

LESSON NAME:

# Intro to the Art of Computer Science

Lesson time: 45–60 Minutes : Prep time: 15 Minutes

**Main Goal: Give the class a clear understanding of what computer science is and how it could be helpful in their lives.**

## OVERVIEW:

This lesson will introduce the concept of “Computer Science” and explain what a “Computer Scientist” does. It will also give students an opportunity to assume the role of a computer scientist. By the end of the activity, each student will have a take-home reminder of what he or she learned.

You may be tempted to skip teaching young children the complicated vocabulary words that are associated with the lessons that follow, but presenting them in a fun and playful light now is far more productive than waiting for more advanced opportunities.

## OBJECTIVE:

Students will —

- Learn the difference between programming, computer science, and computational thinking
- Understand that a computer is a tool and not an excuse to turn off your brain

- Learn to be responsible computer users
- Discover that computer science can change the world

## MATERIALS:

- Binary Decoder Key
- Graph paper or Binary Strips
- Markers

## PREPARATION:

Prepare one Binary Strip (by printing the page provided and cutting it into strips) or piece of graph paper per person

- K-3 students use a Binary Strip with 1 letter
- 4-6 students use a Binary Strip with 2 letters
- 7-8 students use graph paper

Print or otherwise display Binary Decoder Key

Read through lesson intro and modify wording to appeal to age group of your students.

**VOCABULARY:**

**Automate** — To make something happen automatically (without help from people)

**Computer Science** — The art of blending human ideas and digital tools to increase problem solving power

**Computer Scientist** — A person who is skilled at modifying problems for digital solutions

**Data** — Information, including: facts, samples, names and numbers

**Environment** — The world we live in

**Interface** — The way something allows you to connect with it

**Open Source** — Software that is created for free use by everyone

**Programming** — Writing instructions for a digital tool

**Simulation** — Pretending to be (a stand-in for) the real thing



Students have the opportunity to assume the role of a computer scientist

**INTRODUCE:**

Computer science is one of the most amazing art forms in the world. Not only is it incredibly expressive — allowing for an infinite number of combinations of words, images, and ideas — but when done well, it produces a functional and useful product that can entertain and inform the masses.

When most people think of computer science, they generally envision programming. Often, they assume that the field is full of people who sit alone at their computers all day, staring at large screens and drinking coffee. In reality, computer science is an engaging and collaborative activity that includes far more than the ones and zeros for which it has become famous.

The art of computer science often starts with a problem that needs to be solved. Maybe there is too much information produced by a study for one human to be able to sort through it all in a lifetime. Perhaps an elderly person finds it hard to physically connect with long-distance relatives. There might even be an organization that needs to find the best route to get the most miles out of its airplanes using the least amount of fuel possible.

**Optional Question:**

**Q:** Can you think of a problem that a computer scientist could solve?

Once an issue has been identified, it may seem too overwhelming to tackle. That's why computer scientists learn to look at individual pieces, instead of just the big picture. Chopping a task up into manageable pieces is a great way to make progress through a series of little successes.

Figuring out how something should work is much different than actually getting it to work. That's where computer simulation and automated performance come in. Whether you're trying to simulate a real world experience, or create an impossible one, computers are an extremely helpful platform.

**Optional Question:**

**Q:** Have you ever seen a computer simulation before? Describe it.

If your solution runs on a computer, then it is expected that people will need a way to interact with it. This introduces the need for a beautiful, sleek, and artistic interface. A program must also provide direction to its users through graphic elements, a spoken script, or menus and buttons. All of these bits require careful forethought, planning, and incorporation.

So, you see, computer scientists aren't just programmers. They're creative problem solvers, psychologists artists, authors, and yes, even actual scientists. Every industry needs computer scientists. They're needed in biology, video gaming, schools, health care, public service, and anywhere that innovation is involved.

**Optional Questions:**

**Q:** What other jobs could benefit from the help of computer science?

**Q:** Are there any jobs that you can think of where understanding computers would not be useful? Do you think there will be more or fewer of those in 5 years?

**RESPONSIBILITY:**

Computers are great for repetitive tasks. They're also fantastic for providing interactive entertainment. The danger comes when people get so caught up in using a computer that they stop moving and thinking for themselves.

When you learn how to use a computer at a very young age, it is extremely important that you keep in mind the value of balance. When it comes to technology and human-power, balance keeps you on top of the very best things about both sides. Let's look at this three different ways:

**► Technology vs. Physical Movement**

Some people would prefer to spend time on video games, watching television, or surfing the Internet rather than going outside to play. Technology is fun, but if we give up our opportunities to run, jump and climb, then we could miss out on valuable experiences later in life. Make sure you balance your tech by spending an equal amount of time on physical activity. Teachers should encourage "stretch breaks" every thirty minutes if a lesson requires that students be focused on technology for longer than a half-hour.

Similarly, teachers should encourage students to follow the 20/20/20 rule.

