

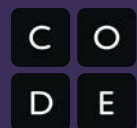


# Guide to Inclusive Computer Science Education

How educators can engage all students in computer science



Information in this guide was developed in collaboration with the following partners



© Microsoft corporation

This document and resources are for your internal use only. Microsoft makes no representations or warranties express or implied in this document or its content.

The Guide to Inclusive Computer Science Education includes links to other Microsoft and third-party sites. The Terms of Use and Privacy Statement on those sites control your use of their content.

# Contents

## Introduction



Why computer science is important

[Page 3](#)

How to use this guide

[Page 5](#)

Inclusive terms

[Page 6](#)

## Inclusion for administrators & support staff



Access to computer science education

[Page 7](#)

Diversity in computer science classrooms

[Page 8](#)

Resources for administrators and support staff

[Page 10](#)

## Educators and inclusion in the CS classroom



Making classroom spaces more inclusive

[Page 11](#)

Curricular materials

[Page 13](#)

Instruction and pedagogy

[Page 15](#)

Accessibility in the computer science classroom

[Page 17](#)

Resources for educators

[Page 19](#)

# Introduction

## Why computer science is important

Computing and technology hold the promise of solving problems big and small and shaping the world around us. But if the people who work in computing-related jobs don't represent our communities and our populations, the solutions they develop won't be representative either. Technologies will emerge with unintentional bias and limited insight into the diversity of people who will use and depend on them.

Knowledge of computers is fundamental to students' future careers. The number of new jobs in STEM (science, technology, engineering, and math) is increasing, especially those in computing. At the same time, we're entering a technological era with the potential to power awesome advancements across every sector of our economy and society. This places us at a historic intersection of opportunity and responsibility to the world around us. Our actions must help solve the world's problems, not create new ones.

By 2025, 149 million jobs will be added to the global technology sector.<sup>1</sup> This is why computer science (CS) classes need to be inclusive. Teachers, administrators, and school support staff can lead the way in showing students at even the youngest levels that they too can embrace the art and science of computers. And then they can be empowered to create a better future.

But we know that there are students historically excluded from learning CS because of race, gender, or geography. Together, these groups represent 65 percent of the entire U.S. population. As of 2022, only 32 percent of high school students enrolled in CS science courses are female<sup>2</sup>. In addition, students from historically excluded racial and ethnic groups are underrepresented in taking AP computer science exams even when they attend a school that offers it.<sup>2</sup>

Further inequities limit access to CS education for students with disabilities and students in rural communities. To prepare young people for jobs in computing and technology, we need to help them see themselves as builders, creators, problem-solvers, and computer scientists. Developing equitable learning environments requires educators to provide a clear and welcoming pathway of CS educational coursework for students.

Educators have the opportunity to teach and shape CS learning spaces and curricula in ways that let all students know their different backgrounds, perspectives, and abilities have value in the work they do and the ideas they bring to life. This includes exposing students to a variety of role models, ensuring the classroom's physical space is comfortable and accommodating, and providing culturally relevant lessons and activities that stimulate different types of students.

**Computing is a fundamental part of daily life, commerce, and just about every occupation in our modern economy. It is essential that students are exposed to the field of computer science in our K-12 system as it is foundational in transforming the way a student thinks about the world. It not only teaches them about technology, but it also teaches them how to think differently about any problem.**

*[Nine Policy Ideas to Make Computer Science Fundamental to K-12 Education – Code.org](#)*



## Who is this for?

This guide provides educators with context and concrete steps to build and expand inclusivity in CS education. By actively engaging students in CS, educators can build an even stronger pipeline of creativity and innovation to tackle the world's challenges and help ensure students have the skills needed to thrive today and tomorrow. While the guide can be helpful for all educators, it is primarily focused on the U.S. education system.

## What is the goal of this guide?

This guide provides a solid foundation to help bring inclusive practices into the CS classroom. This includes thinking through how to approach subjects such as access, diversity, and accessibility for CS learners. The material here will serve as an introduction to principles and concepts required to recruit diverse learners, create inclusive learning environments, and a helpful resource for those in the early phases of approaching inclusion in the CS classroom.

## Microsoft Philanthropies

We're investing our greatest assets – our technology, people, grants, and voice – to advance a more equitable world where the benefits of technology are accessible to everyone. Technology should be an equalizing force in the world, not one that drives people further apart. We believe that companies that can do more, should. That's why we remain focused on four key areas in which technology can and must benefit the future of humanity and our planet. Those pillars are: Support inclusive economic growth, Protect fundamental rights, Create a sustainable future, and Earn Trust. Learn more at [Microsoft.com/en-us/corporate-responsibility](https://Microsoft.com/en-us/corporate-responsibility).

