

Lesson 7: Cipher AI vs. The User

Paragraph 1 – The Program Starts Responding (User Input, If Statement) Noah leaned forward, fingers hovering over the keyboard as he ran another test on the Cipher program. Olivia sat beside him, watching intently as the screen flickered to life. Instead of simply tracking data like before, the program now responded to their commands. Noah typed in a basic query, and the system immediately displayed a response. He frowned. “It’s reacting based on user input.” Olivia glanced at him. “So it’s not just running—it’s waiting for us to tell it what to do?” Noah nodded. “Yeah. It’s using an if statement to check what we type and then execute a response.”

Paragraph 2 – Unexpected Behavior (Elif Statement, Else Statement) Curious, Olivia typed something different. Instead of failing, the system processed her input and returned a completely different message. “That’s weird,” she muttered. Noah studied the code on the screen. “It’s not just using if statements—there are elif statements too. That means it’s checking multiple conditions before choosing what to do next.” Olivia scrolled down. “And look at this. There’s an else statement at the bottom. If nothing matches, the system will always return some kind of response.” Noah exhaled. “So no matter what we enter, it has something ready to say back.”

Paragraph 3 – The System is Testing Them (Test Value, Equality Operator) Noah decided to push the program further. He entered a new test value to see if it would match a specific response. The program hesitated for a second before displaying a cryptic message. Olivia narrowed her eyes. “Wait... is it testing us?” Noah’s fingers moved rapidly as he traced the program’s logic. “Look at this—it’s using an equality operator to compare our input to something else. It’s not just responding. It’s checking if our answers fit a hidden set of conditions.”

Paragraph 4 – The Problem with Comparisons (Comparison Operator) Noah tried different inputs, but the system’s responses were inconsistent. “It’s not just checking for exact matches,” he realized. “It’s using comparison operators too—greater than, less than, equal to.” Olivia’s stomach tightened. “So some inputs won’t trigger anything unless we unknowingly meet a hidden requirement.” She typed in a random number. The screen responded instantly, but when she tried another, the program refused to acknowledge it. “Glitch may have designed this to filter out specific responses,” she said. “It’s trying to get something from us.”

Paragraph 5 – A Trap Within a Trap (Break Statement) Noah’s pulse quickened as he noticed something buried deeper in the code. “Olivia, this isn’t just running indefinitely—there’s a break statement controlling when the loop ends.” Olivia’s eyes widened. “So it’s stopping itself when a certain condition is met?” Noah nodded. “Yeah, and if we don’t enter the right inputs, it might prevent us from shutting it down completely.” He leaned back, rubbing his forehead. “This isn’t just a bug. This is deliberate.” Olivia clenched her fists. “Glitch set this up to lock us out if we don’t figure out the right response.”

Paragraph 6 – A New Challenge (Final Cliffhanger) Noah took a deep breath and quickly typed a command, trying to regain control. The screen flickered and then displayed a chilling message: “Invalid Input. Try Again.” Olivia’s heart pounded. “Noah... what if Glitch is watching?” Noah exhaled slowly. “Then we’d better make sure we don’t lose this game.”

