

Reading and analysing unfamiliar code

40 minutes

Strand 3: Coding at the next level

Outcome 3.9

CURRICULUM OUTCOMES

Outcome	What the specification asks for
3.9	Analyse code to determine its function and find errors.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Read an unfamiliar program and work out what it does by tracing it by hand (LO 3.9)
- Find and fix an error that produces a wrong answer without any error message (LO 3.9)
- Judge an AI assistant's explanation of code against your own trace, deciding where it is right, vague or wrong (LO 3.9)

BEFORE THE BELL

- Students logged in at devices with Scratch open at scratch.mit.edu
- Paper or copies ready for trace tables; this lesson is traced by hand before anything is run
- Have the broken copy ready to display on the board so the class reads the same version at the same time

Reading and analysing unfamiliar code

HOW THE LESSON RUNS

4 min	introduction	Start: what we're building
2 min	content	Create a new Scratch project
3 min	content	Make a list of scores
4 min	content	Add up the scores
3 min	content	Fix the counting order and run it
3 min	content	Here is a broken copy
4 min	content	Trace it by hand
5 min	content	Fix it at your device and prove it
3 min	content	What the AI assistant said
3 min	activity	Your turn
3 min	content	Make sense: what worked, what broke
2 min	content	Reflect
1 min	summary	What you covered

TEACHING MOVES

- Hold the line on tracing before running. The whole lesson turns on students predicting 12 themselves; once the computer has said it, the point is gone
- Do the first pass of the trace table on the board together, then let them finish the remaining three alone
- On the AI explanation, do not ask 'is it right'. Ask which sentence is wrong and how they know, so the answer has to come from their own trace
- In the swap task, insist the bug is found by tracing rather than by guessing or by clicking, which is the skill being assessed

WHAT TO WATCH FOR

- Running the broken copy before tracing it, which removes the only evidence the lesson is built on
- Confusing set count to 1, which happens once, with change count by 1, which has to happen every pass: this is the planted bug and worth naming slowly
- Reading no error message as proof the program is right, which is exactly the habit this lesson exists to break
- Breaking their own program so badly it crashes or does nothing; the task is a wrong answer that still runs cleanly

DIFFERENTIATION

- Support: hand out the trace table with the first pass already filled in, so the pattern is visible before they continue it
- Extension: rewrite the AI explanation in two sentences so it is actually correct about this program
- Extension: break the program a second time in a different way and describe what each bug does to the answer