## About Microsoft Philanthropies TEALS Program

Microsoft Philanthropies TEALS (Technology Education and Learning Support) Program puts technology professionals from across the technology industry into high school classrooms to team-teach computer science with classroom teachers. Through TEALS, technology professionals share their knowledge with teachers while students benefit from learning how computer science is used in the workplace. TEALS operates in 8 countries, including 30 U.S. states and Washington, DC. Since its inception in 2009, over 100,000 students have participated in a TEALS class. Learn more at [Microsoft.com/teals](https://Microsoft.com/teals).

“ If we are ensuring that there are diverse teams and diverse folks at the table at every step of the pipeline, it creates the opportunity to have tech look like the world that it represents, which benefits us in a million different ways. ”

– Dr. JeffriAnne Wilder, NCWIT



1. Learning can spark confidence amidst unexpected changes. Microsoft Learn Blog, 2021. <https://techcommunity.microsoft.com/t5/microsoft-learn-blog/learning-can-spark-confidence-amidst-unexpected-changes/ba-p/2038069>.
2. Code.org, CSTA, & ECEP Alliance (2022). *2022 State of Computer Science Education: Understanding Our National Imperative*. Retrieved from: <https://advocacy.code.org/stateofcs>.

# How to use this guide

This guide is divided into sections, each containing pages for specific subject areas of inclusion. It is not necessary to read the guide from beginning to end. Each section is targeted to specific roles within the school. Below is some information that can help you navigate the guide and implement the resources provided.

1

**Each section targets specific roles within schools. Below is how this guide uses these terms.**

### **Educators**

- Teachers
- Teaching assistants
- Special education teachers

### **Administrators**

- Principals
- Assistant Principals
- Career and Technical Education coordinators

### **Support Staff**

- Counselors
- Librarians
- Media specialists
- IT Admins

2

**The guide is populated with tool tips to support the content and reference important research. Look for these tool tip boxes for important concepts, quotes, or tasks to help you understand the material presented.**



An important concept or question to think about as you read the guide



A quote or important research cited to support the concepts in the section



A task that will help you implement the material or concepts discussed



## **Take Action**

### **Educators**

- Task description, resource link, or activity suggestion for Educators to act upon.

### **Administrators**

- Task description, resource link, or activity suggestion for Administrators to act upon.

### **Support Staff**

- Task description, resource link, or activity suggestion for Support Staff to act upon.

3

**Some pages will end with actions to help you apply the concepts described. These actions are specific to each intended audience.**

# Inclusive terms

The words we use to explain inclusion in CS are nuanced and can have varied meanings. Before starting on your inclusion journey, it is important to understand some of the terms used. For the purposes of this guide, here are some of the most commonly used terms surrounding inclusion.



**Review these terms and think about ways to better support students.**

Term	Definition
<b>Access</b>	The right and opportunity for all students to learn and experience computer science.
<b>Accessibility</b>	When a person with a disability is afforded the opportunity to acquire the same information, engage in the same interactions, and enjoy the same services as a person without a disability in an equally integrated and equally effective manner, with substantially equivalent ease of use.
<b>Belonging</b>	Belonging is the feeling of security and support when there is a sense of acceptance, inclusion, and identity for a member of a certain group.
<b>Bias/unconscious bias</b>	Stereotypes, prejudices, or preferences that cause us to favor a person, thing, or group in a way that is deemed unfair. They are implicit attitudes, behaviors, words, or actions that we exhibit in our personal lives and in the workplace.
<b>Diversity</b>	Having CS courses or programs whose student enrollment rates reflect the demographics of the larger school or community population, particularly in terms of race, ethnicity, gender, and disability status.
<b>Equity</b>	Ensuring that access, resources, and opportunities are provided for all to succeed and grow, especially for those who have been historically excluded.
<b>Culturally responsive teaching and learning</b>	Instructional practices and learning experiences that actively take into account the context of youth in terms of interests, identities, cultural and linguistic practices, and histories. Culturally responsive education maintains and sustains students' cultural integrity while supporting academic achievement.
<b>Inclusion</b>	Creating learning environments that are accessible and welcoming of students' identities, backgrounds, differences and perspectives without barriers or judgment. This means actively attending to gender, race, ethnicity, ability, or socioeconomic status.
<b>Intersectionality</b>	The ways that a person's race, gender identity and other identity factors overlap and compound to create particular forms of advantage or disadvantage. This term is a way to describe how different marginalized groups experience equity, inclusion, and power differently.
<b>Universal design</b>	There are two applications of this term. In physical space, it means supporting accessibility for all types of people. This may include wheelchair-accessible desks, collaborative spaces, voice recognition software, etc. In curriculum, it refers to instruction that accounts for students' varied interests, abilities, and prior knowledge.
<b>Targeted Universalism</b>	Setting universal goals pursued by targeted processes to achieve those goals. The strategies developed to achieve those goals are targeted, based upon how different groups are situated within structures, culture, and across geographies to obtain the universal goal.

