

Tips for Introducing CS



CSTeachingTips.org/Tips-for-Introducing-Computing

1 Describe programs as instructions
to connect programming with students' everyday life.

Programming is just bossing a computer around!

2 Point out products of CS
to help students see computing around them.

Traffic lights are controlled by computer programs!

3 Explain that bugs are expected
to encourage students to embrace mistakes.

Even for professional programmers, programs never work at first!

4 Introduce synonyms for CS
to demystify terms describing CS jobs.

There are lots of names to describe doing CS!

5 Promote collaboration & creativity
to dispel stereotypes about CS.

You'll work together today like computer scientists do!

6 Model programming
to show problem solving strategies.

That didn't work like I expected! What could I try next?

7 Publicize resources for learning CS
to help students see how they can continue learning CS.

If you want to learn more CS, you can find resources at ...

csteachingtips

1

Describe programs as instructions

Explain that programming languages are just languages that the computer understands and programs are just instructions for the computer to follow. This can help students see that programming relates to their experiences giving and receiving instructions.

2

Point out products of CS

Computer scientists are involved in the creation of most everyday objects. Students might know that computer scientists make apps, but might not realize that computer science is behind lots of everyday objects (e.g. traffic lights, microwaves, cash registers). You can have students list things that are important to them and discuss how those things rely on computers (and therefore computer science).

3

Explain that bugs are expected

School sometimes reinforces the idea that mistakes are bad. Instead, focus on how mistakes are part of the learning process and that this is particularly true in programming. Explain that when professional programmers write code, it rarely works the first time and that the most important thing is continuing to try things even if they don't work at first. Give examples of the errors that students might see to help them decode the error message. Explain that nothing they do will damage to the computer.

4

Introduce synonyms for CS

Students have likely heard a lot of terms about CS: programmer, hacker, software engineer, coder, computer scientist, or developer. To demystify CS, explain that the differences between these terms aren't particularly important. Tell students see that CS has a lot of specialties that have different balances of programming, design, math, and problem solving.

5

Promote collaboration and creativity

Try to challenge stereotypes of CS as solitary and boring by providing opportunities for collaboration and creativity. Tell students that computer scientists always work in teams to solve big problems. Consider looking up the number of employees at a big tech company students have heard of to illustrate that computer scientists work together. Explain that computer scientists use creativity when they're inventing new products and when they're coming up with creative ways to solve problems.

6

Model programming

When students are new to programming, they don't know what to expect. Display your screen while programming to model the process of tinkering, reading error messages, making and fixing mistakes, and problem solving within a programming environment. Students likely won't be able to replicate all of the things you demonstrated, but it can be helpful for them to see what activities programming involves.

7

Publicize resources for learning CS

Students might not know how they can pursue more CS learning outside of class. Give examples of what they might want to learn to make and help them identify resources for learning CS online. There are lots of free resources available online!