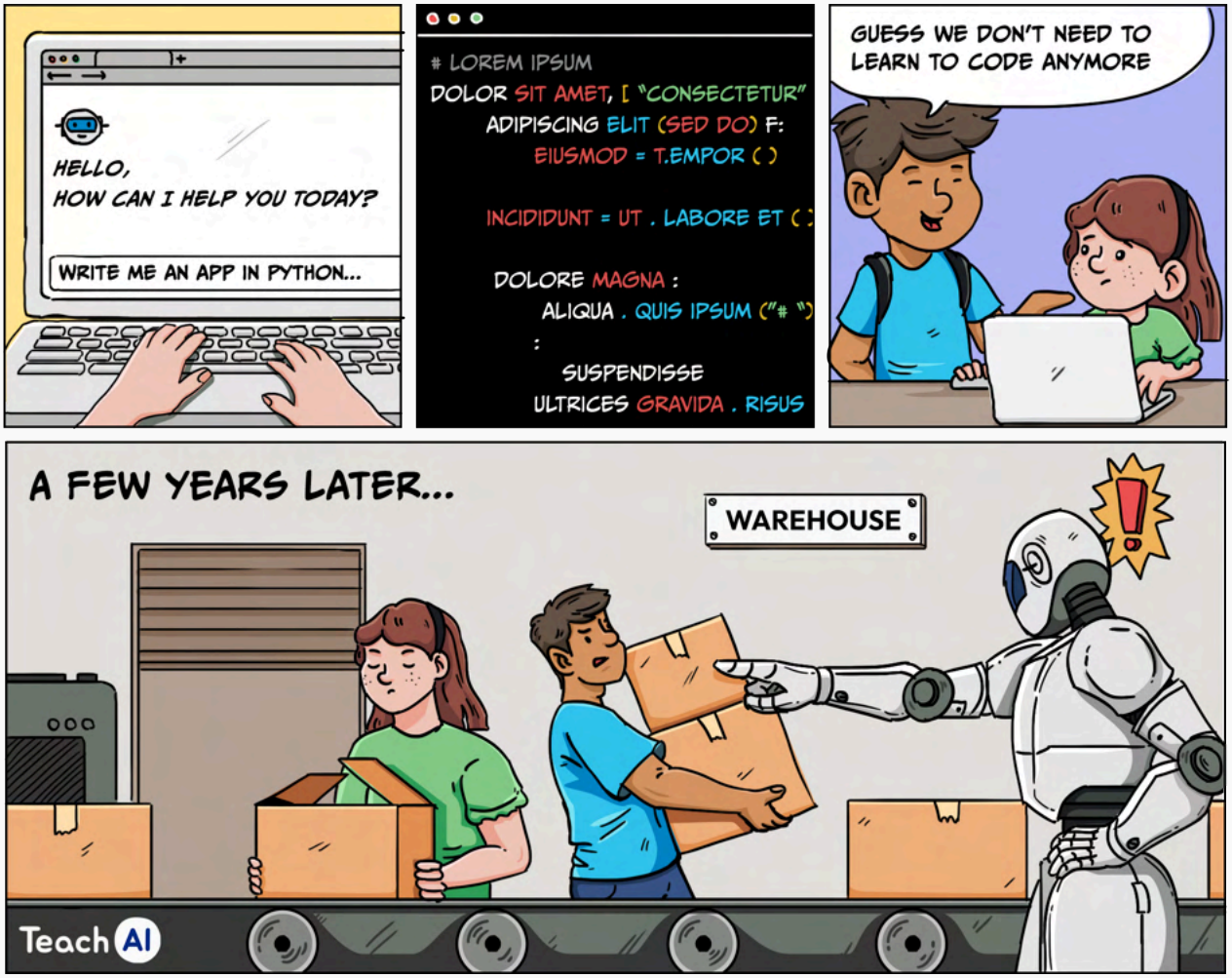
Code sense supports the ability to debug effectively, optimize performance, and understand the broader context of the code within a system.

> "In my vision, the child programs the computer and, in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building."
>
> - **Seymour Papert**, Mindstorms: Children, Computers, And Powerful Ideas, 1980, p.5

For our purposes, "programming" is used interchangeably with coding and includes aspects of software development. Programming is defined as the creative process of designing, writing, testing, and maintaining code to perform specific functions and solve problems.

In this brief, we dispel the myth that Gen AI coding tools make it unnecessary for students to learn to program.

Learn to Program or Follow Commands: A Comic



AI is not magic. It needs human expertise and guidance.

Myth 1: AI coding tools make programming knowledge and skills unnecessary.

To maximize the value and effectiveness of AI tools, developers need to know how to structure effective prompts, refinements, and requests (Kirova et al., 2024). Learning to program, which is more than learning the syntax of a programming language, gives students opportunities to understand programming logic, practice solving problems, and develop an understanding of algorithms and data structures. This knowledge is necessary to direct AI tools effectively.

For example, early research studying novices aged 10 to 17 years using code generators has shown that those with greater programming fluency benefit more from the tool. This suggests that AI tools are most effective in augmenting, rather than replacing, programming knowledge and skills (Kazemitabaar et al., 2023).

⟶ How Will Programming Evolve in an Age of AI?

From assembly language to modern block-based languages, programming has continuously evolved to be more accessible by abstracting lower-level programming details. Gen AI tools may continue this trend and allow programmers to focus more on creativity, problem solving, and addressing ethical concerns rather than the mechanics of translating instructions into a programming language. Some programming tasks will be augmented by AI, and some will be replaced.

How should learning to program evolve to reflect the changing nature of programming itself? What insights can CS education gain from how software engineers use Gen AI tools? Sign up for updates to learn more about the future of CS education.

"We need to help students learn to ask questions well. We need to teach them to have ownership of these tools."

- **Christina Gardner-McCune**, Associate Professor, University of Florida and Co-Lead of AI4K12

AI is not perfect. It needs human oversight.

Myth 2: Students do not need to learn to program because AI can do it accurately and independently.

Although Gen AI tools are improving, they are known to suggest code that is inaccurate, contains security vulnerabilities and biases, or is misaligned with a user's intent. A 2023 study of three popular code generators found that they produced correct code between 31% and 65% of the time (Yetiştiren et al., 2023). Another study showed that human and AI pairs critique code more effectively than either can on their own. Learning to program lays the groundwork for developing the code sense and computational thinking skills needed to evaluate, debug, modify, and situate AI-generated code in the context of a larger program.

"AI makes mistakes. Students need to understand the basics first, so they can critically understand what they need from AI and also how to evaluate the results."

- **Neda Blackburn**, STEM Director, Roland Park Country School

⟶ Survey Insight

CS teachers responding to a May 2024 CSTA/TeachAI survey commonly noted that even with AI's prevalence and coding abilities, learning to program is still essential for students to foster the code sense and computational thinking skills necessary to use and build future AI applications.

AI will not replace the need for programmers. It needs human creativity and domain expertise.

Myth 3: Students do not need to learn to program because AI will replace all programming jobs.

The rapid uptake of Gen AI in the workplace is changing programming practices without obviating the need for humans. Learning to program is shifting from focusing on code generation to more code reading, evaluation, debugging, and refactoring (Kirova et al., 2024).

⟶ Focus on the Fundamentals

Gen AI coding tools are rapidly improving to help software engineers by eliminating repetitive tasks, suggesting code, and even translating code from one programming language to another (Stryker, 2024). While industry professionals will continue to adapt to these innovations, students learning to program need to build the capacity to understand the fundamental, durable concepts behind these evolving tools to use them appropriately.

Human creativity and domain expertise have been and will be crucial in programming. Creativity drives innovative problem solving, intuitive user interface design, and feature development, while domain expertise ensures the solutions are accurate, relevant, and address specific use cases. Together, they enable the creation of robust, user-friendly programs that meet both technical and contextual needs.

> "There is a longstanding need for people who have software development skills. What you need to know may shift – things that were obviously critically important or seen as peripheral will need to change in their relative importance."
>
> - **Ben Shapiro**, Associate Professor, University of Washington

AI can code, but there is more to programming than code.

Myth 4: The only purpose of learning to program is to produce programs; now, AI can do that for us.

Programming goes beyond translating instructions into a programming language; it involves exploring needs, writing specifications, and testing, debugging, and analyzing code. Learning to program offers students a platform for collaborative problem-solving, creative expression, and discovering joy in creating something new (Kafai et al., 2015).

When students work together, they may find it easier not only to absorb new concepts and vocabulary but also to express their curiosity and enthusiasm. Research shows pair programming can boost student retention in CS, particularly among some underrepresented groups

(Campe et al., 2019). Having these opportunities while learning to program nurtures resilient, life-long learners equipped to thrive in a rapidly changing world. See How Might Learning Computer Science Evolve in an AI-Driven World?

Learning to program also has emotional value. A small-scale study of primary school students showed that children felt significantly happier, more excited, and more in control after learning to program (Tisza et al., 2023). Results like these depend on many factors, especially a high-quality teacher, and they highlight the benefits learning to program can have on students' emotional state, reminding us to consider how AI use may affect students' feelings about CS.

> "Learning how to code isn't just about preparing young people for potential participation in the technology workforce. Learning how to code can be an incredible opportunity for creative expression, enabling learners to share their ideas and voice with the world, bringing joy, satisfaction, and a sense of purpose."
>
> - **Karen Brennan**, Professor, Harvard University

⟶ Computer Science is More than Programming

Learning to program is just one aspect of learning CS. For example, the CSTA K-12 Standards also include understanding how computers work, how messages are sent through the Internet, how data is collected and analyzed, and how technology impacts society. Even if programming were obsolete, a foundational CS experience would still be essential to:

- fueling innovation across multiple fields, such as data science or computational biology,
- addressing cultural and structural barriers to pursuing post-secondary CS,
- expanding access to high-paying, highly satisfying jobs, and
- promoting greater diversity in a myriad of fields (Lewis, 2017).

See How Can AI Enhance Creative Thinking in Computer Science?

Three Ways to Take Action

Why is it Still Important to Learn to Program?

Reflection	1. Given that AI can generate code, how can you balance the use of AI tools with the need for students to develop fundamental coding skills and code sense? 2. In what ways can you explicitly emphasize computational thinking skills, whether teaching students programming, writing curriculum, or training teachers? What practical activities or resources can help?
Call to Action	**Administrators** 1. Continue offering introductory CS courses that include programming. 2. Support CS teachers' professional growth through communities such as a CSTA chapter. 3. Support schoolwide professional development and help teachers share lessons learned. 4. Develop guidelines based on resources like the TeachAI AI Guidance for Schools Toolkit.
	Teachers 1. Be explicit about the fundamental concepts and skills you want students to learn (see CSTA K-12 Standards), such as debugging, regardless of whether they use AI tools. 2. Explore using AI tools in the learning process before introducing them to students. 3. Discuss if, when, and how AI tools should be used in CS education. Develop principles to guide AI use in programming assignments.
Further Study	1. How does learning how to program with Gen AI tools compare to traditional instructional methods? How might they help or hurt? 2. How might coding assistant chatbots impact students with less prior CS experience in CS? 3. What are the most common errors and biases found in AI-generated code in educational settings? How can educators train students to identify, evaluate, and correct these issues?

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

How Are Computer Science Educators Teaching With and About AI?

Computer science (CS) is a natural context for students to explore AI's benefits, limitations, and societal impacts, and CS teachers overwhelmingly want professional development to facilitate these learning opportunities effectively. Many teachers already teach with and about AI in a CS context to emphasize computational thinking, demystify AI, and equip students to use AI tools responsibly.

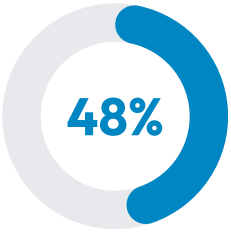
As AI becomes pervasive in students' lives in everything from recommendation systems to voice assistants, CS teachers are incorporating lessons to demystify AI and encourage a critical mindset toward its benefits, limitations, and societal impacts. CS teachers are using AI tools to help explain code, provide timely feedback to students, and help them break through "coder's block" and syntax challenges so that they can focus on the creative aspects of programming. All teachers should follow existing policies, such as age restrictions based on privacy or security concerns, and consider the appropriateness of AI tools for specific activities even when the tools are permitted. There is a growing body of unplugged resources to help students learn about AI when policies or limited technological infrastructure prevent AI use.

In this brief, we examine how teachers teach with AI, such as using AI as a tool for student learning or teacher support, and teach about AI, such as how it works and its ethical and societal implications.

Survey Highlights

Guidance on Computer Science Education in an Age of AI is informed by a survey of CS teachers (n = 364 teachers, 24% primary, 76% secondary, 12% international) administered by the CSTA and TeachAI in May 2024.

of teachers think students in introductory courses should learn about AI.



of teachers said they feel equipped to teach about AI.

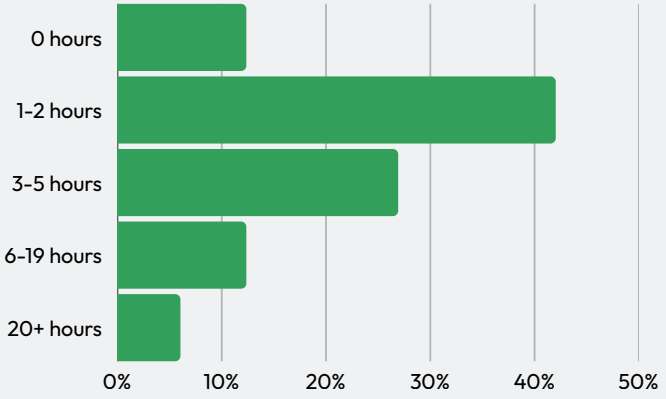
of teachers said they would benefit from professional development to learn how to use and teach about AI.

Teaching About AI

CS teachers are incorporating AI into discussions, demonstrations, and unplugged activities to demystify AI and make it more approachable.

CS teachers are teaching students how AI works and discussing AI's ethical implications. Most teachers (79%) agree that curriculum and standards should be updated to emphasize AI.

⟶ How much do students currently learn about AI in your computer science curriculum?



Teachers indicated that foundational CS concepts like algorithms help students develop the problem-solving skills and awareness of societal impacts required to use AI tools critically and to create with them responsibly. Classroom activities about AI can examine topics such as:

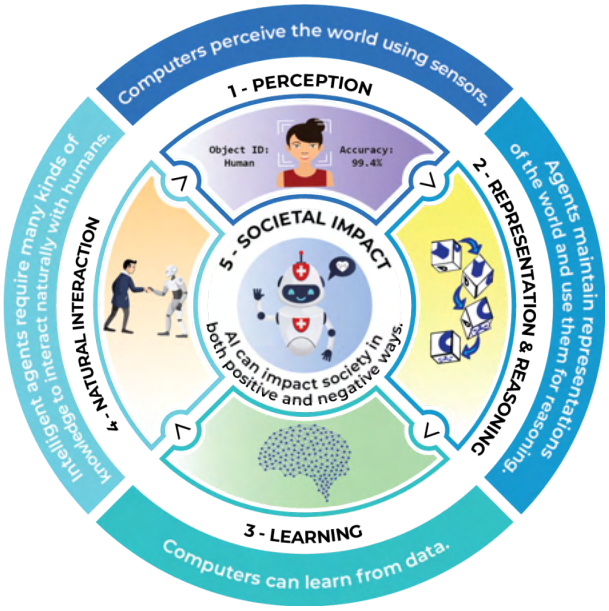
- underlying programming and algorithmic concepts,
- common uses,
- limitations, and
- ethical use and impact on society (Lee & Kwon, 2024).

What's more, topics such as algorithms, classifiers, natural language processing, and AI's ethical issues extend beyond the CS classroom and are relevant in mathematics, language, biology, and social science (Casal-Otero et al., 2023).

> "I do hope that educators realize the importance of facing the reality of AI as well as understanding that the goal of a paradigm shift like this should not be about policing students but about enabling students to think about using [and] understanding this new technology."
>
> - **Ben Smith**, CS Teacher, Rowland Hall School

⟶ AI4K12

Since 2018, the AI4K12 Initiative has been developing national guidelines for teaching about AI, organized around the 5 Big Ideas in AI.



AI in Primary CS Education

Vicky Sedgwick, a primary school educator, integrates machine learning into lessons that address the CSTA K-12 Standards on data and analysis. Students sort, classify, and label data and use the data to train a simple model. Students can then predict how the model

might classify new pieces of data and explore how machines learn and begin to recognize bias in data. AI concepts like natural language processing and image recognition also fit into algorithms and programming standards. Primary and early secondary school students can explore how AI works with resources such as Code.org's AI for Oceans, Machine Learning For Kids, or MIT RAISE Playground. See What AI Experiences Are Foundational for Every Student?

There is also a growing body of unplugged activities that do not require access to AI or a computer and are available for teaching AI concepts. For example, AI technology playing cards teach about input, algorithm, and output to demonstrate how developers' choices affect results (Long et al., 2021).

Revised CSTA K-12 Standards, slated for release in summer 2026, will include AI-related learning goals. In the meantime, the Reimagining CS Pathways project has defined foundational CS content that includes learning progressions specific to AI.