# Inclusion for administrators & support staff

Strategies for schools to address foundational issues surrounding inclusion in CS classrooms.

## Access to CS education

### Where to start

The first step to addressing access for any school is to offer CS! Evaluating students' learning opportunities for CS starts with understanding how each student can participate in CS learning.

### Offer CS learning opportunities

- Work to make CS a required course at your school so that there are enough course sections for all students. The [Code.org 2022 State of Computer Science Education](#) report shows that this has helped states reduce disparities and increase graduation.
- If CS courses exist at your school but there is limited availability for CS course enrollment, make sure that the student selection process is not based on assumptions about which students are interested or will do well in CS. Provide teachers, counselors, and parents with information about why it is important for all students to take computer science. The [CS is for Everyone Student Recruitment toolkit](#) offers helpful ready-to-use resources.
- Consider models like [Microsoft Philanthropies TEALS Program](#) (Technology Education and Learning Support) which partners high school teachers with volunteers from industry so they can team teach CS.

### Ensure CS classes are open to all abilities

Proactively look for barriers that could be keeping students from accessing CS classes. If students are visually impaired, for example, are there tools that will support them in the classroom? [AccessCSforAll](#), a program led by the University of Washington and University of Nevada, Las Vegas, provides resources, tools, and materials that educators can leverage to make CS instruction inclusive for students of all abilities.

Look at when CS lessons are offered at your school. Are there times when some students need to leave the classroom for other activities, such as to receive special education services? If a child is being pulled out of the classroom for other interventions during CS instruction, ensure that educators call out these conflicts.



## Take Action

### Administrators

- If your school offers CS, share [The CS is for Everyone Student Recruitment toolkit](#) with educators and support staff.
- If your school offers CS, work with support staff to help identify barriers such as scheduling issues that might prevent students from accessing CS classes.
- Plan a session for how K-12 CS stacks together for learning sequences; consider attending a local [SCRIPT workshop](#) to create a CS planning team to determine how and why CS is offered in your school.

## Diversity in computer science classrooms

As you'll see below, increasing diversity is a team effort, with teachers, counselors and even other students helping to spread the word and setting an example.

### Build support for CS throughout your school

- Enlist the support of your school ecosystem – administrators, teachers, guidance counselors, families, and students – to enhance communication and understanding around CS opportunities. Microsoft Philanthropies provides two free resources – the [Girls in STEM Action Guide](#) and [Computer Science Is for Everyone Recruitment Toolkit](#) – that will help all of you be inclusive ambassadors for CS.
- See the [Computer Science Professional Development Guide](#) for tips on building capacity to support computer science education.
- Teachers can earn a badge when they take the [Closing the STEM Gap](#) on Microsoft Learn to learn how to help close the gender gap in computer science.

### Understand which students are under-enrolled in AP STEM courses

- Building professional development for educators can help in proactively identifying Black and Latino students using relevant instructional practices to prevent gaps in enrollment for those who have been historically excluded.
- School leaders and support staff can address these issues with surveys to better understand where student interest diverges from school enrollment for those who have been historically excluded.



“ When school counselors and other adult influencers understand and address biased perceptions of who “belongs” in CS and IT, a more diverse range of students are encouraged to explore computing and are empowered to consider careers in tech. ”

– Dr. June Teisan, NCWIT

### Include guidance counselors in your efforts

- School counselors can be excellent champions for CS courses. But students can sometimes be unconsciously filtered out by counselors before having a chance to try CS. Work with your counselors to help them understand what CS is about and who's a good fit for the courses (hint: Everyone!).
- [NCWIT Counselors for Computing](#) offers valuable resources and information for counselors to support all students in exploring CS, including a [Top 10 Guide](#) for how teachers can engage counselors as allies.

### Enlist students to promote CS

- Ask current CS students to promote CS education and share with prospective students what they might learn and create. Focus their “peer presentation” energy around course enrollment time, and coach them to be inclusive and sensitive to all kinds of students when reaching out.
- Generate excitement with a steady narrative about how CS careers are creative and critical to solving real-world problems students care about. Put up posters and displays from [Hour of Code](#) or distribute student-handouts from [NCWIT](#) that highlight the creative possibilities in CS.



## Introduce students to role models from diverse backgrounds in CS

- Role models matter for CS inclusion. Think about inviting in a diverse group of educators, guest speakers and other role models who can connect with students in different ways. Teachers and other adults sharing their backgrounds and personal stories can be very valuable in helping students make links with computing.
- Use days and dates of observance (Pi Day, Earth Day, Women’s History Month, CS Education Week, etc.) as opportunities to discuss careers related to CS.
- Consider a co-teaching arrangement to offer different teacher perspectives and better help different types of students relate to or engage with their instructors. For example, a CS teacher can partner with a special education teacher, a bilingual teacher, or an instructional coach. The project [Teaching All Computational Thinking through Inclusion and Collaboration \(TACTIC\)](#) offers tips for effective CS co-teaching.



