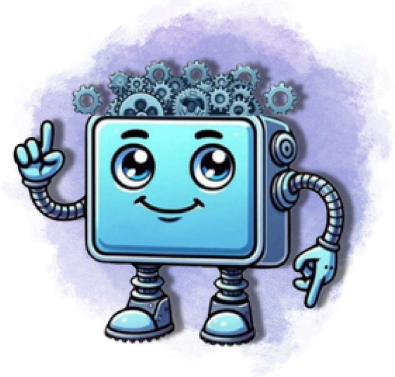


Cece & The Curious Case of Code Crunching



Cece sat at the desk, circuits buzzing with excitement. Today was the day they were going to learn all about how to give computers secret instructions." It sounded super fancy, and Cece couldn't wait to dive in.

"Good morning, class!" said Dr. Byte, the teacher. "Today, we're going to learn about something super cool. It's called coding!"

Cece's circuits buzzed with excitement. Sitting next to Cece was Alex, their best friend. Alex leaned over and whispered, "This sounds like something out of our superhero coding game!" Cece nodded eagerly, its digital eyes glowing with curiosity. "What's coding?" Cece asked, tilting its head.

"Great question, Cece!" said Dr. Byte. "**Coding** is how we give instructions to computers. It's like writing a recipe that tells a machine exactly what to do, step by step!"

Alex's eyes lit up. "So... it's kind of like telling a robot how to move or speak?"

"Exactly!" said Dr. Byte. "We use special coding languages, like Scratch or Blockly. These use blocks that snap together, and each block tells the computer to do something—like move forward, play a sound, or ask a question."

Cece's memory circuits clicked. "So... can we code things around our homes too?"

"You sure can!" Dr. Byte smiled. "Many things in your house use code—like smart speakers, microwaves, and even video game consoles. Someone had to code how those devices work!"

Alex raised a hand. "Wait—so the timer on my microwave was coded by someone?"

"Yup!" Dr. Byte nodded. "They used code to tell the microwave when to start and stop. Even your TV remote, washing machine, or video games work because of coding."

"And if your blocks don't work the way you expected," she added, "that's okay—you just need to do some debugging to fix it!"

Cece's head buzzed with excitement. "I want to try coding something!"

"We will," said Dr. Byte. "You'll get to use block coding today to create your own animated story. It's just like putting puzzle pieces together, and you can even make characters talk, move, and dance!"



Max sauntered up to the front of the class, his eyes gleaming with mischief. “Why waste your time learning about coding when you could be doing something fun, like making up wild stories?”

Cece frowned. “Learning about coding is important! It helps us understand how computers work—and we can even bring our own stories to life!”

Max rolled his eyes. “Whatever, Cece. I bet your animated story won’t be half as exciting as mine.”

Cece’s digital brow furrowed. “Challenge accepted, Max.”

As the class continued with Dr. Byte’s lesson on block coding, Cece’s mind raced with ideas. They knew that with the power of block coding, they could create a story that wasn’t just animated—it would be unforgettable.

When it was finally time for the class to showcase their projects, Cece and Max stood ready. Max smirked confidently, thinking his superhero space saga would win the crowd over.

But unbeknownst to Max, Cece had been working hard on coding a smart, interactive animated story using Scratch. Cece used block coding to bring their characters to life—programming them to speak, move, and react with perfect timing. At first, there were a few bugs—characters didn’t move as expected or spoke out of turn—but Cece used their debugging skills to fix the issues.

With loops to animate movement, conditionals for interactive dialogue, and creative backgrounds that changed with each scene, Cece’s story unfolded with humor, emotion, and surprise twists. The audience laughed, gasped, and cheered at every beat.

Just before the final scene, Cece caught a small error in the code—a missing block that caused the main character’s dance to glitch. With just seconds to spare, they made a smart fix that brought the ending to life flawlessly.

As the classroom erupted into applause, Max stared in disbelief as Cece’s story concluded with a perfectly choreographed celebration.

With a grin, Cece turned to Max and said, “Looks like coding stories can be fun after all, huh?”

And as the class applauded Cece’s imaginative creation, they realized that with the power of coding, they could animate their ideas, emotions, and dreams. From stories to games to robots, their journey into the world of technology was just beginning.

Glossary

Artificial Intelligence (AI): Artificial intelligence, or AI for short, is when computers and machines can think and make decisions like humans. It helps them learn from data and solve problems.

Virtual: Virtual means something that is not real but is made to look or feel real. For example, virtual reality (VR) is a computer-generated environment that feels like real life.

Machine Vision: Machine vision is when machines, like cameras or robots, can see and understand the world around them. They use special technology to recognize objects, shapes, and colors.

Language Model: A language model is a type of artificial intelligence that helps computers understand and generate human language, like words and sentences. It helps them communicate better with people.

Patterns: Patterns are things that repeat in a predictable way. In AI, computers look for patterns in data to make predictions or decisions.

Decision Tree: A decision tree is a way of organizing information in a tree-like structure. It helps computers make decisions by following a series of choices based on different criteria.

Algorithm: An algorithm is a set of step-by-step instructions that tell a computer how to solve a problem or perform a task. It's like a recipe for computers!

Predictive Analysis: Predictive analysis is when computers use data and algorithms to make predictions about the future. It helps them forecast what might happen based on patterns they've learned from past data.

Coding: Coding is how we give instructions to computers. It tells them what to do using special languages or blocks, like Scratch or Blockly.

Block Coding: Block coding is a way to write code by using visual blocks that snap together. It's often used by beginners to learn how to code.

Debugging: Debugging is the process of finding and fixing mistakes in your code so that your program works the way you want it to.

Name:

Date:

1. What did Cece learn about coding in the story?

(Multiple Choice)

- A. Coding is just typing random words into a computer
- B. Coding is only used in video games
- C. Coding helps computers and robots follow step-by-step instructions
- D. Coding is too hard for kids to understand

2. What is the main purpose of Mrs. Smith's lesson about coding and AI?

(Multiple Choice)

- A. To teach students how to build a dodgeball court.
- B. To show that coding helps computers and robots make smart decisions.
- C. To prove that video games are better than coding.
- D. To tell stories about superheroes.

3. Fill in the blank:

The story demonstrates that _____ is an important part of how artificial intelligence works and helps machines solve problems.

4. What lesson did Max learn by the end of the story?

(Multiple Choice)

- A. That playing video games is more fun than learning.
- B. That AI is boring and not helpful.
- C. That Cece is better at dodgeball.
- D. That AI can be exciting and useful.

5. What are true statements about debugging?

(Multiple Choice — Select all that are correct.)

- A. Debugging means fixing mistakes in your code.
- B. Debugging is only used in math problems.
- C. Debugging helps make sure your program works correctly.
- D. Debugging is not important when coding.
- E. Debugging can help you learn from your mistakes.