> "[T]here is a tendency to make AI seem either magical, sentient, infallible, or overly human . . . Since such (mis)representations are rife in mainstream discourse, K-12 education needs to work extra hard to address this challenge through approaches to demystify AI and lift the hood on how it works ." (Grover, 2024)
>
> - **Shuchi Grover**, Director, Looking Glass Ventures and Edfinity

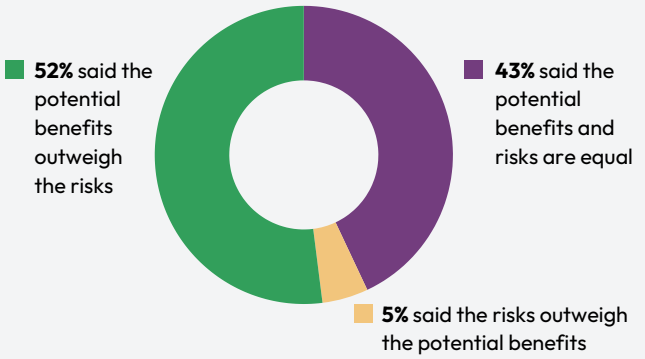
Teaching with AI

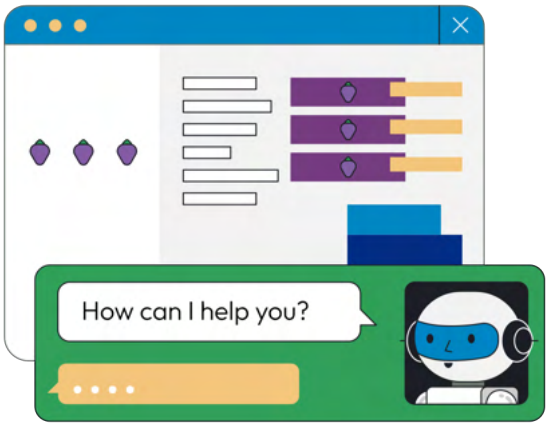
CS teachers are integrating AI tools into instruction, assignments, and assessment.

A systematic review of K-12 AI education from 2018 to 2023 describes a variety of early initiatives to teach with AI by using machine learning tools and AI platforms to help students understand basic concepts, such as training data, testing models, and identifying bias (Lee & Kwon, 2024). Hands-on experiences with AI tools not only helped students develop an understanding of AI concepts but also made learning more fun. Although the specific way that AI is used in the classroom is an essential factor, the review also suggested that using AI in the classroom can help students:

- develop foundational CS skills,
- reflect on critical and ethical considerations about the role of AI and its use,
- apply AI knowledge to real-life problems, and
- improve motivation and interest in technology.

⟶ Thinking about ways AI tools could be used in CS education...





AI Feedback

Maria Camarena, a secondary CS teacher in California, uses ChatGPT to improve students' programming skills. Students input lines of code they have written and a description of the code and receive programming and writing feedback. This approach strengthens programming skills and supports multilingual students with their English language development.

Early research also identifies opportunities to use Large Language Models (LLMs) to improve the CS learning experience and foster critical-thinking skills, including:

- generating code as a starting point for a solution,
- teaching students how to describe solutions in natural language, and
- guiding problem-solving strategies (Prather et al., 2023).

LLMs have also been shown to help provide a vast selection of solutions of varying quality so students can practice critical analysis by examining multiple answers to a problem (Denny et al., 2023).

In one early effort to revise a post-secondary introductory CS curriculum to use LLMs, problem decomposition and explaining/testing/debugging code were emphasized over syntax and writing code (Vadaparty et al., 2024). The scope of students' projects transcended what was typically seen in a CS1 course, and their exam results reflected results similar to previous CS1 classes.

Pairing with AI

Michael Phelan, a CS teacher in North Carolina, allows students in his Python 2 class to use ChatGPT or Copilot to give them an experience similar to pair programming while pushing them to improve their own thinking. However, it is vital to Phelan that students have a comprehensive understanding of what they are doing rather than relying on AI to do the work.

Learning from Math Education

Lori Jacques, a former math teacher and current CS professor, describes three approaches based on math pedagogy to help students look at problems differently and improve their programming skills and conceptual understanding (Jacques, 2023):

- **Create Multiple Representations**: Students can use Generative AI (Gen AI) to generate code, explain code in plain language, illustrate how the output changes as the code executes using a flow chart, and compare the different representations and their pros and cons.
- **Explore Different Approaches**: Students can use AI code generators to create different programs to address the same problem and then compare them to each other and human-generated solutions, including their own.
- **Explain Code Created by AI**: Students can analyze, describe, and explain the functionality of AI-generated "worked examples," which may or may not be correct.

Concerns About AI's Risks

CS teachers responding to the CSTA/TeachAI survey frequently cited these three concerns about the potential risks of using AI in the classroom:

- AI will do the thinking for students.
- Students will become overreliant on AI tools.
- Students will lose essential skills.

These concerns can be mitigated by teaching foundational programming skills and critical engagement with AI tools. This approach ensures that students use AI as a supportive tool, maintaining their problem-solving abilities and understanding the underlying logic.

"Students have a tendency to 'trust' the AI and blindly copy and paste the code but lack the skills to 'talk to' the code with AI and truly understand what is going on. This causes them frustration when later attempts don't work, and they lack the vocabulary or experience needed to properly explain what problems or tasks need to be done to their AI assistant."

- **Graham Nolan**, CS Teacher, Hong Kong International School

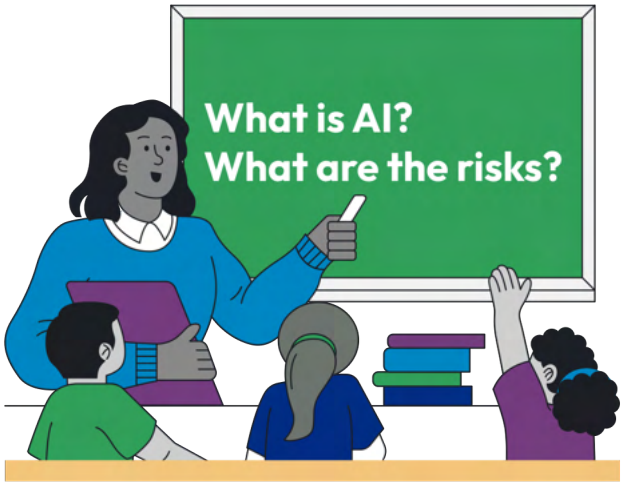
For more on the ongoing need for students to learn CS fundamentals, see Why is it Still Important to Learn to Program?

Teacher Support

CS teachers are using AI to support and streamline their own work.

Teachers who participated in the CSTA/TeachAI survey reported using AI to facilitate their work in various ways, including tailoring and refining lesson plans; providing differentiation for students who need either more challenge or more support; and organizing ideas, breaking them down, and re-explaining them.

These uses mirror common opportunities for Gen AI to streamline teacher workloads. In one early study, researchers prompted the code generator Codex with source code to generate new programming exercises. They found that a large majority were completely novel and aligned with the topic and theme at hand (Denny et al., 2023). In addition to saving instructors' time, the variety of solutions that code generators can create from one input can offer students a view into different approaches to tackling the same problem. See How Can AI Support Universal Design for Learning?



"While AI can automate certain tasks, such as grading or feedback provision, overreliance on AI tools may diminish educators' role in the teaching process. It's essential to strike a balance between leveraging AI for efficiency and maintaining the human element of teaching, including personalized interaction and mentorship."

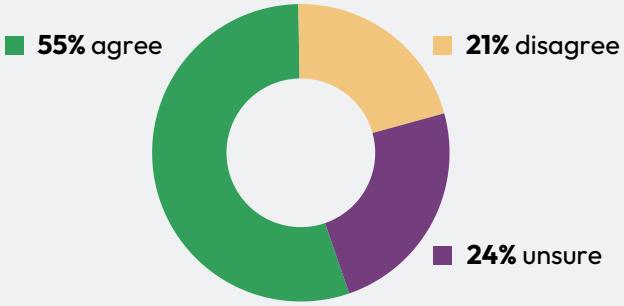
- **Fred Sagwe**, CS Teacher, Shimo La Tewa School and Robotics Society of Kenya

Inaccuracy and Overreliance

Despite the potential benefits of using AI tools to enhance learning and provide instructor support, common concerns involve Gen AI's tendency to produce inaccurate information and students' ability to make sense of the output. Both students and teachers also express concerns that students may not learn their material when they turn to Gen AI for help (Zastudil et al., 2023).

Teachers have mixed opinions about whether students in introductory CS classes should learn to program with AI. The benefits and risks of learning to program with AI tools are still being discovered and will depend on the classroom context and how the tools are used.

⟶ Students in introductory CS classes should learn to program with AI.



⟶ Survey Insight

These survey trends were validated through a larger and more representative study conducted by CSTA and the Kapor Foundation. See a summary in the education chapter of the 2025 AI Index Report.

⟶ Global Perspectives

As of July 2025, 44 state education agencies in the U.S. and over 30 government agencies globally participate in TeachAI. Here are some global perspectives on the future of CS education:

"When we teach students how to understand and responsibly leverage AI, we equip the next generation of technology users and creators with skills and confidence to innovate."

- **Oksana Pasichnyk**, CS Teacher, Lyceum Sykhivsky, Ukraine

"AI in coding classrooms faces challenges in implementation due to varying teacher and student understanding and skill levels. Effective AI integration requires teacher and student training, well-designed learning experiences, ethical guidelines, and ongoing support. Despite these challenges, AI offers immense potential to transform coding education in Thailand."

- **Cheeraporn Sangkawetai**, Institute for the Promotion of Teaching Science and Technology, Thailand

Three Ways to Take Action

How Are Computer Science Educators Teaching With and About AI?

Reflection	1. In what situations might teaching with and about AI be useful? In what situations might it be inappropriate? 2. Where can teaching about AI be integrated into the existing curriculum? 3. What changes can be anticipated in students' engagement, motivation, or understanding of programming concepts after introducing AI tools?
Call to Action	**Teachers** 1. Participate in ongoing professional development (PD) to deepen knowledge and skills in teaching with and about AI. Find PD from CSTA, CSTA chapters, and curated programs. 2. Follow local guidance on the use of AI tools in classrooms. When local policies do not exist, develop guidelines for using AI tools on classroom assignments. Prioritize responsible practices that encourage students to determine if, how, when, and why they would use AI. 3. When using AI in your work, constantly evaluate the output to see how it fits your classroom context, objectives, and teaching style.
	Curriculum and Professional Development Providers 1. Include AI as a topic and tool and explore uses of AI that will enhance activities, examples, lessons, and projects. Proceed cautiously, with an eye towards safe and responsible use. 2. Provide professional development enabling teachers to experiment, share use cases and insights, and field questions and concerns in their schools. 3. Ground questions and projects for students in real-world examples that also illustrate the benefits and risks of AI.
Further Study	1. How should CS curriculum and standards be updated to include AI concepts and tools? 2. How can school systems address barriers to access, support effective pedagogical design, and provide powerful learning opportunities to provide equitable access to AI education? See **Classroom Perspectives on AI** (TeachAI, 2024) for more information on these issues. 3. How do the interactions between teachers, students, and AI tools influence the teaching and learning process and outcomes?

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

How Can Students Become Critical Consumers and Responsible Creators of AI?

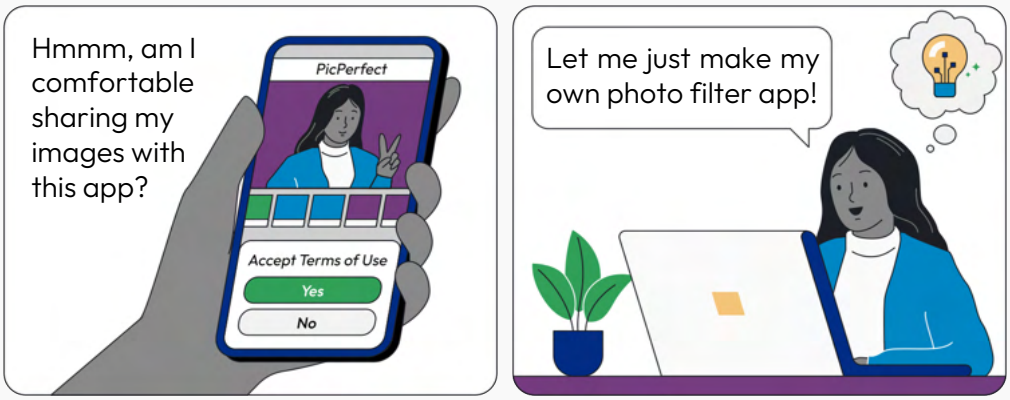
Teaching students to become critical consumers and responsible creators of AI involves integrating ethical and societal considerations into computer science (CS) education. CS teachers are incorporating discussions about transparency, accountability, information accuracy, privacy, fairness, and ethical design into their curricula. These lessons empower students to evaluate AI outputs critically, understand the implications of AI on society, and design technologies that prioritize equity and responsibility. More than any other subject, CS provides an opportunity to explore AI's societal and ethical impacts by learning how AI models are trained, understanding how they work, and developing their own models.

From producing deepfakes to amplifying gender, racial, and cultural biases, AI's potential to harm individuals and communities is in the spotlight, creating increasing demand for incorporating societal and ethical considerations into CS education.

An essential component of teaching students about AI is empowering them to understand how AI models are built and to think critically about the benefits, harms, possible safeguards, and the people and communities affected by this powerful technology. As more CS education involves consuming and creating AI, students must also learn to ask whether and why they should use AI in the first place.

> "When it comes to AI education, we do not have the luxury of burying our heads in the sand. CS teachers have the opportunity and responsibility to lead students in understanding the societal and ethical implications of AI: the good and the bad, the benefits and harms, the possibilities and realities."
>
> - **Charity Freeman**, CSTA Board Chair

Critical Consumers, Responsible Creators

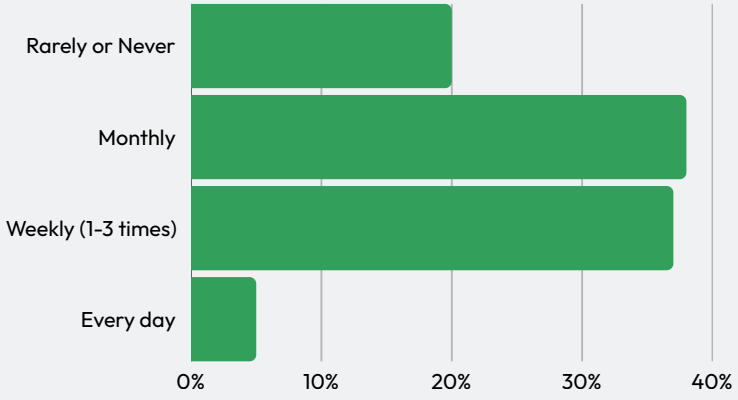


According to Reimagining CS Pathways: High School and Beyond, the societal impacts of computing and related ethical implications are the highest-rated topics of importance across the CS education community.

Additionally, 66% of CS teachers responding to a CSTA/TeachAI Survey (May 2024) said that conversations about AI have helped students understand the societal and ethical issues surrounding AI. Ethical concerns about AI also emerged as one of the most requested professional learning topics.

In this brief, we highlight a few of AI's ethical and societal considerations and offer some practices for developing critical AI consumers and responsible AI creators.

⟶ How often do you supplement your CS curriculum to include discussions of ethical questions related to AI, such as algorithmic bias, digital citizenship, and privacy?



Transparency and Accountability

AI systems are widely perceived as opaque, containing unknowable data sources and inscrutable decision-making processes.

When equipped with explanations about how AI systems work, teachers are less reluctant to use them in their classrooms and less concerned about the technology's impact on personal autonomy. Early research suggests that teachers are more likely to trust AI when given the ability to review and override AI recommendations (Nazaretsky et al., 2022).

Classroom Practice: Create model cards that explain the intended use, training dataset, accuracy, and limitations of a classification or prediction model.

When asked about the potential risks of including AI in the classroom, CS teachers responding to the CSTA/TeachAI Survey emphasized that students must be made aware that AI makes mistakes and can have disparate societal impacts, often perpetuating harm on marginalized communities. These same concerns were commonly cited as reasons for students to deepen their core CS knowledge.

> "Students are more empowered to engage in popular AI discussions when they understand the technical side of the ethical and societal impacts of AI."
>
> - **Sepehr Vakil**, Associate Professor, Northwestern University

Information Accuracy and Integrity

AI makes it easier for unethical parties to produce misinformation and disinformation by mimicking voices, images, and videos.

During the 2023-24 academic year, parents and students in California, New Jersey, and Washington sounded the alarm over AI: Classmates used AI-based image generators to create explicit "deepfakes" from pictures online and circulated the images on social media. Districts responded by updating their social media guidelines and educating students about the appropriate and acceptable use of Generative AI (Gen AI) tools while urging others to learn more about misinformation. There is a broad range of content authenticity issues, from creative works that imitate specific artists to deepfakes in the democratic process, which raise significant ethical and legal concerns about originality, consent, and the potential for misinformation.