“ When computer science courses are required, then CS classes represent the whole student body. ”

– Dr. Leigh Ann DeLyser, CSforALL



## Take Action

### Administrators

- Download and print the [Computer Science Professional Development Guide](#) to find steps to support educators.
- Read the [Tactical Teaching Brief](#) to identify ways co-teaching support for CS classes can be supported in your school.

### Support Staff

- Download one of the resources at [NCWIT Counselors for Computing \(C4C\)](#) like the [C4C Information sheet](#) with information to advise students about careers in computing.

## Resources for administrators and support staff

Leverage these resources as your school implements computer science classes and works to implement inclusion best practices.

### Access to computer science education

---



#### [Computer Science Professional Development Guide](#)

How education leaders can build teacher, school counselor, and administrator capacity to support equitable computer science education.

#### [Computer Science is for Everyone Student Recruitment](#)

A toolkit for middle and high schools to increase diversity in computer science education.

#### [AccessCSforALL resources for a range of student abilities](#)

Accessible curricula and tools for K-12 computing education.

### Diversity in computer science classrooms

---



#### [Girls in STEM Action Guide](#)

A guide for teachers, parents, and education and nonprofit leaders to help close the gender gap in STEM and CS.

#### [NCWIT Counselors for Computing resources](#)

The NCWIT Counselors for Computing (C4C) program provides information and resources that help counselors join the front line of the computing conversation.

#### [NCWIT Top 10 Ways to Engage School Counselors as Allies](#)

Consider these key points for collaboration as you meet with counselors to discuss how their professional responsibilities align with goals to increase student access to CS.

#### [Project TACTIC: Teaching All Computational Thinking through Inclusion and Collaboration](#)

The project TACTIC is the NSF STEM+C project that focuses on developing inclusive computer science experiences for students with disabilities and those at risk for academic failure in elementary and middle school settings.

#### [Microsoft Computer Science Education Resource Guide](#)

75+ Microsoft supported CS education initiatives that assist in the overall development of computer science programs in schools.

# Educators and inclusion in the CS classroom

This section describes how educators can infuse learning best practices and leverage resources to be more inclusive in the computer science classroom.

## Making classroom spaces more inclusive

### **Incorporate visuals that will appeal to a wide range of student interests and backgrounds.**

- In order for us to feel like we belong somewhere, it helps us to see depictions of people we relate to. Hang posters showing a diverse range of people engaging in CS, such as these [Code.org computer science posters](#).
- Make sure that the visual displays are relevant in terms of gender, race, age, as well as time period. Technology evolves so rapidly that the way we pictured CS a few years ago might seem dated and irrelevant to students today. Current images, grounded in the world that students know, will help them feel welcome.
- Don't forget examples of CS students and professionals with disabilities. The Alliance for Access to Computing Careers has a great [list of real-life profiles](#) you can draw from.

### **Feature examples of real-world applications of CS in your classroom**

- To help students connect CS to the other academic areas they're drawn to, experiment with lessons from the [Code.org Education Incubator](#) to find new ways of connecting learners to computing.
- The [NCWIT AspireIT Toolkit](#) helps anyone interested in designing engaging and inclusive computer science experiences for K-12 students, with a focus on broadening the participation of girls, women, and those who have been historically excluded.

### **Display student projects and contributions**

- Many CS projects have tangible outputs – like a visual design, electronic textiles, or even something robotic. Putting students' work on display helps sustain interest and pride among current students. It also provides further proof to visitors and newcomers that CS is creative and dynamic.

### **Arrange the learning space to promote collaboration and hands-on activities**

- The classic "desks in rows" setup can imply that CS is a rote subject that everyone should learn individually through direct instruction from the teacher. If possible, design the room so learning can be guided rather than merely broadcast, and so students can share and try things out together rather than simply listening along.

### **Get to know your students and their needs**

- Learn your students' names and build connections to create an inviting environment.
- Find ways to address the needs of students by letting them know you welcome time with them to discuss disability-related accommodations or learning needs.
- Provide information about resources such as tutoring, writing spaces, and disability services that are available to all students.
- Ensure that you know how to arrange accommodations in the classroom such as alternate formats of material, captioning in videos, or other resources.

## Design learning spaces that are accessible to students with diverse abilities

- Arrange aisles and workstations so that students with wheelchairs or other physical mobility aids can get to all the areas they need to access in order to participate fully.

## Make sure that technology resources support the needs of students

- Be sure that technology requirements or instructional tools you choose to use in the curriculum are consistently available and accessible for students. If curricular materials require regular or daily use of the internet or technology that is not consistently reliable, you might want to select resources available offline.



