LESSON 8

Trace, predict and fix

40 minutes | Strand 1: Introducing computer science | Outcome 1.8

WHAT THIS LESSON DOES

Learn how to use trace tables to step through programs by hand, predicting what they will output before running them. Then trace a buggy program to locate the error and fix it so it produces the correct result.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
1.8	Test code to find and manage errors.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Step through a program by hand with a trace table to predict what it will do before running it
- Find and fix where it goes wrong (LO 1.8)

BEFORE THE BELL

- Prepare paper or notebooks for creating trace tables.
- Print the blank trace-table worksheet (two grids per sheet) if you want a shared layout; otherwise students can copy the board grid.
- Ensure the worked example and programs are ready to display.
- Have Scratch ready on the IWB for the teacher-led run-checks on Programs A and B, and on student machines for the mandatory pair rebuild of the fixed coin program.
- Paper or notebooks ready for trace tables
- Board space for two short programs; Scratch ready on the IWB to run teacher checks
- Scratch ready on the IWB and on student machines for the mandatory pair rebuild of the fixed coin program

Trace, predict and fix

HOW THE LESSON RUNS

4 min	introduction	Start
5 min	content	Trace tables and predict-then-run
4 min	activity	Program A: trace it and predict
2 min	content	How a decision looks in a trace table
5 min	content	Program B: trace a decision
3 min	activity	The buggy coin program
8 min	content	Trace it, find the bug, fix it
3 min	content	What done looks like
3 min	content	Make sense
2 min	content	Reflect
1 min	summary	What you covered

TEACHING MOVES

- Draw a shared Step | Instruction | variable grid on the board and leave it up for the whole lesson (add Notes when you model the if-row).
- Model only the linear worked example in the content step; model the mini if-row on the board for about one minute immediately before Program B (the same filled example is now in student copy as a callout).
- Pace Program A as a whole-class pass with one IWB Scratch check; keep Program B as paper-only pair work with a single teacher-run Scratch reveal. After each reveal, allow a short moment for the Final output box so late finishers can self-check from the board.
- For the coin bug task, treat buggy trace + diagnosis sentence + fixed instruction + pair Scratch rebuild-and-run as must-do; second full paper trace and comment are stretch. Protect about five minutes at the end of that block for the live rebuild.
- Keep the reflect section as a short discussion only.

WHAT TO WATCH FOR

- Updating multiple variables in one row; instruct to handle one instruction at a time.
- Overlooking the set command and treating it as change; highlight the difference between set and change.
- Skipping steps in the trace; encourage following the program order strictly.
- Guessing the if branch without filling the decision row; point them back to the mini if-row callout above Program B.
- Deleting step 4 of the coin program and landing on 6; remind them the target is 5, not just closer.
- Jumping into Scratch on the coin task before a written fix; send pairs back to the paper steps first.

DIFFERENTIATION

- Support: Hand out the blank trace-table worksheet with variable columns pre-labelled (points, lives, coins).
- Support: Allow pair work for the bug-finding activity; if a pair is still on paper near the end, let them join a neighbour's machine for the mandatory Scratch confirm so nobody skips the live run.
- Extension: Complete the second full paper trace and one-line comment; or design a simple program and trace it for peers; or open an earlier saved project and trace a short stretch of blocks.