Classroom Practice: Discuss the importance of verifying information sources and explore ways to validate content. Explore instances of when AI output adds value to individuals and society and when it does not. Discuss the importance of citing sources when using AI.

"Students can potentially misuse AI for themselves and others, but teaching them and being open about the benefits and harmful effects and potential upsides and downsides far outweigh the potential risks."

- **Art Lopez**, Project Manager, Coding Our Future, University of California, San Diego

"The hard problems of AI and ML education come down to civic values, philosophy, ethics, fairness, bias, transparency, explainability. The social studies teachers - we can learn from them."

- **Ben Shapiro**, Associate Professor, University of Washington

Privacy Rights

Data collected by AI-powered systems and educational tools raise concerns about students' privacy rights.

In March 2024, a coalition of 41 educational and civil rights organizations wrote to the U.S. Department of Education, asking to end the funding of AI-powered security systems in K-12 schools. They were concerned about AI's potential to violate students' civil rights. Especially alarming, the group said, was the increasing use of facial recognition technologies, social media surveillance, behavioral threat assessments, predictive policing, and other surveillance tools.

As students turn over growing volumes of personal data, including data collected by educational tools, they must understand the potential for combining it to divulge sensitive information. Moreover, tracking and monitoring components in AI systems that predict student performance, weaknesses, and strengths can inhibit their participation at school (Akgun & Greenhow, 2021).

Classroom Practice: Discuss how, when, and why students' personal data are collected, analyzed, interpreted, and used. Consider how decision-makers might use this data to their advantage — even if it is something students are not okay with. Discuss students' rights and how they can opt out of data collection.

"Anyone developing technologies has a responsibility to keep ethics top of mind and create things that will not cause harm. There should be a pledge or agreement that these persons go through so that 'doing no harm' is at the basis of their profession. . . This must also be followed with action."

- **DaQuan Bashir**, Professional Learning Manager, CSTA

Fairness and Justice

AI systems' disparate impacts on different demographic groups raise alarms about fairness and justice in our digitally driven society.

AI systems can reflect and amplify the implicit biases of their creators. Understanding AI biases involves understanding how people create technologies like Large Language Models (LLMs) and how biases can creep into all stages of development, such as selecting training data and selecting which features in the training data to prioritize (Broll & Grover, 2023). Examples of discrimination against women, people who are economically disadvantaged, or people of color as a result of algorithmic bias have shown up in hiring, health care, insurance, and mortgage lending decisions, in addition to criminal justice assessments.

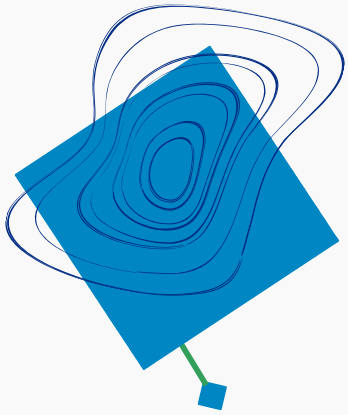
The non-profit journalism group ProPublica in 2016 was the first to uncover racial bias in COMPAS (Correctional Offender Management Profiling for Alternative Sanctions), an AI-powered system used to predict a criminal defendant's recidivism risk. COMPAS falsely identified Black defendants as likely future criminals nearly twice as often as white defendants. The AI model was trained on historical crime data, and, as a result, individuals who were part of a demographic historically targeted by law enforcement were likely to receive higher recidivism scores. A related issue raised by the COMPAS example is transparency, as the designers of COMPAS refused to reveal details of the software and its algorithms.

Classroom Practice: Discuss the pros and cons of using AI for different projects and consider when another solution might be a better option. Discuss how AI tools might be used to detect and measure bias in datasets.

"[A]ll young people need to understand whether the AI with which they knowingly or unknowingly engage has treated them fairly." (Holmes et al., 2022, p.31)

⟶ Responsible AI & Tech Justice

The Kapor Foundation's Responsible AI & Tech Justice: A Guide for K-12 Education helps educators develop learning experiences that prioritize critically evaluating technologies and creating more equitable solutions. One recommendation is to "examine the AI technology creation ecosystem from who designs and develops products and how they are developed, to who invests in their creation and benefits from their adoption."



Ethical Design

Transparency, accountability, equity, and privacy should be principles in technology development.

Ethical design recognizes that all technologies are embedded in social systems and reflect designers' beliefs. Ethical design considers societal values, includes potential users throughout the design process, especially those most affected by the design, and follows principles of transparency, explainability, accountability, equity, and privacy.

In a study of recommendation algorithms and ethical design, students as young as 5th graders were shown to be capable of recognizing design agendas and identifying influences on those agendas (DiPaola et al., 2020). Students were asked to identify relevant parties associated with YouTube and their potential values, encouraging questions about the data collected from the platform, how it influences what they watch, and how it affects their beliefs. Students then had an opportunity to develop their own design goals and agendas using ethical design principles.

Classroom Practice: Early in the design process, have students explore different user personas: representative examples of potential users with different backgrounds, desires, ability levels, and perspectives. For more information, see Practice 1: Fostering an Inclusive Computing Culture in the K-12 CS Framework.

⟶ AI4K12: Societal Impact

AI is a disruptive technology that will not only impact the economy and employment but also shape social and cultural norms (AI4K12 Framework's Big Idea 5).

See How Might AI Be Used to Broaden Participation in Computer Science?

Ethical Concerns and Practices

Topic	Ethical Concern	Classroom Practice
Transparency and Accountability	AI systems are widely perceived as opaque, containing unknowable data sources and inscrutable decision-making processes.	Create model cards that explain the intended use, training dataset, accuracy, and limitations of a classification or prediction model.
Information Accuracy and Integrity	AI makes it easier for unethical parties to produce misinformation and disinformation by mimicking voices, images, and videos.	Discuss the importance of verifying information sources and explore ways to validate content.
Privacy Rights	Data collected by AI-powered systems and educational tools highlight student privacy issues.	Discuss how, when, and why students' personal data is collected, analyzed, interpreted, and used.
Fairness and Justice	AI systems' disparate impacts on different demographic groups raise concerns about fairness and justice.	List the pros and cons of using AI for different projects and consider when other solutions might be a better option.
Ethical Design	Transparency, accountability, fairness, and privacy are crucial in technology development.	Address the needs of diverse users and consider when the risks of using AI might outweigh the benefits for an individual or community.

Three Ways to Take Action

How Can Students Become Critical Consumers and Responsible Creators of AI?

Reflection	1. How can you incorporate ethical and societal issues related to AI into CS content? How can you address the potential challenges when engaging in sensitive topics? What can be learned from other subjects? 2. How can you model the ethical and responsible use of AI tools? What steps can students take to understand AI tools' limitations and biases and evaluate their outputs critically? 3. What data do you and students share on online platforms and with AI systems, both knowingly and unknowingly? What risks are associated with sharing personal data?
Call to Action	**Standards Writers** 1. Ensure that ethical considerations and societal impacts of AI and technology are embedded in primary and secondary education standards. 2. Connect the societal and ethical impacts of AI to how AI works and how people develop AI technologies. 3. Develop standards that integrate CS education with other disciplines, such as science, social studies, philosophy, and ethics, to provide a holistic view of AI's societal impacts.
	Curriculum and Professional Development Providers 1. Include deliberate conversations about bias, ethics, and AI's impact throughout the curriculum. Provide students with reflection questions at the start and end of each project. 2. Encourage perspective-taking by asking about user stories and how tools might work. Frame the potential for positive, negative, and unintended consequences. 3. Create space for honest conversations and thoughtful questions. Include questioning protocols to foster deeper thinking about how what people design will affect others.
Further Study	1. How does incorporating ethical considerations, such as bias and fairness, into CS content affect students' understanding and critical analysis of AI technologies? 2. What are effective methods for teaching transparency and accountability? How do these methods influence students' trust and willingness to engage with AI technologies? 3. How can educational interventions be designed to effectively reduce biases and promote fairness in students' AI projects and understanding?

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

How Might Learning Computer Science Evolve in an AI-Driven World?

When CS teachers reimagine their curriculum with AI topics and tools, students evolve from writing code to critically engaging with AI systems—interpreting, refining, and shaping how AI behaves, while developing the ethical judgment and design skills needed to solve real-world problems.

Introduction

Computer science (CS) education has evolved over the past few decades to become more approachable and engaging. Programming is a great example, as it has shifted from focusing on low-level details, such as memory allocation and hardware management, to emphasizing higher-level concepts like computational thinking (CT) and integrating the societal and ethical impacts of algorithmic decisions. Today, learning to program incorporates innovative tools, including block-based coding environments, that make learning CS more approachable for a broader range of students.

Generative AI (GenAI) tools continue this trend by allowing students to shift their focus from the specific syntax of a programming language to engaging more in code evaluation and human-centered design.

Explore this accompanying brief, "How Might AI Be Used to Broaden Participation in Computer Science?"

But AI is influencing more than just the teaching of programming. It is changing:

- what CS **topics** are prioritized;
- which **practices**—like human-centered design—deserve more emphasis;
- how **instruction** changes when students and teachers have AI support; and
- which classroom **tools** support students in creating, rather than merely consuming, modern technology.

This brief envisions how CS education evolves in an age of AI from a student's perspective.

⟶ Terms to Know

Generative AI (GenAI) is a type of artificial intelligence that learns patterns from massive amounts of existing data and then creates new content, like text, images, graphics, or music. It produces outputs by recognizing and applying complex relationships within the data it has processed.

⟶ The Importance of CS in the Advancement of K-12 AI Literacy

CS classrooms are uniquely positioned to advance K-12 AI literacy and foster future-ready skills such as computational thinking, creativity, and collaboration. Here's what sets CS classrooms apart:

- **AI is CS**
  From neural networks to algorithms and data processing, AI reflects the tools, methods, and mindsets central to computer science.
- **AI is the newest coding interface**
  Modern IDEs already utilize chatbot assistants and instant code review. Interacting with a model that writes or checks code is as simple as typing; the CS class is an authentic place for students to experiment with this new workflow.
- **CS is a place to connect code to consequence**
  Core CS topics, such as algorithms and data, offer a lens for students to engage with ethical questions around bias, privacy, and fairness, all while exploring the real-world impacts of computing.

Avery's Story

Avery reflected before entering her CS class. Last school year, she had avoided CS—"Too techy," she'd told herself, picturing endless lines of code on a dark screen. But over the summer, her social media feeds had filled with posts about artificial intelligence (AI).

AI wasn't just a headline anymore; it was in the apps she used, the news she read, even the recycling-tracker app she recently dreamed of building for the community center. Avery didn't just want to swipe through that future; she wanted to shape it responsibly.

The course description for her CS class promised projects on how to make chatbots and to connect AI's social impacts to how it works. CS suddenly felt like an opportunity to understand the tech shaping her life. So, several months ago, Avery signed up for a CS class.

Starting with Big Questions

Avery's CS journey began with lessons that helped foster a responsible and curious attitude toward AI and its role in problem-solving.

Her early experiences included exploring and evaluating AI-generated content to understand how human oversight is often required and improves problem solving with AI. She learned how harmful societal biases are perpetuated in generative AI models and how designers can mitigate those risks. Unlike a typical introductory CS class that starts with programming, she examined topics such as content ownership, ethics, and creativity in the context of AI.

Scaffolding AI Assistance

Building on that foundation, Avery's next steps included learning Python—not just as a programming language, but as a tool for thoughtful problem solving. Initially, she programmed without AI assistance to build a fundamental sense of how code works. However, AI was used in other ways to provide feedback. For example, when Avery's while loop miscounted items, the AI assistant supported her debugging progress until she identified the faulty line. The assistant offered three rewrites to help her in developing the skills necessary for critically analyzing AI-generated code.

Note: Avery's story is fictional. While it is informed by current trends and supported by research where noted, it is not intended as a prediction or recommendation. For counterpoints, see our briefs on Universal Design for Learning and Broadening Participation in CS.

She compared and learned from them, then edited her code herself. The tool saved time; her judgment shaped the answer. AI was also used to generate different approaches to a programming task (i.e., worked examples, including some in different languages) and create multiple representations of code, such as pseudocode or a UML (Unified Modeling Language) diagram, all of which improved her ability to not only write code but also read, evaluate, and explain it (see Jacques, 2023).

Programming with AI

Soon after, she began interacting with AI assistants to brainstorm, write, and refine code, developing a deeper understanding of how to use these tools effectively and responsibly now that she had grasped the fundamentals. These tools provided suggestions as students coded, while also explaining errors (see Hoq et al., 2025, Liu et al., 2025, Demirtas et al., 2025).

Avery didn't see AI as just a tool for quick fixes. AI introduced a more sophisticated way of experimenting with programming. She could rapidly test multiple design paths and explore different ideas, then evaluate how they worked, without waiting for teacher feedback or completing static course assignments. And Avery wasn't just accepting AI output; she was thoroughly evaluating it and learning from it in the process. Avery remembers how her brother had taken the same class years before and was just happy when his code compiled. She was thinking with the mindset of a code editor, not just a code writer.

Building for People

The importance of applying human-centered design (HCD) principles to ensure her programs were inclusive and ethical became a central theme throughout Avery's CS experience. She learned to place the needs, abilities, and experiences of real people at the core of her development process. This meant focusing on user experience and completing iterative cycles of gathering feedback and refining designs (see Shehab et al., 2021).

The class examined accessibility needs, such as checking color contrast and ensuring their projects were compatible with screen readers. The principle of keeping a "human-in-the-loop" was consistently reinforced, particularly as they began to integrate AI components and utilize AI code generation tools in their projects.

Lessons on fairness, privacy, and accessibility—especially in the context of AI systems—pushed her to think critically about the societal impact of technology (see Atabey et al., 2025). In multiple projects, the class even recognized that AI tools aren't 100% accurate, and on more than one occasion, discussed the capabilities and limitations of AI systems.

As she iterated on her projects, Avery wasn't just learning to code—she was learning to design with purpose, persist through complexity, and consider how both her decisions and AI's outputs could serve or harm others (see Ryoo, 2019).

Cultivating A Responsible Attitude Toward AI

For Avery, leveraging these tools was a lesson in itself. It helped her understand how to stay current with the fast-paced nature of technology development. More importantly, it taught her to lead tools effectively and not let them lead her.

⟶ Terms to Know

Human-centered design (HCD) is a problem-solving approach that prioritizes the people affected by a solution. It involves deeply understanding their needs, desires, and challenges throughout the design process.

She learned to utilize AI assistants to augment her capabilities, automate routine tasks, and explore complex problems more efficiently, all while maintaining critical oversight and making informed decisions about the technology's application (see Cambaz & Zhang, 2024). Not only did she learn to view computer science as a creative, ethical, and AI-informed endeavor, but she also developed the skills and mindset to use AI thoughtfully in other aspects of her life.

Exploring AI and Data Science

Avery's teacher often presented the class with a reverse-engineering challenge. One day, Avery investigated how one of her favorite video apps recommended content to her. She diagrammed input-model-output flows and hypothesized where biases might be introduced (see the AILit Framework). These challenges altered her perspective on AI. AI tools were no longer mysterious black boxes, but systems people can interrogate and improve.

These tasks also naturally led to more in-depth data investigations. Moving beyond simple classroom datasets, Avery engaged with larger, public datasets. For example, Avery's team once worked on a project that analyzed air quality in their region using the USA's Environmental Protection Agency Air Quality System data. They identified gaps in the data, such as missing values or counties with no air quality monitors ("missing voices"), and removed noise, including blank cells, before analysis. Finally, they discussed how missing data or biases within a model's training data affected the performance of AI models.

⟶ Ethics and Societal Impacts

Woven throughout the course were ethical considerations and the societal impact of technology and AI. These discussions weren't confined to a separate module; instead, they arose organically from the projects that students were developing. As they built tools, they confronted questions of bias in algorithms, data privacy, and fairness in AI decision-making. The class regularly paused to consider what responsible technology creation means, so the issues felt more concrete and personally relevant.

For AI literacy ideas, check out our brief "What AI Experiences Are Foundational for Every CS Student?" and the AILit Framework.

⟶ How can experiences with AI develop students' CT skills?

Traditional CS courses develop computational thinking skills through coding. AI activities provide alternative pathways:

- **Data-driven CT**: Grover (2024) highlights opportunities to transition from rules-based CT to data-driven CT. Using tools like Teachable Machine, students can train models to sort images of local flora and debug misclassifications (Vartiainen et al., 2021).
- **Unplugged AI**: Games like Guess Whose Face (Lim et al., 2024) enable learners to build mental models and manage uncertainty as they identify a hidden face, sharpening algorithm design skills and evaluating datasets, all without coding or the use of screens.