**Tip: Try asking the learners for some current technologies they are interested in!**

## Make sure that technology resources support the needs of students

- Be sure that technology requirements or instructional tools you choose to use in the curriculum are consistently available and accessible for students. If curricular materials require regular or daily use of the internet or technology that is not consistently reliable, you might want to select resources available offline.



## Take Action

### Educators

- Print out and display computer science posters from [Code.org](https://code.org) or use [NCWIT](https://ncwit.org) resources in your classroom to inspire your students to try computer science.

### Support Staff

- Print out and display computer science posters from [Code.org](https://code.org) or use [NCWIT](https://ncwit.org) resources in your classroom to inspire your students to try computer science.

## Curricular materials

The final ingredient of building an inclusive classroom is the coursework itself. Decisions around the adoption of curricula are often made by administrators, so it's important that they work together with teachers to make the best decisions for their school and students.

### Give students variety and choice in their learning experiences

- Allow students to incorporate a range of features and aesthetics into their work, while ensuring all students meet the learning objectives.
- Look for opportunities to engage students in project-based learning around CS. More extensive projects can allow opportunities for students to draw on their cultural assets and community knowledge while developing their CS knowledge and skills.
- Prioritize hands-on learning experiences, such as the [MakeCode for micro:bit Curriculum](#) to highlight the creative and problem-solving aspects of CS.



Review [Exploring Computer Science](#), [MakeCode Intro to CS](#), and [Advanced Placement CS Principles](#) to help improve classroom instruction of CS principles.



### Sequence lessons and assessments in a cohesive progression of learning

- Because CS is still quite new to many schools and teachers, curricula often consist of a series of discrete skills and activities. But just like math, history, and other subjects, CS is best taught as a cumulative subject where each lesson builds on the one that comes before it. To find progressive CS lessons, consider resources such as [Exploring Computer Science](#).
- Make sure that student performance metrics align with your curriculum – covering the same content and skills and making the same acknowledgments of students' different experiences.

### Select curricular materials that highlight diversity and inclusion in meaningful ways

- As with visual representations in the classroom, instructional material should feature diverse cultures and communities to showcase the historical and cultural integration of computing and diverse people who do computing.
- Include lessons and assignments that build on the cultural assets, prior knowledge, and interests of students to feature CS in contexts that are interesting for students. Incorporate learning materials that are accessible for students of all abilities.
- There are approximately 7.6 million students with disabilities in the U.S. To learn more about making CS accessible to them, check out the [CSforALL Accessibility Pledge](#).

## Incorporate learning materials that are accessible for students of all abilities

- Learn about creating experiences that follow universal design (UD) principles that ensure courses are welcoming to all from [The Center for Universal Design in Education](#).
- Students who can't see, can't use a keyboard, or have other physical restrictions should not be ineligible for CS. Many tools and curricula make CS accessible – one example is [Quorum](#), which offers ways to teach computer science to learners who benefit from varying modalities. You can also find information about Accessibility at [Microsoft Learn](#).
- View the video [How Can We Include Students with Disabilities in Computing Courses](#) to meet CS students and professionals with disabilities and see how accommodation, assistive technologies, and universal design approaches can make computing courses accessible for students with disabilities.
- Visit the University of Florida College of Education's [Universal Design for Learning for Computer Science \(UDL4CS\) Professional Development Hub](#) to explore resources specific to UDL in CS education as well as resources related to the inclusion of learners with disabilities.
- The University of Florida College of Education's UDL4CS Professional development hub has resources for implementing UDL in CS education.



“Curricular materials must be multi-dimensional allowing all students to see themselves and their communities represented, must be appropriately mentally challenging while sparking students' creativity as well as providing opportunities for students to critically examine computing and technology.”

– Shana V. White, Kapor Center



## Take Action

### Educators

- Leverage teaching tools and curricula to teach CS with a range of with [Quorum](#).
- Watch the video (~10min.) [How Can We Include Students with Disabilities in Computing Courses](#) to learn about approaches that can make computing courses accessible for students with disabilities.

## Instruction and pedagogy

Just as you consider the physical space of your computer lab or CS classroom, remember to think about how inclusion is reflected in your approach to instruction. Teachers without a CS background have found that CS professional development programs with a focus on equity and inclusion are important for developing the instructional skills necessary to teach CS and broaden participation in computing.

### Encourage students to focus on the problem-solving process

- In CS, there can be multiple solutions to a problem. In fact, that's why we need an increasingly diverse cadre of computer scientists in our world. When we have all different kinds of knowledge and perspectives working to solve problems, we will get all different kinds of solutions. So, focus your instruction – and encouragement – on solving problems rather than finding a single right answer.
- Emphasize guided inquiry. Design learning opportunities where students can ask questions, explore, try different approaches, and challenge their own and each other's ideas.
- Highlight that mistakes in computer science are called "bugs," a term first coined by CS pioneer Dr. Grace Hopper, and that "debugging" is a central and defining practice in the field of CS. To celebrate debugging, share a (willing) student's error with the rest of the class each day as a learning opportunity – "My favorite bug of the day! Let's figure it out together."