That is: **"Every 20 minutes, look at least 20 feet away for at least 20 seconds."**

**Optional Questions:**

**Q:** What happens to your muscles when you stop using them?

**Q:** Your eyes have muscles, too. How do you think you could get them extra exercise?

► **Technology vs. Environment**

Technology uses a lot of resources. When we take advantage of technology, it is our responsibility to offset the damage that we do by being careful with our electricity other places. Be sure you turn the lights off when you leave a room, shut down and unplug your computer whenever you are done using it, and recycle any unwanted paper that you produce.

**Optional Questions:**

Q: What other electricity do you use that you could try to save?

Q: Is there anything else you could recycle to help offset paper use?

► **Technology vs. Philanthropy**

Computer Scientists make a lot of money. But when you can do something that most other people cannot, it isn't very responsible to take advantage of others because of it. It is far better to get in the habit of "paying it forward". If you learn computer science, make it your goal to teach at least one other person. Not only will you have helped brighten someone else's future, you will find that explaining computer science helps you to understand the lessons better yourself. If you ever have the opportunity, try to get involved in creating open source software for others.

**Optional Questions:**

Q: How would you help the world if you were a computer scientist?

Q: Who is the first person that you are going to teach this to once you learn it?

**ACTIVITY:**

Let the students know that the best computer scientists understand what it's like to "be" a computer. Tell them that they are going to learn to understand how computers read and store data.

Once students are situated (either in groups or individually) it is time to pass them their **Binary Strips**. Provide them with a copy of the **Binary Decoder Key** (or display a large copy in the front of the room). For younger kids, the entire exercise can be done as a class.

Choose a letter from the Binary Decoder Key. Display the letter as a sequence of colored boxes only, then challenge the class to use their key to figure out which letter you have selected. If the students have a difficult time, show them how to match the first four boxes to either the first half of the alphabet or the second half (this is why there is a small gap between the first 4 squares and the last 4 squares). This should narrow down their choices. Keep selecting letters until the majority of the class is able to yell out your letter at the same time.

When the students understand the way the key works, inform them that this is the same method that computers use to store data. Computers format everything (letters, numbers, images, even sounds) as some representation of on and off. Now, you are going to let them act like computers as they “store” letters of the alphabet as binary. K-3 students can use the binary strip with 1 letter to store the first letter of their names to the Binary Strips (or store the first letter of their school, if you are working as a class.) Students in grades 4-6 can use the Binary Strips with 2 letters to encode their initials. Older students can store their entire name on a piece of graph paper. Challenge older students to figure out how many squares they will need to store their first, middle, and last names altogether. (numOfLetters \* 8)

If there is time remaining at the end of the exercise, allow students to encode whatever they want, and have them switch their message with a classmate who can try to decode.

**Optional:**

Students may also use a 4x4 grid from the Binary Squares sheet to store their initials. Students can use one color for the letter of their first initial (putting the 8 boxes on 2 lines of 4 boxes instead of 1 line of 8 boxes) and a second color for their last initial. Teachers or students can attach this square to a magnet to have a neat way to keep their initials in binary!

**ADJUSTMENTS:**

**K-3:** Use Binary Strips with just 1 letter. Work in groups or as an entire class.

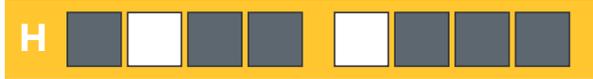
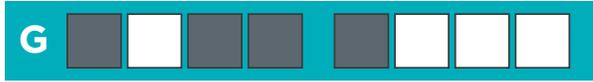
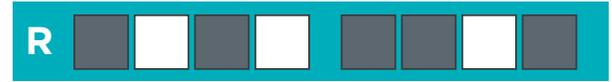
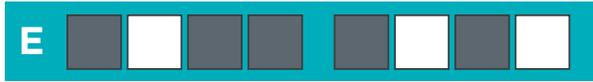
**4-6:** Use Binary Strips with 2 letters. Students can store their initials on these 2 rows. Students may want to work in groups.

**7-8:** Students should be encouraged to encode longer messages on graph paper.

**STEPS:**

- 1) Introduce computer science
- 2) Explain a computer scientist’s responsibility
- 3) Hand Binary Strips or graph paper to students
- 4) Quick binary encoding exercise to help students understand what it’s like to “be” a computer.

BINARY DECODER KEY



**BINARY STRIPS**

**Binary**

**Letter**

--	--	--	--

--	--	--	--

--



--	--	--	--

--	--	--	--

--



--	--	--	--

--	--	--	--

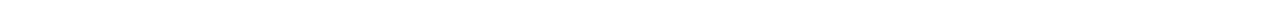
--



--	--	--	--

--	--	--	--

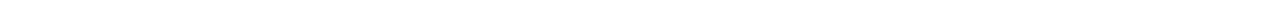
--



--	--	--	--

--	--	--	--

--



--	--	--	--

--	--	--	--

--

### BINARY STRIPS

Binary

Letter

--	--	--	--

--	--	--	--

--

--	--	--	--

--	--	--	--

--



--	--	--	--

--	--	--	--

--

--	--	--	--

--	--	--	--

--



--	--	--	--

--	--	--	--

--

--	--	--	--

--	--	--	--

--



--	--	--	--

--	--	--	--

--

--	--	--	--

--	--	--	--

--

## BINARY SQUARES