Computational Thinking and Collaborative Judgment

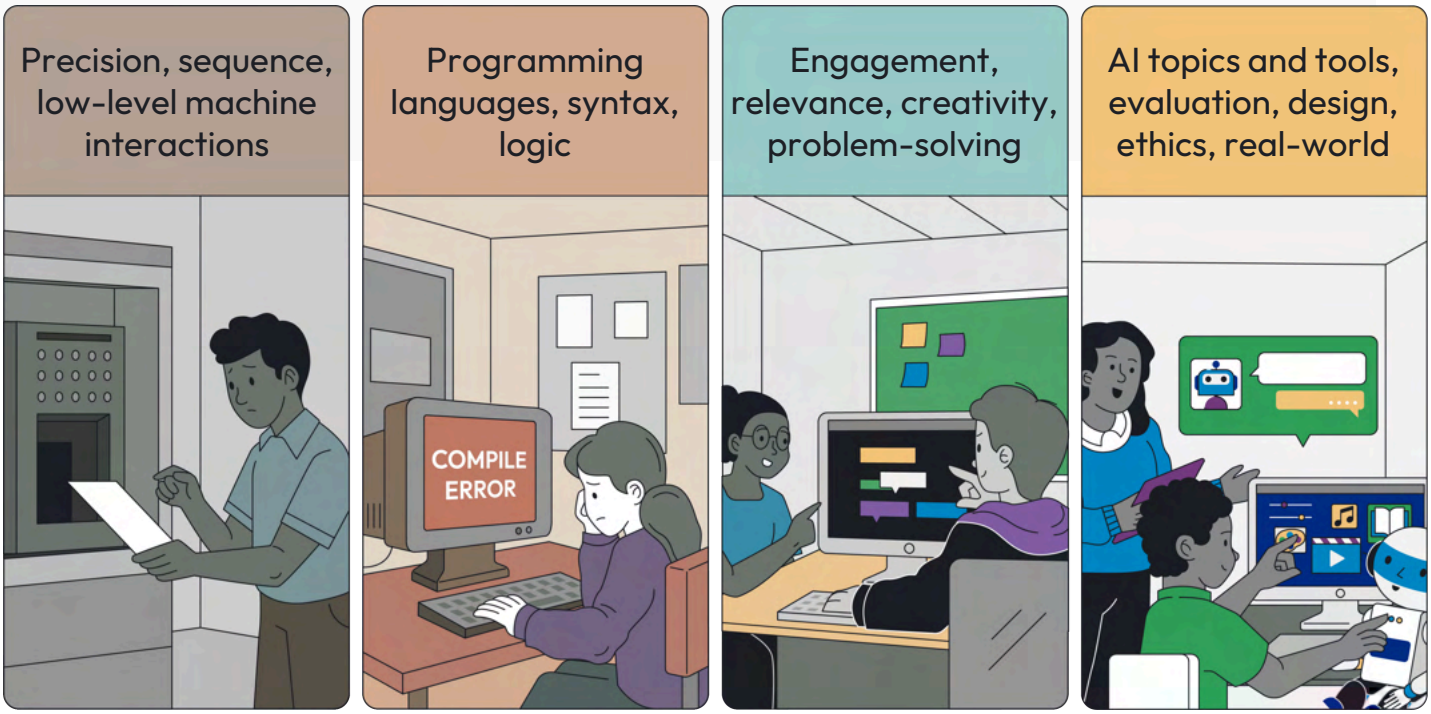
The skills that Avery and her classmates developed through a broad range of analytical and collaborative activities in their CS classroom were valuable in various domains. They expanded their application of computational thinking (CT) to now include AI-rich contexts (or, as Tedre and colleagues call it, CT 2.0).

Students also exercised roles that showcased **human-AI collaboration** (see Grover, 2024; Kim, 2023). For example, teammates rotated between four roles when collaborating with AI on a group project:

- Prompt Lead: crafts precise questions for AI tools.
- Skeptic: evaluates AI output and explores edge cases.
- Bias Scout: checks data and output for fairness.
- Quality Assurance Captain: writes automated tests.

The teacher's goal with this rotation approach was to give Avery and her classmates practice in experimenting, planning, and explaining ideas to one another (see Brennan, 2023) and adopting habits that mirrored industry practice (see Barros et al., 2023). Avery loved the Skeptic role and asking questions like **"What if the input field is empty?"** and **"What if the user speaks another language?"** Just as important, she also grew comfortable hearing feedback when roles shifted and her code landed under another student's microscope.

Computer Science Education Evolves



⟶ A Teacher's Evolving Role

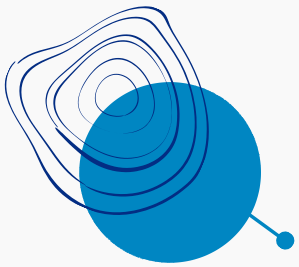
Consider Mr. Sakowicz's advanced CS classroom in Miami-Dade County. Here, a student is deeply engaged in developing an app that transforms audio input into AI-generated sign language visualizations — a project born from a desire to bridge communication gaps. While AI tools may assist students with initial code drafts or debugging, Mr. Sakowicz guides them through the more nuanced, human-centric aspects of the process. He uses the human-centered design process, coaches students through user testing and feedback cycles, and guides critical discussions on the ethical implications and real-world impact of students' projects. AI handles some routine tasks, freeing him to foster essential skills like critical thinking, design thinking, and project management.

> "These are the activities I want to see more of," Mr. Sakowicz noted. "It is motivating to see students doing well and using AI tools appropriately, effectively using [the tools] to propel them and finding intrinsic motivation to persist in computer science. If I, the teacher, can harness that and build on that, I am energized to see what my future CS classroom can look like."

This example points to potential shifts in CS instruction. Teachers might:

- Move from delayed grading to providing real-time support using AI checkers.
- Shift from demonstrating singular "correct" solutions to facilitating exploration, guiding discussions on effective AI prompting (for example, using guidance from resources like Porter and Zingaro's Learn AI-Assisted Python Programming), and navigating AI-assisted problem-solving sessions with multiple pathways.
- Leverage AI to help generate multiple representations of difficult CS concepts, making abstract ideas more tangible and accessible for diverse learners (see our brief on "How Can AI Support Universal Design for Learning?").

Although AI can manage data and computation, teachers still provide the "relational intelligence" that technology lacks—empathy, collaboration, and ethical reasoning (see Hau, Love to Learn). By guiding students to consider societal impacts and develop innovative solutions that benefit humanity, teachers help preserve the critical human element in an increasingly AI-driven world.



Conclusion

As AI expands its reach across the social, cultural, and relational dimensions of our lives, CS education can evolve in tandem with AI by preparing students to influence how AI systems behave and impact others. Avery walked out of CS with a full portfolio of projects, a habit of asking tough questions, and the sense that she can shape the systems shaping her world in return.

Avery didn't just learn how to code with AI tools; she learned how they worked, when to question them, and how to shape the role of AI in society.

⟶ TAKE ACTION

CS Education in an Age of AI is...	
More...	**Less...**
Students interpreting suggestions and refining their questions before seeking further assistance from the teacher.	Teachers answering every "Why won't it run?" question.
Providing an AI coach that answers questions in plain language, shows a short code snippet, and can translate or scaffold explanations for multilingual learners.	Providing static reference sheets and slides for syntax help.
Building on foundational programming skills to read, refine, and evaluate AI-generated code across many programming languages.	Writing code in a block or text-based programming language.
Pair programming with classmates and AI: one student prompts, one student evaluates AI-generated code, both decide what to keep.	Programming by yourself.
Leading short investigations across tools—robots, vision sorters, traffic-light controllers—to reveal how different AI systems work and where they fail.	Using chatbots or voice assistants as the only examples of AI.
Running feedback loops early and often, and adding accessibility checks (contrast, alt text, and screen-reader test) while AI speeds up prototype development.	Students demoing projects to their peers only at the end of a project.

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

What AI Experiences Are Foundational for Every Student?

Foundational AI experiences turn curious AI users into confident AI creators. When CS teachers help students identify AI, evaluate its outputs, build a simple model, and enhance their projects with AI, students develop the insight and judgment to not only use AI responsibly but also shape AI's behavior.

Introduction

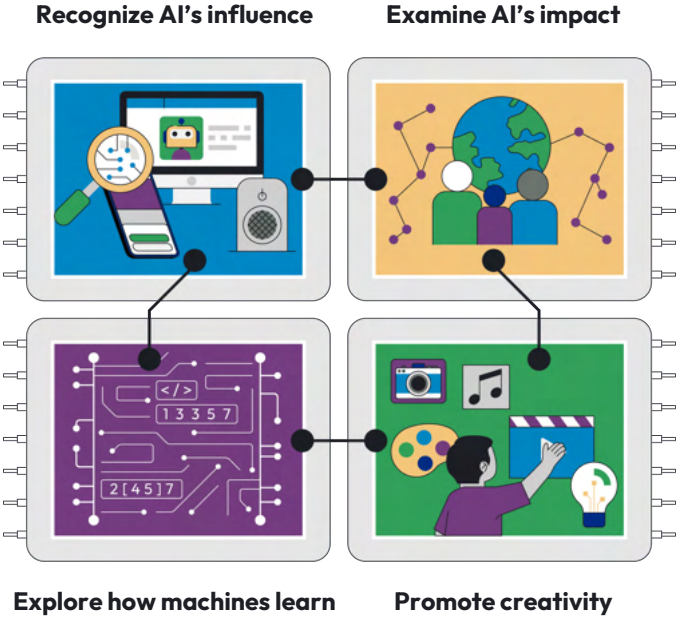
Ask students where they experience AI, and they may point to a phone that translates speech, a camera app that tags friends, or an opponent in a video game. These everyday use cases highlight why students need intentional AI experiences to develop essential dispositions for interacting with AI such as:

- **Technical Insight:** Understanding how data flows through a model.
- **Critical Judgment**: Deciding when to trust, challenge, or refine AI-generated outputs.

CS classrooms are the natural place for these experiences, not as supplementary activities, but as integral, core content that enriches existing instruction in algorithms, data, and ethics. For more on why CS is well-positioned to provide students with foundational AI learning opportunities, see our brief on "How Might Learning Computer Science Evolve in an AI-Driven World?"

This brief highlights four types of foundational AI experiences[1] for all students. The experiences are numbered for clarity, not priority or preference.

1. Recognizing AI's Influence
2. Examining AI's Societal Impacts and Ethical Considerations
3. Investigating How Machines Learn
4. Leveraging AI for Creative Problem Solving



[1] See the Executive Summary for details on how we selected tools discussed in the briefs. In addition, we use Primary Grades as those with students aged 4-11 and Secondary as those with students aged 12-18.

⟶ AI Unplugged

New technologies can exacerbate the digital divide, and mitigating this risk begins with ensuring connectivity and access. Even when students have access to technology, a gap can emerge between those who have the opportunity to engage in active, critical, and creative use and those who are restricted to passive use (Margolis et al., 2017). Where infrastructure or local policies restrict the use of AI tools, unplugged activities offer meaningful ways to learn about AI and its impacts.

In primary grades, hands-on, unplugged experiences can spark curiosity and build a strong foundation for later technology use. For each of the foundational AI experiences presented below, we offer suggestions for unplugged activities, including role-playing, group discussions, and card games.

⟶ AI and CS Standards

Revised CSTA K-12 Standards, slated for launch in the summer of 2026, will define foundational learning outcomes related to AI across grades K-12. AI learning priorities that will be incorporated in these Standards can be previewed in a 2025 report from CSTA and AI4K12. Education systems developing AI-related standards and revising their CS standards should rethink the relevance of existing CS topics and accelerate revision cycles so that curricula keeps pace with rapid changes in AI.

In 2018, AI4K12 set the stage for teachers to incorporate AI concepts and experiences in the classroom by providing a rich resource bank for educators. TeachAI and CSTA thank Christina Gardner-McCune, David Touretzky, and the AI4K12 working group for their invaluable work toward incorporating AI into school curricula.

Experience 1: Recognizing AI's Influence

Students bump into AI every day—filters that guess their mood, playlists that predict the next song, and chatbots that answer homework questions. Classical CS concepts remain essential to CS education, but they no longer fully explain the digital world in which students live. Popular misconceptions about AI abound, making it increasingly important for computer science students to **recognize when AI is present, understand its role, and determine its influence**. This will help them apply their CS content knowledge and critique the growing presence of AI in society.

Key Learning Goals

- **Identify AI in everyday life**: Students identify the type of AI (e.g., predictive, generative) in tools they interact with regularly, such as social media filters or facial recognition systems.
- **Debunk misconceptions**: Students learn what AI is and what it is not, to distinguish between AI-powered technologies and non-AI processes in their lives. They also learn about AI's limitations and the role humans play in ensuring the appropriate implementation of the technology.
- **Develop critical reflection skills:** Students reflect on how AI influences their lives and explore the benefits, risks, and ethical implications of AI.

Sample Activities and Resources

- "How AI Works" Video Series, **Code.org** (2 weeks) All Ages Students build an understanding of machine learning, neural networks, computer vision, and algorithmic bias through collaborative, hands-on activities and simulations paired with videos.
- AI Literacy with Minecraft, **Microsoft** (15-60 minutes) All Ages Learners watch short videos on core AI ideas, then tackle real-world challenges inside custom Minecraft worlds.
- AI + Me, **ReadyAI** (60 minutes) Primary Grades Students earn badges for learning AI4K12's Five Big Ideas in AI, drawing on familiar technologies, including streaming video services and home voice assistants, to discover what AI can and cannot do.
- What is AI, **DAILy Workshop** (15-60 minutes) Secondary Grades Students connect AI concepts to systems they already know, such as social media feeds and camera filters, and explore how AI uses datasets, algorithms, and predictions by discussing examples of AI and engaging with interactive tools.
- Unplugged Introducing AI, **Expressive Machinery Lab at Georgia Tech** (60 minutes) Secondary Grades Students explore foundational questions about AI, prompting them to reflect on their experiences with AI, analyze examples, and imagine future AI applications.

Experience 2: Examining AI's Societal Impacts and Ethical Considerations

Students need to learn to decide whether and when AI **should** be used to address a situation, not just **how** to use it. CS education is uniquely positioned to foster early conversations about AI's role in shaping our digital and physical worlds. With every CS project, teachers can guide students to ask: **Which data trains our model? Whose values guide design decisions?** Grounded in algorithms and debugging, the discussions can expand to include fairness and real-world consequences. For more on addressing the societal impacts of AI, see: How Can Students Become Critical Consumers and Responsible Creators of AI?

Key Learning Goals

- **Analyze AI's impact on society**: Students examine how AI affects jobs, privacy, inequality, the environment, and power dynamics.
- **Understand and address algorithmic bias:** Students discover how biases in data and algorithms can perpetuate injustice and inequality, and they learn how to design more equitable AI systems.
- **Develop ethical reasoning**: Students build a strong ethical foundation to assess when and how AI should be used, focusing on accountability and fairness.
- **Foster metacognitive skills**: Students reflect on how they engage with AI outputs, developing durable skills in questioning, evaluating, and thoughtfully interacting with technology.

Sample Activities and Resources

- Unplugged Construct an AI Bill of Rights for Your Classroom, **EngageAI Institute** (120 minutes) All Ages Students learn about major AI ethics topics, linking them to their own experiences, and design their own "AI rights" to showcase on posters.

- Share Your Voice, **Code.org** (45 minutes) Primary Grades Students choose a medium (e.g., letter, speech, song, poster) to explain whether and how AI tools should be used in schools.

- Bias In, Bias Out, **Raspberry Pi** (60 minutes) Secondary Grades Students create an ML model that classifies images, describe the impact of incomplete data on the accuracy of the model's outputs, explore the need for fair and high-quality training data, and learn how bias influences the predictions generated by a model.

- What principles should guide ethical use of AI?, **Stanford CRAFT** (60 minutes) Secondary Grades Students consider the ethical challenges of AI within a gamified lesson centered on critical thinking, collaboration, and decision-making.

- Unplugged AI Audit, **MIT Media Lab** (60 minutes) Secondary Grades Students play "AI Audit," a competitive card game in which they act as AI startup founders who build AI-powered technologies while identifying and mitigating potential harms to their systems.

"An increasing number of sociotechnical systems are data-driven, and machine learning is code for 'data-driven.' There are benefits—cancer research and treatment, language translation, AI tutoring, and better and more differentiated instruction. But there are harms where we may decide as a society that we want a different way of these systems functioning, including without machine learning."

- **Ben Shapiro**, University of Washington

Experience 3: Investigating How Machines Learn

Training a machine learning (ML) model offers students and educators the opportunity to connect core CS concepts with technical understandings of emerging technologies. When students build and train their own models, they learn how data is collected, classified, and used to generate outputs. Additionally, they learn about representation and reasoning in various AI models, such as classifiers (e.g., spam filters), predictors (e.g., forecasting tools), and recommenders (e.g., movie suggestions). Understanding how AI "learns" will help students recognize its limitations and empower them to question AI's outputs.

"We are the data. Students need to reflect on how models are trained and where the data comes from."