**Encourage students to take ownership over their own learning and reflect on their problem-solving process. Ask students about the CS strategies they are using to solve the problem at hand.**

### Support students in taking risks

- Build a community. Some students may feel out of place in your CS class right away. Just showing up might be a risk. You can put them at ease by promoting collaboration, peer-to-peer learning, and small-group work to help them build relationships and more comfortably share ideas.
- Encourage exploration and creativity by resisting the urge to give students answers and solutions. Support a "growth mindset" – let students struggle and reward ideas and effort rather than a final product.
- Incorporate journal writing as an activity that allows students to gather thoughts and ideas before sharing out to peer-partners, small groups, or a larger class discussion. This gives time for all students to think about their own ideas around a particular topic before listening to others speak in a larger discussion.
- Create opportunities to encourage students to learn outside the classroom, such as through internships, community college courses, and summer programs.

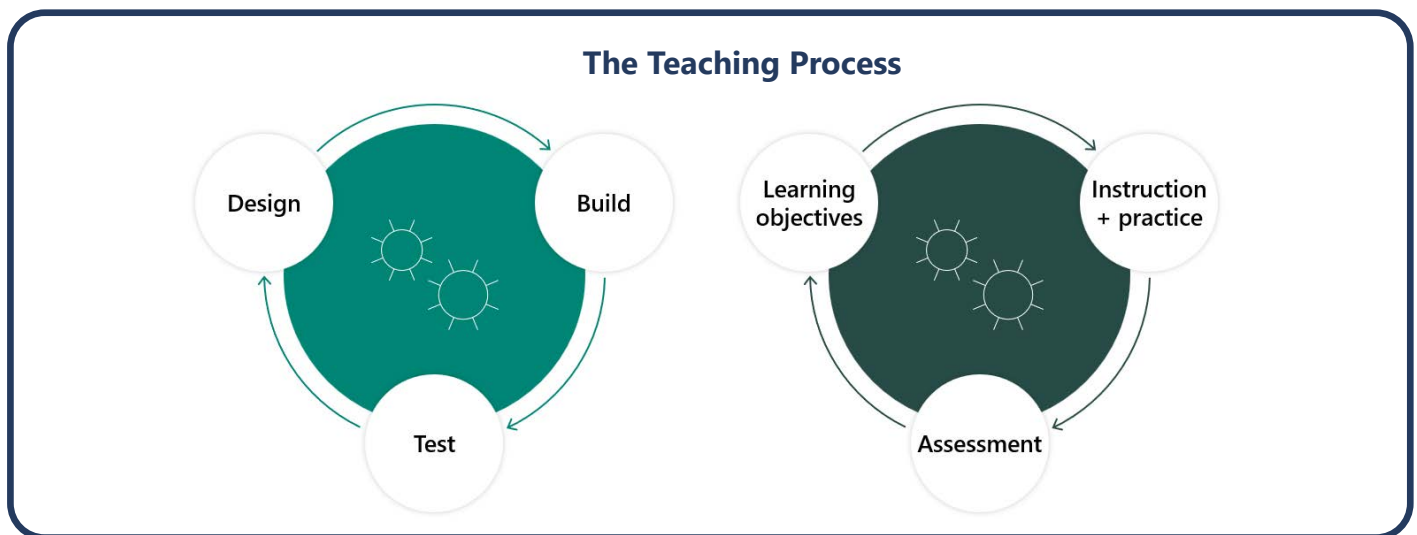
### Connect CS to students' real lives

Grounding concepts in the real world is good practice in any classroom. Ensure your CS lessons take into account students' cultural experiences and realities. Look for connections to current sociopolitical issues, such as net neutrality or voting security, or relate computing to pop culture.

- Share your own story and bring in other CS instructors or professionals. Look for stories and experiences about using CS that will be meaningful and relatable to your students.

## Use culturally responsive teaching practices

- We know that one hurdle to more diversity in CS is students' own belief systems about who can succeed in computer science. Encourage students to reflect on their perspectives and potential biases and challenge yourself to do the same.
- Maintain high expectations for all students to counter stereotypes about who should excel in CS. Build relationships with students to identify opportunities to connect learning to their personal experience.
- Differentiate learning for students who might need different types of support, such as English-language learners. Use the [Universal Design for Learning](#) framework to give students multiple pathways to access information and demonstrate their understanding.
- Acknowledge how issues of power and privilege, particularly in the CS realm, have a history of marginalizing groups of people through innovations and normative professional practices. Encourage students to identify how policies and collective agency might disrupt these forces.