- **Christina Gardner-McCune**, Associate Professor, University of Florida and Co-Lead of AI4K12

Key Learning Goals

- **Understand how AI learns from data:** Students build an understanding of the core principles of machine learning by training models and seeing firsthand how data influences decisions.
- **Explore representation and reasoning in AI models:** Students examine how abstract representations in AI models focus on key features, guide decision-making, and shape outputs in AI-enhanced technologies.
- **Evaluate fairness and bias:** Students assess the accuracy and fairness of AI outputs, understanding how bias enters AI systems and manifests as real-world impacts.
- **Build data literacy:** Students learn how to work with data—a critical skill for making informed decisions in an AI-driven world—and discover how the quality of AI's training data significantly impacts its outputs.

Sample Activities and Resources

- **Micro:bit CreateAI, Micro:bit Educational Foundation, and Aarhus University** (60+ minutes) All Ages Students train a computer to differentiate between physical movements like clapping and waving, using physical computing to understand ML concepts.

- **AI for Oceans, Code.org** (45 minutes) Primary Grades Students train AI to distinguish fish from non-fish, discovering how AI models function and adapt while learning about the role of training data in shaping AI outcomes.

- **Machine Learning for Kids** (20+ minutes) Secondary Grades Students train a computer model to recognize certain objects and then use the model to develop an interactive game in Scratch.

- **How to Solve Problems with Machine Learning Models, Raspberry Pi** (60 minutes) Secondary Grades Students train and test a model to solve a real-world problem from a user-focused perspective. They learn how to define the problem, prepare data, and order the stages of an AI project lifecycle before testing their model and reporting on its accuracy.

- **Unplugged** **Guess Whose Face, NCSU** (60 minutes) Secondary Grades Students explore facial recognition technology by using tracing paper to extract facial features from familiar images (e.g., cartoon characters or celebrities) while others guess the face's identity.

Experience 4: Leveraging AI for Creative Problem Solving

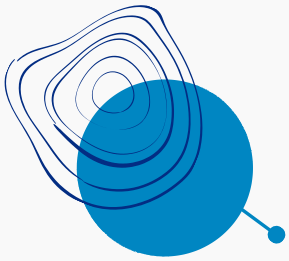
Students of all ages can explore AI's capabilities as a creative "partner" to iterate and refine code, art, or project designs. This experience serves as an easy entry point to foster curiosity about AI's various use cases, and it encourages students to assess and evaluate AI's outputs. Beyond the classroom, students will need to know how to work with AI tools to enhance their skills while maintaining ethical oversight and creative control. "Collaborating" with AI on creative projects will help demystify the technology for students and introduce discussions about ethical partnerships, appropriate attribution, and intellectual property. See our brief "How Can AI Enhance Creative Thinking in Computer Science?" for more details.

Key Learning Goals

- **Use AI to solve problems and refine solutions:** Students use AI tools to enhance their work, whether in art, coding, or project design.
- **Think critically about AI's role in creativity:** Students consider how much creative leeway AI should have and develop a sense of ethics in working with AI. They explore the distinctions between AI and human creativity, recognizing that AI possesses unique abilities but lacks human qualities such as empathy and intuition.
- **Understand intellectual property in AI-generated content:** Students explore the ethical and legal implications of AI-created works, focusing on ownership and fair use. They consider how AI tools use existing data without permission to produce new content, examine the impact on artists' rights, and debate attribution and ownership of AI-generated creations.

Sample Activities and Resources

- Intro to Creative Coding, **Code/Art** (45-60 minutes) All Ages Students learn how to code computer programs that create abstract art, games, 3D models, and more. Free resources can be found on Infosys Pathfinders or their website.
- AI-Generated Art, **DAILy Workshop** (15-95 minutes) Secondary Grades Students explore how AI can generate visual art through systems like Generative Adversarial Networks (GANs), considering the role of human and non-human contributors in the creative process.
- AI & Drawing "Quick Byte" Lesson, **AI4ALL** (60 minutes) Secondary Grades Students use Google's Quick, Draw! to learn core AI concepts, including how AI processes data, how humans input biases, and how AI is used in diverse fields like art.
- Ownership, Ethics, and Creativity, Code.org (45 minutes) Secondary Grades Students explore the ethical tradeoffs of using AI in creative contexts by engaging with real-world controversies involving ownership, attribution, and compensation.
- Unplugged Mission Agent Training, **University of Florida** (60 minutes) Secondary Grades Students simulate the phases of chatbot training by role-playing as agents, users, and developers.



Conclusion

AI is increasingly woven into the fabric of our daily lives: suggesting songs, fixing photos, answering code questions. Educators should equip every student with the knowledge to understand how these AI tools function, how to collaborate with them effectively, and when to question their outputs critically. By thoughtfully integrating the four experiences into existing lessons, we can strengthen core computer science concepts while empowering students with the essential insight and judgment they need to navigate and shape an AI-driven world.

⟶ TAKE ACTION

1. Replace a starter question with a 10-minute task in which the class traces the input → model → output path inside a favorite student app.
2. During any data or machine learning project, pause for a quick "What voices are not represented in our dataset?" discussion.
3. Identify one unit where a week-long AI build project (e.g., Machine Learning for Kids) can replace an older project.
4. Share and reflect on your experiences and your students' experiences with colleagues and your CSTA chapter!

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

How Can AI Enhance Creative Thinking in Computer Science?

Students—not algorithms—generate true originality. As computer science teachers design project-based experiences and integrate them into classrooms, AI can widen the range of creative possibilities and provide timely support without replacing the human creative spark.

Introduction

Creative thinking is the human capacity to generate, evaluate, and improve ideas that are both original and useful (The PISA 2021 Creative Thinking Framework). This capacity is rooted in human experience and motivation, not in algorithms. AI cannot feel curious or judge social impact. What it can do is:

- devise more starting points than a single mind can conceive;
- give instant, low-stakes feedback so students iterate faster; and
- free teacher time, so pedagogy — not tool use — remains central.

⟶ Enhancing Pedagogical Approaches with AI to Fuel Creativity

Approach	What We Are Doing Now	AI Enhancements*
Problem-Based Learning	Teachers pose open design challenges; students research, prototype, and test.	A code-aware chatbot fields routine syntax questions, letting the teacher coach human-centered design.
Design Cycles	Students iterate on code through time-consuming peer review cycles.	A student evaluates an LLM's categorization of peer comments into "keep," "refine," and "explore" buckets and identifies what they will act on.
Culturally-Relevant Projects	Classrooms explore culturally relevant contexts that draw on local examples.	Generative AI tools can modify traditional projects to incorporate AI-related contexts that are relevant to students' communities.

*These pedagogy enhancements focus on generative AI.

Classrooms that prioritize pedagogy (problem-based learning, design cycles, culturally relevant projects, etc.) give students the agency that fuels creativity. When classrooms merge those strong, project-based methods with targeted AI support, students keep ownership of ideas while enjoying a larger, more navigable array of approaches (Viruel et al., 2025).

AI-enhanced CS learning goals allow students to exercise their creative thinking by:

- crafting meaningful solutions to complex challenges;
- bridging CS with art, science, and humanities; and
- infusing personal perspectives into their projects.

Thoughtful integration of AI into CS projects ensures that creativity remains the domain of humans while AI serves as an accelerator of student ideas.

AI-Powered Creativity in Computer Science

AI doesn't replace creativity—it reflects, reshapes, and sometimes redirects it. It allows students even more ways to express themselves and engage their interests.

Three Ways AI Can Support Creativity

Goal One: Craft Meaningful Solutions to Complex Challenges

Crafting meaningful solutions typically begins with a coding problem assigned by a teacher (e.g., build a flood-alert map or a recycle sorter) with a short design cycle and limited data. AI can help students take ownership of the problem-finding phase of their projects (Kim, 2011) and incorporate more expansive, more relevant data. Tools such as MagicSchool's MagicStudent can expand their brainstorming list, while a weather-forecast API or a text-to-table generator supplies rich, edge-case datasets in minutes.

Tools such as Jippity and Code Breaker Byte can explain unfamiliar code snippets or propose alternative algorithms, allowing teachers to focus on the ethical impact, user feedback, and design iteration. A quick assessment prompt—"What trade-off did you choose and why?"—captures reflection for a process-based assessment rubric that values both creative reasoning and working code.

Goal Two: Bridge CS with Art, Science, and Humanities

Interdisciplinary projects often rely on stock images and manual mash-ups, which can divert time from deeper inquiry, domain connection, and individual creativity. AI widens the palette: students can feed their sketches to an image model in tools like Canva, debate style appropriation using current artist copyright cases (motivated by articles like Harvard Business Review's "Generative AI Has an Intellectual Property Problem"), or even code a website into a creative art piece like an interactive mural or data-driven animation that tells their story (with support from the resources at Code/Art). They can turn poem lines into soundscapes or sensor logs into music using multimodal generators, revealing code as a medium rather than a barrier.

Projects that require students to draw by hand can use programs such as Cand.li, which encourages artistic expression and offers AI-powered tutorials to guide the learning and teaching of game design, art integration, and coding.

⟶ Real-World Inspirations

Fashion designer Gala Marija Vrbanic uses AI to create digital clothing designs. See NPR for an inspiring profile.

Goal Three: Infuse Personal Perspective into Projects

Most students personalize code only at the surface level, by renaming variables after pets or swapping images in a template, for example. Many may hesitate to create a larger narrative or advocacy project because the blank screen feels intimidating. AI is already reducing that barrier. Generative art tools (e.g., Adobe Firefly, used in Humble ISD's Digital Creator Fest) enable learners to transform their ideas and imagination about historical figures into portraits. Students can experiment with different descriptive words, building analytics skills as they troubleshoot their prompts and bring their vision to life.

In MIT's 2024 Generative-AI Mobile App workshop, a novice coder built a "Lucky Color" app using App Inventor. The "Lucky Color" chatbot suggests a random lucky color, generates matching images with an App Inventor imagebot block, and then displays where to find the items visualized. The approach allowed the student to code an app that reflected a personal ritual, such as choosing clothing each morning, in a creative way (Kim et al., 2024).

This goal directly aligns with CSTA Standards for CS Teachers (e.g., Standard 5c), which promote student self-efficacy by fostering creativity, offering choice in both product and process, and encouraging self-directed learning.

"For young people, staring at a blank screen is very, very scary. AI can take that fear and anxiety away and allow kids to pick up problems that seem big and scary, fostering individuality and creativity through coding."

- **Ben Forta**, Adobe Senior Director of Education Initiatives and CS Teacher for over 30 years

Safeguards That Keep Students Thinking

- **Practice Human-in-the-Loop Skills:** Require students to annotate every AI suggestion they accept or reject, explaining their reasoning.
- **Focus on Coding Foundations:** Continue to explicitly teach algorithms, data structures, testing, and debugging. AI autocomplete never substitutes for that fluency, as emphasized in the upcoming 2026 CSTA Standards revisions.
- **Monitor Cognitive Off-Loading:** Offer regular unplugged design sessions to help restore balance. Studies link heavy reliance on AI to "metacognitive laziness"—a decline in self-regulation and critical thinking (Fan et al., 2024).
- **Build an "AI pause" routine:** Instruct students to justify any AI-generated code or art before it becomes part of their final project. This keeps the agency with the learner and can help address concerns about overreliance.

⟶ Human-in-the-Loop Skills

These skills involve actively engaging with AI suggestions—interpreting, testing, and adapting them—to create solutions that reflect human values. For example, when an AI tool suggests code changes, students assess their quality, experiment with modifications, and integrate their insights, ensuring that AI remains a tool that enhances rather than replaces their problem-solving.

Conclusion: Embracing the Future of Creative CS Education

Put simply: students create, AI accelerates. When teachers couple solid CS foundations with human-in-the-loop routines and effective, creativity-inducing pedagogical approaches, AI can amplify originality, agency, and joy in coding. This is exactly the kind of creative future our CS classrooms aim to cultivate.

⟶ TAKE ACTION

1. Pick one project that your students already do and add just one step where AI can help, such as brainstorming ideas or getting instant feedback on code for their project.
2. Develop a rubric that rewards the process of creation—inquiring, imagining, iterating—rather than just the finished code.
3. Talk with your colleagues about how AI is impacting creativity in your classroom and collect feedback from your students about whether AI made their projects feel more personal or if it helped them express their ideas in new ways.

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

How Can AI Support Universal Design for Learning?

Universal Design for Learning (UDL) provides computer science (CS) teachers with a blueprint to reach every learner. AI provides supports, such as captions, code hints, and voice input, that turn barriers into choices. When teachers guide these tools with clear goals and care, students with and without disabilities gain full, supportive access to computer science.

Introduction

Picture a secondary school computer science lab. Alex depends on captions, Sam navigates with a switch device, and Ángel loses track of multi-step tasks unless directions stay on screen. Students like Alex, Sam, and Ángel each have unique learning needs. However, typical CS tools, such as standard Integrated Development Environments (IDEs), often reward fast typing, sharp vision, and flawless working memory. This mismatch between student needs and tool design creates a significant hurdle. It's a key reason we observe an underrepresentation of students with disabilities in CS, especially when considering how students with disabilities often have a negative perception of CS and how teachers sometimes view the abilities of these students in the CS classroom (Blaser et al., 2024; Israel et al., 2022).

"Everybody benefits from Universal Design for Learning. Everybody benefits from equitable and inclusive access."

- **Rudy Escobar**, Project Coordinator, STEM, Stanislaus County Office of Education

⟶ Principles of Universal Design for Learning

CAST's UDL framework aims to increase inclusion in education by ensuring, from the outset, that learning is meaningful and accessible to all students. The framework is built on:

- **Engagement:** Design options for recruiting and maintaining student interest, sustaining effort and persistence, and building emotional capacity.
- **Representation:** Design options for presenting new content to students based on different ways of perceiving and receiving information.
- **Action and Expression:** Design options for interacting with content and sharing what is learned.

For more information about using AI to support UDL, visit CAST's AI & UDL website.

Universal Design for Learning (UDL) asks teachers to plan for varied needs from the start and offer more than one way to perceive content, participate in the work, and show understanding. The UDL approach has proven to be a strong foundation for more accessible CS instruction (Hutchison et al., 2022). A growing body of work has focused on integrating UDL into K-12 CS education (e.g., UDL in CS Remix Table, UDL4CS). Advancements in AI tools can support teachers' lesson planning by:

- supplying on-demand **academic support**,
- lightening **executive-function** load, and
- providing flexible **sensory and physical options** in CS activities.

This brief outlines concrete steps—first-week tweaks, low-prep add-ons, and caution flags—to integrate AI into the UDL processes without compromising your teaching voice or student privacy. UDL, combined with AI, will not replace individual accommodations, but it can lower barriers and enable more students to learn computer science.

Enhancing Academic Support

Programming languages like Java can be particularly challenging for students with dyslexia or other cognitive processing difficulties, due to their complex logic and symbols (Ladner, 2020). Many teachers feel CS curricula lack sufficient differentiation options that could address these and other challenges (Blaser et al., 2024).

Customizing CS instruction to provide greater academic support starts with a clear view of a student's level of understanding. AI-fueled assessment tools, such as GitHub Classroom's auto-grading, provide rapid, precise feedback on coding assignments, quickly identifying where a student is struggling with syntax or logic. With more precise assessment data in hand, teachers can use AI to create more options for presenting learning materials and engaging students, including more choices in software, topics, and activities.

A few examples of AI enhancing academic support:

Engagement	Representation	Action & Expression
Interest-Based Learning: AI features in tools like Jippity can suggest projects tailored to students' interests, such as creating a program related to their favorite hobby, fostering curiosity and persistence.	**Multilingual Support:** AI tools like Google Translate can translate learning materials into multiple languages or provide subtitles in real time, enabling greater accessibility for English language learners.	**Feedback Systems:** AI provides immediate, personalized feedback on coding assignments in IDEs like GitHub Classroom (13+), helping students identify errors and improve their work iteratively.

*Icons in the table were retrieved and modified from udlguidelines.cast.org, ©CAST, 2025.

⟶ Counterpoint Considerations: - Watch for Bias in AI Feedback

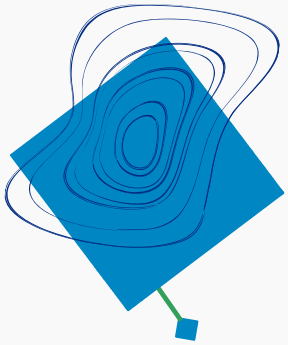
Large language models (LLMs) have been found to contain ableist biases that are rooted in their training data (Glazko et al., 2023). They have also been shown to mirror harmful ableist stereotypes (Gadiraju et al., 2023). Perpetuation of bias by AI presents a risk to students in any classroom, but for students with disabilities, perpetuating ableist biases could reinforce misperceptions about who "belongs" in CS. For example, students with disabilities can be excluded from CS because of a misperception that they don't have the capability to participate (Blaser et al., 2024).

As Sofía de Jesús, Associate Program Manager at Carnegie Mellon University's Computer Science Academy, highlights in our brief "How Might AI be Used to Broaden Participation in Computer Science?," it's often best to feed only the error message or two or three lines of code into AI tools. This "small-chunks" approach is especially helpful when using tools like Google Translate to provide feedback or translate code errors into a student's native language. The approach reduces AI "hallucinations" and lowers cognitive load, providing better support for the learner.