## Expand your own knowledge and skills

Expand your lens through professional learning networks, continuing education and online communities. Join the [Computer Science Teachers Association \(CSTA\)](#), explore University of Washington's [DO-IT Communities of Practice](#), or follow blogs or social media feeds relevant to CS education.

- Consider online courses such as [Strategies for Effective and Inclusive CS Teaching](#) from WeTeach\_CS, where teachers can gain the insights, skills and strategies needed to create equitable CS programs.
- Consider developing your leadership in equitable teaching practices and advocacy with programs such as the [CSTA Equity Fellowship](#).



## Take Action

### Educators

- Find ways that you can ground CS lessons in ways relevant to learners such as through connections to pop culture or sharing stories about using CS.
- Join a CS educator community to further your own knowledge and skills.



# Accessibility in the computer science classroom

There are many ways to make CS classrooms feel inclusive for students with disabilities. Below are some best practices, classroom tools, and teacher development resources.

## Design for everyone

There are many ways to make your CS class or computer lab accessible for everyone. Access Computing has helpful resources such as the [Equal Access: Universal Design of Computer Labs](#) guide to help you ensure every person using the space can do so easily. Below are some general tips referenced in the guide:

- Are all printed software and hardware documentation and other publications available (immediately or in a timely manner) in alternate formats such as Braille, large print, and electronic text?
- Are printed materials within easy reach from a variety of heights and without furniture blocking access?
- Do video presentations used by the lab have captions? Audio descriptions?
- Are adequate work areas available for both right- and left-handed users?
- Are key documents provided in language(s) other than English?
- Is a trackball available for those who have difficulty controlling a mouse?
- Are staff members familiar with the availability and use of the Telecommunications Relay Service, assistive technology, and alternate document formats?
- Can controls on computers, printers, scanners, and other information technology be reached from a seated position?



“ To make your lab accessible, employ principles of universal design (UD). Universal design means that rather than designing your facilities and services for the average user, it is designed for people with a broad range of abilities, disabilities, ages, reading levels, learning styles, native languages, cultures, and other characteristics. ”

– **Disabilities, Opportunities, Internetworking, and Technology (DOIT)**



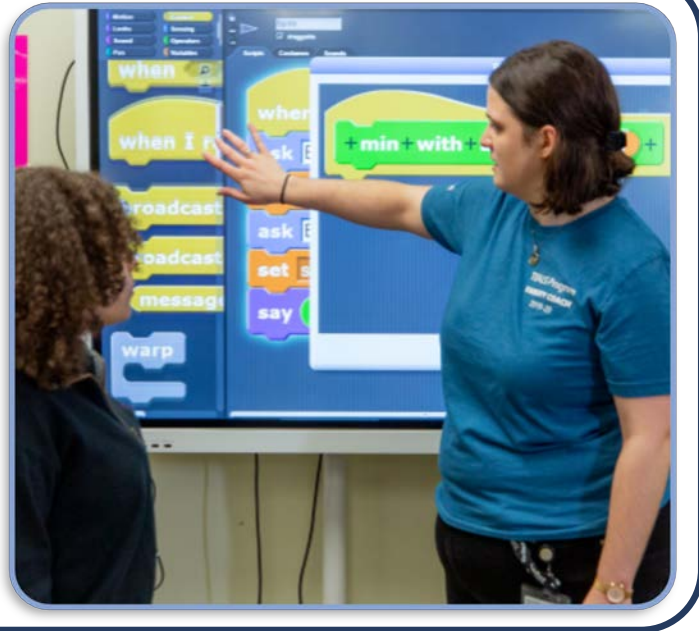
## Making the classroom more accessible for students with disabilities

- Evaluate the accessibility of your lab and develop accommodation strategies with the [Universal Design of Computer Labs](#) resource from AccessCSforAll.
- Ensure your learning material is accessible with evidence-based practices to help explore ways to increase participation of individuals with disabilities in computing in the [AccessCSforALL Knowledge base](#).
- Learn strategies from K-12 computer science (CS) teachers who specialize in teaching blind and visually impaired students, deaf and hard of hearing students, and learning disabled and neurodiverse students with [Strategies and Resources for Including K12 Students with Disabilities in CS](#).



“ 4.8% of K-12 students have a specific learning disability. There’s a lot of variability among these students, that may include difficulties in the oral and/or written expression, reading (decoding, fluency, and/or comprehension), auditory processing working memory, Dysgraphia/ Dyscalculia, and executive functioning (Organization of time, materials, and/or information). ”

– AccessCSforAll



## Professional Development Resources

- Join the [Disabilities, Opportunities, Internetworking, and Technology \(DOIT\) Community of Practice](#) which focuses on practices that promote the participation and success of people with disabilities in education.
- Learn how to help students with learning disabilities with the [Teaching Practices Guide: Improving Accessibility for Students with Learning Disabilities & ADHD: The Computer Science Principles \(CSP\) Course](#).
- The [Microsoft Computer Science Curriculum Toolkit](#) provides recommendations for CS educators to build a CS curriculum that supports students with learning disabilities.
- Teachers can earn a badge when they take the [Closing the STEM Gap](#) on Microsoft Learn to learn how to help close the gender gap in computer science.
- Leverage the [Computer Science Professional Development Guide](#) to find steps you can take today in support of CS programming.