⟶ Long-Standing Challenges to More Accessible CS:

Many traditional CS classrooms present persistent barriers for students with disabilities, including:

- Websites and tools that are inaccessible (e.g., incompatible with screen readers, limited keyboard navigation).
- Learning materials that require visual prompts or demonstrations.
- Uncaptioned videos.
- Inflexibility of physical tools, such as keyboards or mice.
- Lack of targeted support for students with dyslexia, dysgraphia, or other disabilities.



Augmenting Executive Function Support

> "As we personalize, we could expand the achievement gap by over-scaffolding. If a student needs repeated practice, it doesn't mean problems should get easier."
>
> - **Maya Israel**, Professor, University of Florida

The many rigors of computer science—attention to detail, abstract thinking, and time management, to name a few—can lead to frustration for students with executive function challenges. Students must manage their time effectively, plan projects efficiently, and persist through challenging tasks. These requirements can become barriers for students who struggle with executive functions (Blaser et al., 2024).

AI tools provide practical support by helping students organize their work and monitor their progress. Research shows AI-driven tools can lower anxiety related to project management and boost students' confidence (Glazko et al., 2023). Additionally, intelligent agents support collaborative coding by prompting students to ask deeper questions and reflect more effectively on their work (Earle-Randell et al., 2024).

A few examples of AI supporting executive function include:

Engagement	Representation	Action & Expression
Organization and Reflection: AI-powered tools like Goblin.Tools can help students break tasks into smaller goals, set reminders for deadlines, and manage their schedules. Platforms like Riff encourage students to reflect on learning choices, aiding self-regulation.	**Simplified Explanations:** Generative AI can help teachers rewrite complex instructions in simpler language or create step-by-step guides, ensuring clarity for all learners.	**Project Planning with AI:** Generative AI can help teachers brainstorm and structure coding projects, even for block-based environments. It offers students alternatives to choose from and helps them develop project design skills.

⟶ Counterpoint Considerations: Balance AI Scaffolds with Independent Thinking

Personalized instruction that provides options for engaging and interacting with learning materials can be beneficial to diverse learners, but there is the risk that personalization results in inadequate instruction (Bulger, 2016). In CS education, teachers are particularly concerned that students could become overly reliant on AI at the expense of learning foundational CS skills. Overreliance was among the three most frequently cited concerns by CS teachers in the 2024 CSTA/TeachAI survey.

AI can provide organization, reflection, and project planning aids, but be aware of overly relying on AI support. Encourage students to take ownership of their learning and build self-regulation skills.

Providing Physical and Sensory Support

The hurdles that CS presents for students with challenges related to mobility, vision, or hearing are well documented. Conventional screen readers, for example, do not necessarily convey the full complexity of visual components in coding environments, making them not entirely helpful for blind learners (Stefik et al., 2019). Programming languages designed to be easily navigated with screen readers, such as Quorum, can increase the accessibility of CS learning for students with impaired vision. However, options such as Quorum are not always provided (Ladner, 2020).

AI can enhance the performance of assistive technologies that support vision, hearing, and mobility needs, thereby expanding students' options for interacting with content and sharing their learning. Although research has demonstrated that UDL principles may be underapplied, particularly in offering multiple means of action and expression (Israel et al., 2020), AI may help bridge the gap by aiding teacher planning and delivering diverse options.

A few examples of AI enhancements to assistive technologies:

Engagement	Representation	Action & Expression
AI-Assisted Accessibility Features: Tools from Microsoft Education and Google for Education have built-in accessibility features (e.g., voice to text, screen reader, Braille display, keyboard shortcuts) that enable personalized settings, helping students comfortably engage with content.	**AI-Enhanced Screen Readers:** Programs like Quorum Studio paired with NVDA screen readers, can describe dynamic visuals, which could make coding environments more accessible for students with impaired vision.	**AI-Enhanced Voice Commands and Speech-to-Text:** Tools like Serenade (13+) allow students with a mobility-related disability to interact with coding environments using voice commands.

⟶ Counterpoint Considerations: Test and Verify Accessibility Tools

AI-enhanced assistive technologies can dramatically improve accessibility. However, teachers should pilot these tools to confirm they effectively support students' needs, ensuring features like enhanced screen reading and voice commands fully capture complex visual elements.

Moreover, AI tools (e.g., those using video, audio, and/or text) may collect personally identifiable information. Before implementing an AI-powered classroom management tool, review the product's data policies with your school's IT team to understand data collection, storage, usage, and deletion practices, including limited or no sharing with third parties. It can be helpful to discuss the pros and cons of the tool with students, parents, or caregivers to maintain transparency and open conversations.

"Students come in at all different levels, and we need many entry points for everyone to get in. AI can be helpful in this, but in most cases, it's something that you need to tailor. AI makes educators more necessary—good teachers are going to be even more important in taking in information, molding it, and making it usable."

- **Allen J. Antoine, Jr.**, Director, CS Education Strategy, Expanding Pathways in Computing (EPIC), Texas Advanced Computing Center

Conclusion

AI will not make a classroom inclusive on its own, but it is a powerful tool that can help remove barriers and enable more students to thrive in CS classrooms. By utilizing AI to provide flexible options for learning and demonstrating understanding, you can ensure that every student, regardless of their needs, has the opportunity to truly participate and succeed in CS. Each small, teacher-led step moves CS toward a classroom where disability never decides who gets to code.

⟶ TAKE ACTION

1. Assess the accessibility of your current CS tools and platforms, and identify alternatives or AI enhancements that align with UDL principles and support all students.
2. Regularly review AI outputs to identify and discuss ableist or harmful stereotypes.
3. Use AI-based assessments (e.g., GitHub Classroom) to quickly identify and address individual learning needs.
4. Implement AI-powered assistive tools (e.g., Serenade AI) and verify effectiveness with students.

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

How Might AI Be Used to Broaden Participation in Computer Science?

Accessibility and relevance fuel broad participation in computer science (CS). When teachers use AI in ways that recognize and support the diversity of their students, they can significantly increase student engagement, inclusion, and achievement for all CS learners.

Introduction

Too many students still miss out on a fundamental computer science education. Despite notable progress in promoting CS in recent years, young women, Hispanic/Latinx students, English Language Learners, students with disabilities, indigenous students, and other student groups continue to be underrepresented in foundational CS courses. Black students now enroll in entry-level courses at rates closer to their overall representation in schools; however, they participate less in Advanced Placement (AP) CS exams (State of Computer Science Education, 2024).

The widespread availability of AI tools in recent years presents opportunities to tackle challenges in CS participation and achievement in new ways. This brief presents the perspectives of five thought leaders who approach these opportunities from different angles—in Latinx communities, Indigenous classrooms, school systems seeking equitable course pathways, girls' entrepreneurship programs, and multilingual settings. Their culturally responsive approaches demonstrate how teachers can turn AI into a bridge, not a barrier, to help grow CS participation by:

- **Sparking curiosity** by linking CS to music, storytelling, and local issues that matter to students.
- **Removing access barriers** by translating content, generating scaffolds, and adapting examples for varied cultural contexts.
- **Supporting project-based learning** approaches by developing teachers' background knowledge, providing code samples, and facilitating formative feedback.
- **Guiding critical inquiry** by helping students inspect bias in data and models and propose better solutions that meet their needs.

⟶ Culturally Responsive-Sustaining Computer Science Framework

The approaches highlighted in this brief align with teaching practices that honor students' cultural backgrounds while developing their CS skills. For more guidance on culturally responsive CS education, see the Kapor Foundation's Culturally Responsive-Sustaining CS Framework, which provides practical strategies for making CS education more inclusive and relevant to all students.

This brief does not present perspectives from every historically underrepresented student group or every perspective within a given group. We recognize that each community has unique circumstances, assets, and needs that shape how AI and CS can best serve their students. This brief is not a comprehensive overview of the disparities, challenges, or potential of AI to broaden participation. It is intended as a starting point for curriculum and professional development providers, as well as a source of inspiration for teachers working to create more equitable and culturally responsive CS learning experiences.

Check out our brief on "How Can Students Become Critical Consumers and Responsible Creators of AI?"

State of Computer Science Education 2024

- Black students, Hispanic/Latinx students, and Native American students are less likely than White students to attend a high school that offers foundational computer science courses.
- Young women still choose foundational high school CS courses far less often than young men.
- Black/African American students make up 17% of enrollment in foundational high school CS courses, but only 7% of participation in AP CS exams.
- Nationally, 11% of students are English language learners, but only 7% of students enrolled in foundational high school CS classes are English language learners.

From Curiosity to Creativity: Using AI and Music to Draw Every Student into CS

Perspective by Allen J. Antoine Jr.

Director of CS Education Strategy for Expanding Pathways in Computing (EPIC), Texas Advanced Computing Center, the University of Texas at Austin; Project Director, Computer Educator Opportunities (CEO) Initiative.

When engaging with school system leaders and administrators, two priorities consistently emerge: cultivating top-tier schools and ensuring all students have equitable access to cutting-edge coursework. While computer science (CS) is not a new field of study, its applications continue to evolve, and the computational thinking skills it fosters can enhance student achievement across multiple disciplines (Lemay et al., 2021). This makes building a successful CS program an ideal focal point for school systems striving to excel. It is essential to implement inclusive recruitment strategies that reach every student and provide them with the support they need to succeed once they express interest in CS courses.

As education professionals, we can sometimes lose sight of what it's like to be a kid. Why do so many primary school students gravitate toward science courses? I would suggest that they are drawn to the sense of wonder that comes from conducting experiments and witnessing natural phenomena that spark curiosity about how and why things work. In CS, educators have long used physical computing lessons and devices to engage students in a similar way (Theodoropoulos et al., 2018). However, I believe artificial intelligence (AI) holds even greater potential to ignite students' curiosity and attract more learners to CS. If our goal is to broaden participation in CS, we must actively focus on reaching student populations that have been historically excluded from the field. But how do we effectively engage these students?

⟶ TAKE ACTION

Build units that pair AI music tools with coding tasks, allowing students to hear their sounds in CS.

Our research highlights a strong connection between minoritized students, music, and cultural representation. My colleagues and I have led workshops focused on music creation using AI tools such as Code.org Music Lab, Flow Scholar, Music FX, and Suno AI. Teachers have reported that these experiences have successfully attracted more students to computing courses (Antoine & Armant, 2024). We strive to cultivate that kind of intrinsic motivation, encouraging diverse students to engage with the content. In another project I am currently leading, Advancing AI Literacy via Teacher-Crafted Inclusive CS Lessons, we are examining the significant influence educators can have by designing AI-based activities that resonate with students traditionally underrepresented in computing. While the potential to use AI as a tool to inspire student participation in CS is immense, it also comes with inherent challenges.

One of the greatest risks of rapidly adopting AI in schools is the potential overreliance on these tools for lesson development. While generative AI can provide teachers with valuable lesson ideas aligned to content objectives, failing to tailor assignments to the specific needs of their student population can lead to teaching practices that reinforce inequity and bias. Just like traditional textbooks and search engine results, AI-generated lessons and activities can reflect the societal biases embedded in the datasets used to train the models. However, this should not deter us from using the technology. Instead, it underscores the importance of using AI thoughtfully and critically.

As educational professionals, we must lead the effort in helping students understand AI and critically examine the biases embedded in this technology. While it's essential to acknowledge that AI is not perfect, its potential benefits for student recruitment and lesson development are undeniable. Skilled teachers will play a crucial role in interpreting AI-generated information and adapting it to create equitable and engaging learning experiences for their students. And I have no doubt that we are up for the challenge!

A Bridge to Access: How GenAI Can Help Multilingual Learners in CS

Perspective by Sofía De Jesús

Associate Program Manager,
Carnegie Mellon University Computer Science Academy

In K-12 computer science education, we have adopted technology broadly and embraced its potential to provide more access and accessibility to students who may not have had access to it before. As with other educational technologies, AI, specifically GenAI, is being sold with much promise for the future of our students and education in general. In analyzing what GenAI is and isn't - and what it can and cannot do - we have to be careful about overpromising on a currently flawed technology while also embracing its possibility.

One of the promising areas of GenAI is its ability to support multilingual learners (MLL; Prather et al., 2024). Often, students who are not native English speakers are not placed in CS or other STEM classrooms due to the language barrier. This is a mistake that can be remedied with more professional development on the impacts of cognitive load on students and how teachers can help support students in advocating for their own needs when their brains are overloaded (De Jesús et al., 2025). In addition, combining theory and practice, accompanied by technology, can help alleviate some of that load for both students and teachers (Ehrensberger-Dow et al., 2020). GenAI can provide us with access to translators that are both fast and helpful in alleviating cognitive overload.

That said, GenAI contains a lot of flawed information, so there may be instances where we see culturally biased responses, translations that may be inappropriate and inaccurate, and other types of bias. Using the technology broadly allows for more of that to happen, but when used in small doses, the results increase in accuracy and decrease in bias. For example, if an AI-powered translator is given a full page to translate, it may produce more errors (or what some call hallucinations) than if it is fed only a few sentences at a time.

⟶ TAKE ACTION

Use AI translators systematically (only a few sentences at a time) to help multilingual learners reduce cognitive overload.

Artificial intelligence has a place in the K-12 computer science classroom. However, it should not be a free-for-all, where anything and everything goes and guardrails are removed in the name of progress. GenAI can and has caused harm to both people and the environment, and it is not always the most appropriate tool to use (Gebru & Torres, 2024). It is important for us to familiarize ourselves with the potential and the harmful effects of AI to make the best use of it.

Because students require foundational computer science knowledge to use AI and understand its impact, initiatives should focus on specific applications and

general AI literacy. It is imperative that students and teachers have all the tools available, including the following:

1. Access to quality computer science education
2. Access to language support for multilingual learners in the computer science classroom
3. Training on how to identify when and how to use available resources for multilingual learners
4. Ongoing professional development that reinforces the use of critical theories, such as Cognitive Load Theory, and how to present that information to students so they can advocate for the tools they need and when they need them

Focusing on practices rather than just the tools will ensure that both teachers and learners have agency in improving their own educational journey.

Empowering Latinx Students Through Culturally Responsive AI Education and Computer Science

Perspective by Rudy Escobar

STEM and CS Coordinator, Stanislaus County Office of Education, CSforCA Co-Chair, CSTA Board, Equity Fellow, Policy Workgroup Member, and LatinX Affinity workgroup lead, ISTE U AI Instructor

Integrating AI into culturally responsive education can increase engagement among Latinx students in computer science by enabling educators to create culturally relevant learning experiences and by empowering students to address real-world issues in their communities.

Research has shown that AI can develop tailored learning modules for Hispanic-Serving Institutions (HSIs), guiding students to identify social issues and apply AI-driven solutions (AAAI, 2023). GenAI tools, such as AIStory, enable students and parents to co-create culturally relevant narratives, making learning more meaningful (Informatics UCI, 2024). AI-driven assessments generate culturally responsive questions, ensuring students see their backgrounds reflected in their education (MDPI, 2024).

A major barrier to Latinx participation in computer science is the hurdle between being a consumer and a creator of technology. The Latinx community often engages with AI tools without fully understanding their underlying mechanisms. Encouraging students to move beyond passive consumption to active creation fosters critical thinking and agency, making AI a gateway to, not a replacement for, computer science, data science, and cybersecurity. Messaging must emphasize the necessity of understanding AI's functionality and societal impact.

AI serves as both an entry point to computer science and a tool for critical analysis of the technology's impacts, development, limitations, and biases. For example, GenAI often misrepresents cultural diversity, such as the diversity within Guatemalan identities, thereby reinforcing stereotypes rather than reflecting true variation. Prompts like "Draw a Guatemalan woman" may initially yield diverse images, but they tend

to become more stereotypical over time. This discrepancy highlights gaps in AI models and the need for improvements. Addressing such biases in AI education ensures students understand who is excluded in AI models and how AI systems can be improved.

⟶ TAKE ACTION

Create AI projects that let students critique bias and design fixes for their community and/or evaluate AI's impact on environmental and natural resources, particularly as it affects underserved communities.

AI can also serve as a bridge for non-English-speaking students and their parents. AI-assisted bilingual coding platforms and chatbots help bridge language gaps and create inclusive learning environments. However, AI-generated translations require verification, as inaccuracies can lead to miscommunication (Hill, D.C et al., 2022), underscoring the importance of using AI as a tool to enhance, rather than substitute for, human qualities in education.

Responsible AI integration is critical to preventing educational disparities. Improper AI use risks widening the digital divide, particularly in underserved communities (Vesna, 2025). AI should support, not replace, teachers by automating administrative tasks, allowing educators to focus on personalized instruction (Luckin, 2018). By integrating AI responsibly, educators can broaden participation in computer science among Latinx students and empower them to move from consumers to creators, fostering meaningful engagement and equitable access to technology-driven opportunities.

Guiding Girls to Build Real-World Tech Start-Ups with GenAI and Project-Based Learning

Perspective by Tara Chklovski

Founder and CEO, Technovation

Technovation opens up technology education and entrepreneurship opportunities to girls around the world - voices that are still missing in computing. Our programs help girls develop technical skills and an innovator's mindset while solving real problems in their communities.

Project-based learning (PBL) is central to this mission; however, we find that many teachers hesitate because complex projects require in-depth content knowledge and ongoing coaching. Generative AI can help remove these barriers. Our work has highlighted the roles Generative AI can play in supplying background knowledge, suggesting project frameworks, and giving our participants targeted feedback, so teachers can focus on classroom culture instead of mastering every domain needed for their students' varying projects.

For example, a student may be interested in developing a solution for Central Valley farmers to tackle drought-related water management in California. Previously, guiding such a project would require teachers to have expertise in climate science, data analysis, programming, and agriculture. Now, GenAI tools make it easy for any teacher to help students find relevant satellite datasets such as NASA's SMAP, and then structure a solution that adds last-mile mobile data on local water usage, well levels, crop conditions, and irrigation needs. The teacher can focus on developing students' critical thinking and collaboration skills, while the AI supports technical implementation details, allowing students to prototype, test, gather user feedback, and rapidly iterate within a timeframe that aligns with standard classroom schedules.

The Technovation model can serve as a useful starting point for educators of any background to guide students in using Generative AI tools to tackle complex real-world problems. Here is what you can do: Investigate a local environmental or social challenge in your area. Search for and combine open datasets (e.g., government records and satellite imagery) with student-collected information on the problem. Link the project to a relevant UN Sustainable Development Goal. Identify data gaps, design an AI-enabled solution to close them, and together plan for community engagement and impact tracking.

⟶ TAKE ACTION

Let students explore community issues while AI helps gather background facts and provides feedback on their problem-based learning solutions .

Education stands at a crossroads, and the path forward is clear. We have the tools to transform classrooms into accelerators for real-world innovation and changemakers. We have the opportunity to prepare students not just for jobs, but for living lives of purpose and impact, helping them not only use AI, but also build better AI.

If you need that support in taking the first PBL step, I encourage you to try Technovation's step-by-step curriculum, where students identify local problems they care about, develop AI-powered solutions, and launch real startups with global impact—empowering them to shape technology rather than be shaped by it.

Culture-First Coding: Indigenous-Led CS and AI that Honor Sovereignty

Perspective by Marie Casao[1] and Chamisa Edmo[2]

[1] Senior Program Officer at AISES (Advancing Indigenous People in STEM)
[2] Computer Scientist, AIML Curriculum Consultant for AISES, and graduate student in Electrical Engineering and Computer Science at the University of Kansas

Computer Science (CS) and Artificial Intelligence (AI) have become a key focus in AISES' work in PK-12 education in many Indigenous communities across Turtle Island. Recognizing that access to technology varies

widely across all Tribal Nations, AISES has partnered with communities to establish and expand computer science and artificial intelligence programs for Indigenous youth. As one of the first to introduce CS and hands-on training for educators and students, AISES ensures that emerging, high-demand career paths in CS and AI are accessible, relevant, and community-centered.

A major initiative in this space is Seeding Innovation, a partnership between AISES and the Kapor Foundation. It established a "culture-first" model for developing CS curricula in Native-serving high schools, focusing on promoting student engagement through culturally relevant projects and learning objectives. Over the past five years, this model has evolved into a platform for community-driven cultural revitalization projects and a means to ensure that communities play an active role in developing new technologies. This approach to curriculum development requires foundational relationship building, trust, and respect, particularly with Indigenous culture and language bearers. If this grounding in relationality cannot be established, projects are often met with resistance and are unlikely to reach their full potential.

⟶ TAKE ACTION

Co-write "culture-first" lessons with Indigenous partners. Keep control of language data inside the community.

Cultural Knowledge Experts lead project development, allowing students the opportunity to explore and critically engage CS and AI from an Indigenous community perspective and share their innovations with the broader community. Most importantly, curricula and projects are shared only with explicit permission from the community, ensuring that all knowledge is held exclusively within their community, honoring inherent Tribal Sovereignty and Self-Determination in the transmission of cultural knowledge and language. A broad example we can share, following one of our Cherokee language partner teachers' lead, is a lesson on birds. Birds are among the earliest sets of words that students in the Cherokee language class learn; therefore, we co-developed a Cherokee Birds-themed CS lesson. Language students pair up with CS students to learn and teach one another about local birds. They then collaborate to create their chosen bird, using Arduinos and servos to animate it. An updated extension incorporates machine learning to train and classify bird songs with English and Cherokee Syllabary labels, creating a resource that community members can use for bird identification with language embedded into the system.

As AI/ML tools rapidly emerge in the tech industry, many fall short in their ability to maintain secure data management and accurately represent Indigenous histories and knowledge. To address these issues, Seeding Innovation creates a space for teachers and students to critically engage with CS and emerging technologies through PD and mentorship sessions, ensuring that tools are developed and evolve in ways that respect and elevate Indigenous knowledge and perspectives. One of the concerns raised by a language and culture teacher during a discussion on AI tools is that these models are being fed or scraping Indigenous languages into the data without awareness of the intricacies of grammar or sentence structure, and as a result, the models can create new words and sentences that do not, and should not exist. It is alarming, considering they sometimes resemble real words but have been completely fabricated. If people take these models at face value, they will use the language incorrectly and inappropriately.

How can we do better for Indigenous Communities in AI? This is done through intentional relationship building, respect for cultural norms and needs, and by honoring inherent rights to intellectual property and data. The process is integrated into the curriculum, and there is no one-size-fits-all approach to computer science or

AI/ML education for Indigenous communities. Tribal Nations are proving the importance of localized, smaller-scale project development for deeper engagement and sustained learning opportunities for students and the community alike.

Conclusion

AI can bring more voices into computer science, but only if we use it with care. The thought leaders in this brief show what that looks like in practice. They demonstrate how educators play a crucial role in ensuring that AI promotes CS opportunities for all students and does not reinforce existing biases or exacerbate educational gaps. Teachers can take the lead by:

- tailoring AI-generated content to the specific needs of students;
- verifying the accuracy and appropriateness of AI-generated information that students see;
- identifying and correcting for biased AI results; and
- modeling and nurturing critical thinking, resilience, and problem solving.

Contributors' Organizations

Expanding Pathways in Computing (EPIC) is a research and service unit within the Texas Advanced Computing Center at the University of Texas, Austin. EPIC aims to broaden participation in kindergarten through graduate-level computing through research, outreach, professional development, and policy advocacy, particularly for traditionally underrepresented students.

The Carnegie Mellon University Computer Science Academy is an online, graphics-based computer science curriculum taught in Python and provided for free. It has been accessed by more than 14,000 teachers and 500,000 students.

The Stanislaus County (California) Office of Education delivers business, technology, professional development, credentialing, education, teacher recruitment, and support services to 25 school districts and other educational agencies in the county. The office provides professional development initiatives that equip educators with the tools they need to teach CS.

AISES (Advancing Indigenous People in STEM) is a national nonprofit organization focused on increasing the representation of Indigenous peoples of North America and the Pacific Islands in STEM studies and careers. AISES plans to publish its approaches, best practices, lesson examples, and findings on student and education impact from the Seeding Innovation project in 2026.

Technovation is a global tech education nonprofit that sponsors the annual Technovation Girls program, which equips girls to become tech entrepreneurs by working in teams to code mobile apps that address real-world problems. Committed to reaching girls in underserved communities, the program drew 61% participation from low or middle-income countries in 2023.

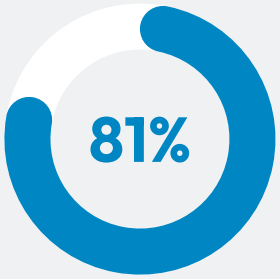
TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

What Is AI?

Artificial intelligence (AI) is a branch of computer science focused on creating programs and machines capable of performing tasks that typically require human intelligence. These tasks include understanding speech, recognizing images, and suggesting the next song.

Most AI today is machine learning (ML). An ML system "studies" mountains of data, "learns" the patterns, and then uses that training to predict, recommend, or create something new.

→ Teacher Pulse-Check



of CS teachers believe using AI and learning about AI should be in a foundational CS learning experience. (CSTA & Kapor Center, 2025)

A Brief History of AI: From Rules to Generators

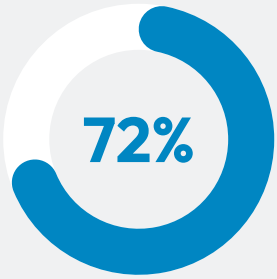
Decade	Milestone	Classroom Connection
1950s- 70s	Researchers coin the term Artificial Intelligence and write rule-based programs for chess and logic puzzles.	Introduce algorithms vs heuristics.
1980s	ML breakthroughs emerge. Rather than hard-coded rules, data drives systems' actions.	Demonstrate how a spam filter works and why it improves with more emails.
1990s- 2000s	As the web exploded, there was even more data for computers to be trained on. Faster chips (GPUs) also sped up training time from months to hours.	Explain how more data plus more computing power opens the door to bigger ideas.
2010s	Deep learning uses multi-layer neural networks and huge datasets to master vision, speech, and translation.	Test a free image-classifier and discuss true vs. false positives.
2020s	Generative AI (GenAI)—large language and diffusion models—creates fluent text, art, and code.	Compare a student-written work to an AI-generated version and critique both.

How Generative AI Works – and Why It Needs Oversight

Generative AI (GenAI) dominates today's tech headlines, so naturally, students bring questions (and sometimes the tools) into class. It's important to separate hype from reality and set ground rules.

Traditional programs follow fixed instructions. GenAI, by contrast, processes massive datasets and **predicts** what should come next, allowing it to draft text, create images, compose music, or write code that **seems** original. Because the model imitates patterns, it can often "invent" facts or even embed hidden bias or "hallucinations." That's why every GenAI output needs a human reviewer to check accuracy, context, and fairness before it is trusted or shared.

⟶ Students Want AI Guidance



of secondary school students say that guidance on how to use GenAI responsibly would be helpful. (CDT, Sep 2023)

⟶ AI in Education Beyond the CS Classroom: Sample Considerations Schools are Navigating

AI integration in schools extends far beyond CS classrooms, creating complex challenges across all subjects and operations. Our collection of briefs explore:

- **Language Support:** Instant translation tools for multilingual learners require checks for cultural sensitivity and accuracy.
- **Learning Assistance:** AI-powered hints and automated feedback systems need safeguards against over-reliance.
- **Privacy & Data**: Student work potentially training third-party AI models raises systemwide policy questions.

These complex, systemwide AI integration challenges complement the foundational AI literacy being built in CS classrooms.

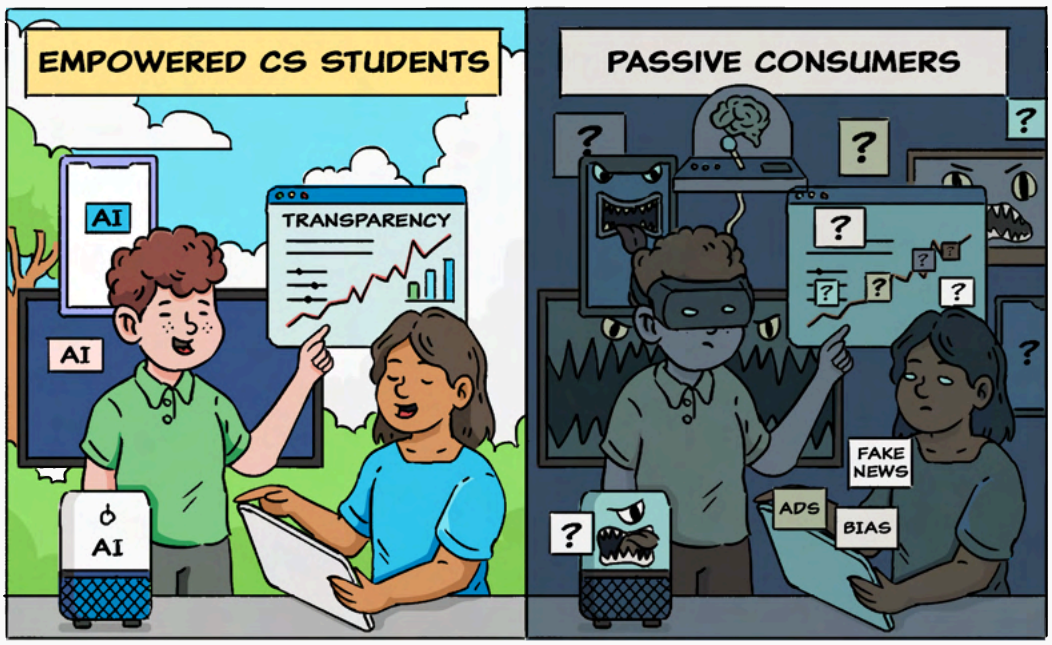
CS is More Important than Ever in an Age of AI

Despite the growing capabilities of genAI tools to write code, programming education develops the problem-solving and computational thinking skills, or "code sense," needed to use AI tools critically and responsibly. Students who can trace logic, test edge cases, and debug errors are equipped to:

- Evaluate and debug AI output;
- Identify and mitigate bias by writing tests that reveal unfair or problematic output; and
- Direct AI tools purposefully by crafting prompts and constraints that align AI capabilities with human goals.

AI does not replace the need for CS education; it elevates the importance of thoughtful, human-directed computing. These skills are vital for future careers across almost all industries, from healthcare to entertainment, where data drives innovation and decision-making. Read our briefs on the Future of CS Education at teachai.org/cs.

Computer Science Education is the Foundation for AI Literacy



⟶ The Importance of Data Science

Good models operate on good data. Integrating data science into CS classrooms can show students why missing values break predictions, how sampling choices shape fairness, and where personal privacy fits into the workflow. This foundation allows them to critically evaluate AI technologies and their impact, fostering a deeper understanding of AI-driven decision-making (Data Science 4 Everyone).

Want Deeper Dives into AI?

Learn more about AI.

- AI4K12's Five Big Ideas in AI resources and K-12 AI Guidelines.
- CSTA and AI4K12's AI Learning Priorities for All K-12 Students.
- AI in Education Presentation. Download the slides to customize your own presentation.
- AI 101 for Teachers from Code.org, ETS, ISTE, and Khan Academy.
- ISTE's AI resources, including Tips for School Leaders.

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

What AI Experiences are Essential for Primary School Students?

Integrating AI into primary school classrooms offers an engaging way for teachers to spark curiosity and foster responsible digital citizenship from a young age.

Why Start Early?

- **Promote Safe and Ethical Use**: Teachers can model and guide students on responsible use, addressing privacy and fairness early.
- **Correct Misconceptions or Misuse**: Students already interact with AI through home and school devices. Experiences led by informed teachers can clarify AI's capabilities and limitations and reduce the anthropomorphism of AI.
- **Develop Critical Thinking**: Introducing core AI concepts early prepares students to develop skills in computational thinking, data literacy, and ethical engagements with AI.
- **Broaden Participation**: Early, hands-on AI activities let every student explore and succeed, opening the path to a more diverse computing community.

Essential AI Experiences for Primary School Students

Experience	Goals	Activities
RECOGNIZING AI'S INFLUENCE	Explain what AI is and what it isn't using everyday examples.	Use workshops like Day of AI's AI for Early Elementary or ReadyAI's AI + Me to introduce AI concepts in simple terms.
EXAMINING AI'S SOCIETAL IMPACTS AND ETHICAL CONSIDERATIONS	Help students understand that AI can affect fairness, privacy, and social values.	Engage students in self-expression by having them build a digital artifact that presents their view on AI using Code.org's Share Your Voice.

Experience	Goals	Activities
INVESTIGATING HOW MACHINES LEARN	Teach students how AI systems rely on data—collecting, analyzing, and interpreting information.	Complete Code.org's How AI Makes Decisions with students to unravel processes AI systems use to analyze data and make decisions.
LEVERAGING AI FOR CREATIVE PROBLEM SOLVING	Help students see how AI drawing tools learn from human sketches and shape the stories we tell with pictures.	Explore different image creation tools, use games to investigate how AI learns from past art, and have students use AI to generate an image with How Has Technology Changed The Way We Tell Stories With Pictures?

AI Literacy for Primary School Students

Teachers can introduce key AI literacy competencies from an early age. This draft AI Literacy Framework from the European Commission, OECD, Code.org, and international experts, includes competences like:

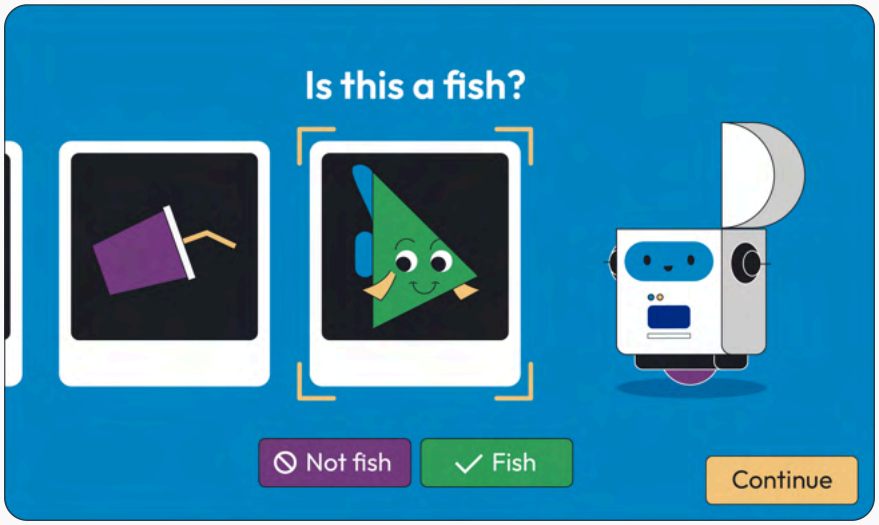
- Recognize AI's role and influence in different contexts.
- Explain how AI systems perform tasks using precise language that avoids anthropomorphism.
- Explain how AI could be used to amplify societal biases.
- Collect and curate data that could be used to train an AI model by considering relevance, representation, and potential impact.