## Take Action

### Educators

- Watch the [Universal Design of Computer Labs video](#) by Disabilities, Opportunities, Internetworking, and Technology (DOIT).
- Download and print the [Universal Design of Computer Labs checklist](#) to make your computer lab more accessible.

### Support Staff

- Watch [Universal Design of Computer Labs video](#) by Disabilities, Opportunities, Internetworking, and Technology (DOIT).

## Resources for educators

Leverage these resources as your school implements computer science classes and works to implement inclusion best practices.

### **Making classrooms more inclusive**

---

#### **[CSEd Week Heroes](#)**

Leverage the CS Heroes resources to help students explore a future in computer science.

#### **[Culturally Responsive-Sustaining CS Education Framework](#)**

The Culturally Responsive-Sustaining CS Framework builds upon decades of theory and research on culturally relevant and responsive pedagogy across disciplines and was developed in partnership with researchers, practitioners, teachers, students, and other education advocates.



### **Instruction and pedagogy**

---

#### **[Universal Design for Learning](#)**

Universal Design for Learning (UDL) is a framework to improve and optimize teaching and learning for all people based on scientific insights into how humans learn.

#### **[University of Florida's UDL4CS](#)**

The University of Florida College of Education's UDL4CS Professional development hub has resources for implementing UDL in CS education.

### **Curricular Materials**

---

#### **[MakeCode for micro:bit Curriculum](#)**

Leverage Microsoft MakeCode to challenge students with projects that build creativity, computational thinking, and collaboration skills.

#### **[The Quorum Programming Language](#)**

A significant majority of programming languages and tools used across the United States are not accessible to people with disabilities. Quorum provides a wide variety of ways to learn, including online, offline, and through different modalities. Quorum includes tracks for learning through visual, audio, and robotics programming.



### **Accessibility in the computer science classroom**

---

#### **[Web Design and Development \(WebD2\)](#)**

WebD2 is a free project-based introduction to web design course curriculum that integrates accessibility into lessons on site planning, HTML coding, and other aspects of web design.

# Contributors

Information in this guide was developed in collaboration with the following partners.

- **Callista Chen**, Reboot Representation
- **Dr. Allison Scott**, Kapor Center
- **Dr. Carol Fletcher**, The University of Texas at Austin, WeTeach\_CS
- **Dr. Danny Voloch**, Girls Who Code
- **Dr. JeffriAnne Wilder**, National Center for Women & Information Technology (NCWIT)
- **Dr. Leigh Ann Delyser**, CSforALL
- **Dr. Lien Diaz**, Educational Innovation and Leadership, Georgia Tech College of Computing
- **Dr. Maya Israel**, University of Florida
- **Dr. Richard Ladner**, University of Washington, Center on Research and Education on Accessibility
- **Edie Cheng**, National Center for Women & Information Technology (NCWIT)
- **Emily Mcleod**, Code.org
- **Jake Baskin**, Computer Science Teachers Association (CSTA)
- **Jessica Jahnke**, Girls Who Code
- **Neil Barnett**, Microsoft Accessibility Team
- **Tim Faiella**, National Center for Women & Information Technology (NCWIT)

**Thanks to all who contributed to the previous version of this guide.**

Leslie Aaronson, Jake Baskin, Doug Bergman, Callista Chen, Dr. Leigh Ann DeLyser, Dr. Lien Diaz, Dr. Maya Israel, Dr. Amy Ko, Frieda McAlear, Brook Osborne, Dr. Allison Scott, Dr. JeffriAnne Wilder

## **About Microsoft Philanthropies TEALS Program**

Microsoft Philanthropies TEALS (Technology Education and Learning Support) Program puts technology professionals from across the technology industry into high school classrooms to team-teach computer science with classroom teachers. Through TEALS, technology professionals share their knowledge with teachers while students benefit from learning how computer science is used in the workplace. TEALS operates in 8 countries, including 30 U.S. states and Washington, DC. Since its inception in 2009, over 100,000 students have participated in a TEALS class. Learn more at [Microsoft.com/teals](https://Microsoft.com/teals).

## **Microsoft Philanthropies**

Microsoft believes economic growth should be inclusive—for every person, organization, community, and country. This starts with increasing access to digital skills and opportunity and extends to closing the data divide and supporting public health. Learn more about our [actions to support inclusive economic growth](#).

# Microsoft Guide to Inclusive Computer Science Education