Computational Thinking and AI

In primary classrooms, where CS enriches subjects like science and math, AI can be a context for sparking exploratory, real-world problem-solving. For example, a character sorting activity can provide an opportunity to develop core CT skills in a way that feels familiar. While organizing story characters, students practice:

- **Decomposition:** Breaking down key character traits or roles to decide how to sort them.
- **Pattern Recognition:** Identifying similarities and differences to form logical groups. This is how AI "learns" to make predictions, like recognizing a cat.
- **Abstraction:** Focusing on only important details to make sorting decisions.
- **Algorithmic Thinking:** Creating clear rules for sorting that others can follow or replicate.

As students explain their reasoning and question the fairness of groupings, they also engage in reflection—a skill critical to both CT and ethical AI literacy.



In Code.org's AI for Oceans activity, students learn how AI and machine learning can be used to address world problems like ocean pollution.

⟶ TAKE ACTION

1. Try out a free, online PD such as Elements of AI to continue learning more about AI.
2. Combine digital and unplugged AI lessons that align with core CS concepts.
3. Encourage ethical questions about technology.
4. Use a short, interactive activity from the tables in this mini-brief!

Explore available resources and curricula from initiatives like CSTA's Identifying AI Priorities for All K-12 students, Teaching AI Starting In (Pre)K-2, Day of AI, AI Club Book Series, AI4K12, Micro:bit CreateAI, Code.org's Primary School AI Curricula, and those linked above.

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

The Role of AI in Computer Science Education
⟶ Results from a Teacher Survey

Key Takeaways

1 CS teachers want to teach with and about AI.

Most teachers (85%) believe using and learning about AI should be included in foundational CS experiences, and (80%) agree that CS standards should include AI. Additionally, 87% say students should learn about careers impacted by AI.

2 CS teachers want support in teaching with and about AI.

A notable 88% of teachers expressed the need for professional learning and additional resources to effectively teach with and about AI, highlighting the necessity for updated professional development programs.

3 CS teachers are mostly optimistic or neutral about the potential benefits and drawbacks of AI in CS education.

While 52% believe the benefits of AI outweigh the risks, 43% see an equal mix, and only 5% said the risks outweigh the benefits. Key concerns include overreliance on AI, the potential for plagiarism, and ethical issues such as bias in AI systems.

4 CS teachers stress the importance of teaching programming skills alongside AI education but disagree on when students should start using AI tools.

Teachers believe programming is essential for understanding, debugging, and effectively collaborating with AI and crucial for developing computational thinking, problem-solving, and critical thinking skills.

While 55% of CS teachers believe students in introductory CS classes should learn to program with AI, 24% are unsure, and 21% disagree.

5 CS teachers cite bias, misinformation, overreliance, and the ethical implications of Gen AI technologies as common concerns.

Across the survey, teachers identified the need to talk with students about AI's ethical and societal impacts. They expressed the desire for guidance on discussing these issues and how AI tools will specifically impact the classroom.

CSTA and TeachAI conducted a survey in May 2024 about computer science teachers' usage of, beliefs about, and attitudes toward AI.

- 364 responses from current CS teachers
- 24% primary school, 76% secondary school
- 12% outside of the US
- 44% urban, 43% suburban, 13% rural

CSTA gathered additional data on these questions in a much larger and more representative landscape survey of CS teachers in the U.S. in the fall of 2024. Overall, the trends from this survey were validated. A summary can be found in the education chapter of the 2025 AI Index Report. Full results will be available at landscape.csteachers.org.

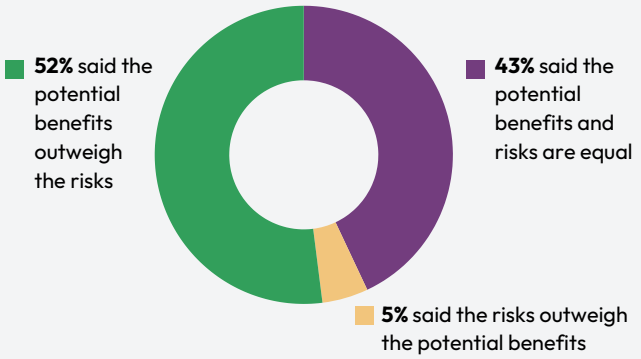
AI Curriculum Content

Why teach with and about AI?

Teachers emphasized that understanding AI is essential for students' future personal and professional lives in the digital age. 76% agree that AI can be used to promote creativity in the CS classroom.

Teachers were mostly optimistic or balanced about the benefits and drawbacks of using AI in the classroom.

⟶ Thinking about ways AI tools could be used in CS education...



What should we teach about AI?

There is a strong consensus among CS teachers about the importance of AI education:

- 87% agree that students should learn about careers impacted by AI.
- 79% believe that CS standards should be updated to include discussions about AI.
- 85% think students in introductory courses should learn about AI.

How should CS curriculum evolve?

Teachers indicated that students should continue learning fundamental programming skills, including topics like variables, loops, conditionals and topics that directly apply to AI, like data structures and algorithms. Many strongly advocated for students to learn about the societal impacts of AI in their foundational CS courses. Many also responded that they were unsure which CS topics should be emphasized or changed.

> "There is no getting around the need for humans to master order of execution, variables, logic, conditionals, looping, functions, modular design, and so on."
>
> - **Secondary CS Teacher**, Maine, USA

"I don't think any topics should be removed. But I think that real world applications and the future need to be emphasized more... I believe that AI needs to be included into the standards as soon as possible."

- **Secondary CS Teacher**, Pennsylvania, USA

"Collaboration should always be emphasized. Struggling with a problem to find a solution should be emphasized. Understanding that there are multiple ways to solve a problem should be emphasized."

- **Secondary CS Teacher**, Maryland, USA

"I think topics should include more about the mechanics of computer science, the changing job landscape, daily life intersection with computer science, and AI fundamentals for students."

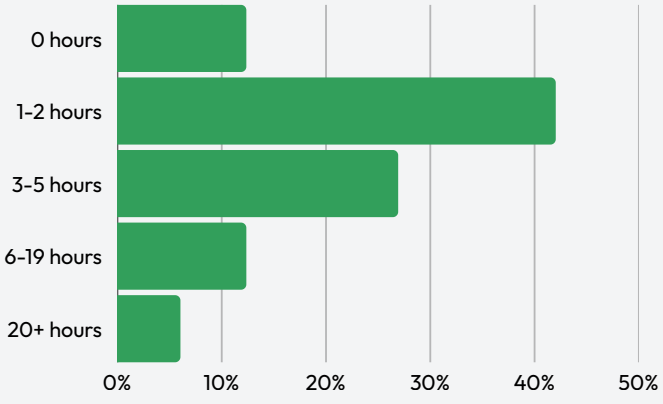
- **Primary and Secondary CS Teacher**, Ohio, USA

Teaching Methods and Support

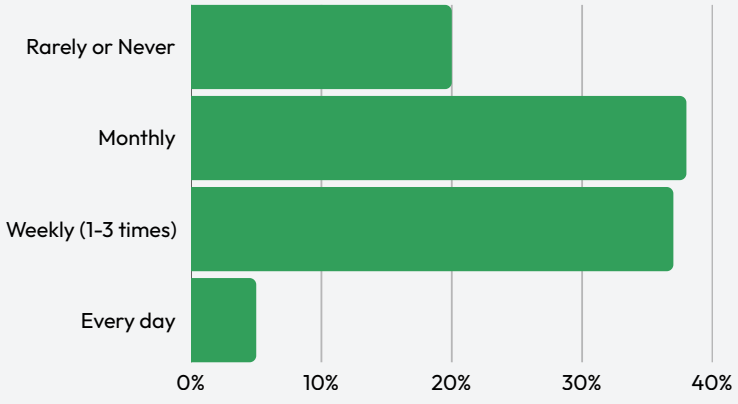
How often are teachers teaching about AI?

CS teachers are teaching students how AI works and discussing AI's ethical implications with students.

⟶ How much do students currently learn about AI in your computer science curriculum?



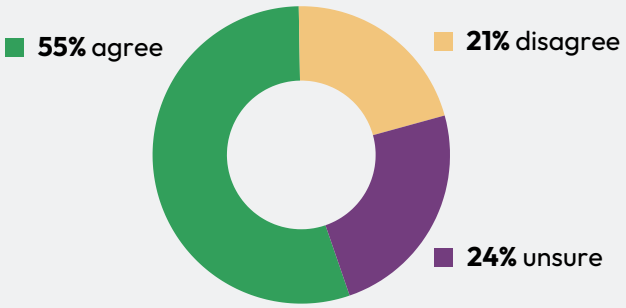
⟶ How often do you discuss ethical questions related to AI, such as algorithmic bias, digital citizenship, and privacy?



When should AI be used in the classroom?

There is a division among teachers on when to introduce AI tools.

⟶ Students in introductory CS classes should learn to program with AI.



Interestingly, 54% of secondary school teachers agreed that students should learn to use AI in introductory programming courses, versus 62% of primary school teachers.

What AI topics should be included in CS teacher professional development?

Teachers want to integrate balanced discussions about the benefits and risks of AI with hands-on projects, but they require additional support to teach with and about AI effectively.

- 48% feel currently equipped to teach AI.
- 88% need more resources for professional learning.
- 4% disagree that they need additional support.

The most commonly requested topics for professional development centered on how AI works, how to use AI, and the ethical impacts of AI tools. While these topics are not exclusive to the CS classroom, teachers identified their relevance to CS learning and their importance to developing digitally literate students.

"I want to be able to spend my time gathering new resources to use in my classroom and less time grading papers and creating worksheets and exams."

- **Primary CS Teacher**, New Jersey, USA

"For professional development tailored to computer science teachers, focusing on AI topics like machine learning fundamentals, ethical considerations in AI development, integrating AI concepts into curriculum, and practical coding applications would be crucial."

- **Secondary CS Teacher**, Bauchi, Nigeria

"I would love to see more PD for primary level CS teachers with ways to ready these young learners without necessarily using Gen AI. Also, teachers need PD more readily available for themselves. Our district thus far has provided none, so unless a teacher is highly motivated to seek understanding themselves, they are not getting it."

- **Primary and Secondary CS Teacher**, Arizona, USA

Student Learning and Development

Why should students still learn to program?

Teachers most commonly responded that given Gen AI coding tools, students should still learn to program to be able to develop new Gen AI applications, understand how AI works, and ground their work in fundamental computer science knowledge.

> "Learning computer science is still crucial for students, even with the rise of AI. Understanding the principles behind coding and computational thinking not only enables students to create AI but also helps them grasp its limitations, ethical considerations, and potential societal impacts."
>
> - **Secondary CS Teacher**, New York, USA

Teachers emphasized that developing a robust "code sense" through learning to program is just as crucial with AI as without. Many teachers noted that AI does not generate perfect code, and students must have a solid foundation in programming to analyze, debug, and use what AI creates.

> "As with many parts of education, students need to learn the mindset and logic of programming. This allows them to fix mistakes or optimize poorly written code, whether that is from a classmate or an AI."
>
> - **Secondary CS Teacher,** Kentucky, USA

Teachers also championed the "programmer's mindset" and shared about the value of computational thinking, problem-solving, and creative skills developed in the CS classroom.

> "AI is no substitute for human ingenuity. Students still need to learn problem-solving, logic, teamwork, communication, etc., which is the heart of computer science."
>
> - **Secondary CS Teacher**, Massachusetts, USA

> "Coding teaches students to think computationally, problem-solve, think through steps, cause and effect, debugging (finding and fixing their mistakes), and critical thinking skills. These skills are not caught; they are taught and learned."
>
> - **Primary CS Teacher**, New Jersey, USA

What are the potential benefits of using AI in the CS classroom?

CS teachers most commonly identified the potential benefits of using AI in their classrooms as increasing learning opportunities, automating tasks, and preparing students for their futures in the digital world.

- They most commonly identified AI as helpful in enhancing students' learning opportunities by strengthening engagement, broadening access, and shaping their content knowledge.
- They described AI's ability to differentiate instructional materials, including translation support for multilingual learners.
- They also noted that AI's ability to provide personalized feedback and additional examples leads to a deeper understanding of information.

> "It can be used to model or explain code and concepts to scaffold a student's learning. It can create additional examples for students to explain, create faulty code for 'find the errors,' recommend sources for research, create sample questions on a topic to help test-prep."
>
> - **Secondary CS Teacher**, New York, USA

CS teachers praised AI's ability to automate tasks, such as designing assessments and lessons and providing feedback on student work. With support from AI in their learning management systems and everyday classroom procedures, teachers can focus more on the students in front of them.

> "AI can free up your time for more one-on-one instruction by grading repetitive tasks and providing instant feedback on student code."
>
> - **Secondary CS Teacher**, California, USA

By introducing students to AI early — how it works and how to use it — teachers prepare students for their future personal and professional lives in the digital age.

> "[We are] preparing students for the future where AI is a tool they can use and work with, not against or replaced by."
>
> - **Secondary CS Teacher**, Maine, USA

Ethical and Practical Considerations

What are the potential risks of using AI in the CS classroom?

The most common concerns were about overreliance on AI, plagiarism, and biases or ethical concerns. CS teachers were particularly concerned that students would use AI instead of learning the content, losing both programming skills and the collaborative experience.

> "Students will not have to 'struggle' with problems. They can use AI to help debug code or outline a program. Students will use AI for that collaborative problem-solving rather than engaging with peers or instructors. Communication skills are really important for students to develop."
>
> - **Secondary CS Teacher**, Montana, USA

CS teachers also described that, without such understanding, students might be more inclined to use AI to cheat on their assignments.

> "Without supervision or intentional usage, it could be misused to create code that students claim as their own."
>
> - **Secondary CS Teacher**, Kentucky, USA

Teachers also expressed concerns that biases in AI algorithms might impact the classroom community.

> "AI systems have proved to be biased and to produce incorrect coding solutions at times. It could lead to incorrect or biased learning for the students, as well as discriminated grading if an AI tool is used for grading."
>
> - **Secondary CS Teacher**, Arizona, USA

Finally, teachers discussed the importance of student privacy and protecting sensitive information. While they agreed that AI might benefit students by providing personalized support, they emphasized the importance of caution and clearly understanding what data is being collected or used and how.

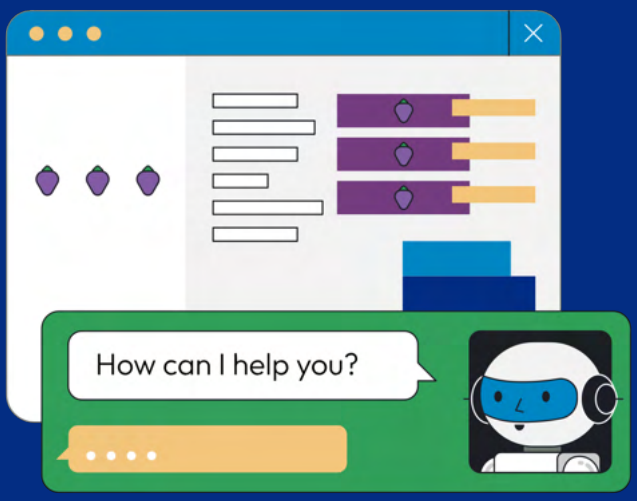
TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

About TeachAI

TeachAI brings together education leaders and technology experts to assist governments and education authorities in transforming education through teaching with and about AI. The initiative is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum, and advised by a diverse group of 170+ organizations and governments. TeachAI's goals include guiding policy, increasing awareness, and building community and capacity.

About CSTA

The Computer Science Teachers Association (CSTA) is a global membership organization that unites, supports, and empowers educators to enhance the quality, accessibility, and inclusivity of computer science education. With the vision that every student should be prepared for a world powered by computing, CSTA advocates for K-12 computer science teachers worldwide, providing professional development, curriculum resources, and a community of practice dedicated to advancing CS education for all students.



Visit teachai.org/cs

TeachAI is led by Code.org, ETS, the International Society for Technology in Education, Khan Academy, and the World Economic Forum.

TeachAI & CSTA (2025). Guidance on the Future of Computer Science Education in an Age of AI. Retrieved from teachai.org/cs.