technology

Thinking like a computer: precise instructions (unplugged)

Lesson at a glance

Open as a literal robot: pupils call out how to reach a chair and you obey every vague word one step at a time until the plan fails. Name algorithm and bug, then groups build short floor-grid paths with sticky-note commands (forward, turn left, turn right), dry-run, and debug a shared public run. The algorithm-sequencer interactive repeats the same thinking on screen. Everyday examples (recipe, packing a bag) feed the Investigation Journal record of one ordered sequence, its buggy step, and the fix.

Learning objectives

- Write a precise ordered sequence of instructions for a simple task
- Spot where a missing or wrong-order step causes a bug and fix it
- Explain that computers and robots only do exactly what they are told

Before the bell — prep

Before pupils arrive, clear floor space and tape a 4-by-4 grid of ~40 cm squares with Start and Target markers on a short 3–5 step path. Pre-count sticky notes per group (6 forward, 3 turn left, 3 turn right). Cue the algorithm-sequencer interactive on the IWB; sticky notes on the board work if it is down.

Materials

Item	Qty	Per	Source	Low-cost substitute
masking tape for floor grid	1	class	classroom	chalk lines on a hard playground surface, or large sheets of paper taped into a grid
start and target markers (cones, books or star cards)	2	class	classroom	two sheets of coloured paper labelled Start and Target
instruction sticky notes or scrap card (forward, turn left, turn right)	1	group	classroom	scrap paper squares labelled with simple arrows
Investigation Journal page	1	pupil	provided printable	a blank page ruled into three parts: steps, bug, fix

Lesson 26

Safety watch-point

Calm walking only on the grid—no running. Tape edges can trip; keep bags and chairs clear of the floor space and walking lane.

Teaching moves

- **Getting Started:** Stand off the grid facing a chair a few metres away. Accept the first instructions and obey them literally—one tiny step for "go over there," one step only for "walk forward," endless spin for "turn." Freeze after 60–90 seconds and ask why you never reached the chair; land on unclear wording, not pupil fault.
- **Exact words, exact order:** Keep this board beat to about five minutes. Bridge once from the chair walk—Was that an algorithm yet? Where was the bug?—then name algorithm and bug and move straight to the floor so the words stick to action. Save "debugging" to name live during the floor run.
- **Direct the robot teacher:** Demo a quarter-turn on the spot, then model a short path with a planted bug, fix it, and re-run. Groups lay sticky notes for the same start, facing and target; hard cut-off after ~8 minutes. Every group must dry-run (finger-walk or quiet floor step) before one public robot run where the class names the buggy step, fixes it, and watches the re-run.
- **Order the steps on the board:** Open the algorithm-sequencer interactive in explore mode. Pupils call out; you or a volunteer drag forward / turn left / turn right and press run. Keep at least one failed run so debugging stays visible. Link aloud: same thinking as the floor, robot now on screen.
- **Instructions older than computers:** Elicit recipe, packing a school bag, or gate-to-shop directions. Ask what "bug" appears if order slips (lunch crushed, bake before mix) and how that matches a missing turn. Land on: computers and robots only do exactly what they are told, in order.
- **Record our algorithm:** Hand the Investigation Journal page. Every pupil records three beats: ordered steps (group dry-run or shared class algorithm), the named buggy step, and the fix. Oral pair-share first; arrows and numbers fine. Check that "bug" points to a specific step, not "it broke."

What it should show

A correct sequence of forward / turn left / turn right should land the robot on the target without leaving the grid. Typical bugs are a missing turn, a turn the wrong way, or an extra forward that walks off the edge. If a group's path "works" only when someone shuffles mid-run, they changed facing or skipped a sticky note—re-run from a still start reading every step aloud.

Misconceptions & interventions

- **"The robot is being silly" or "I'm just bad at this" when the path fails.** — Freeze on the wrong square and say the robot did its job—it only follows the list. Point to the exact sticky note to change; name that fix as debugging and re-run so the algorithm, not the child, takes the blame.
- **Pupils treat "turn left/right" as stepping onto a new square, or "forward" as walking several squares.** — Re-demo on one square: turn = pivot a quarter-turn on the spot; forward = one square only. Have the class copy one left and one right on the spot before they rebuild the sticky-note line.
- **A bug is described vaguely—"it broke"—instead of naming which step went wrong.** — Ask "Which number step?" and have them point to the sticky note. Require the journal line to say e.g. "step 3 should be turn right," then the fix.

Differentiation

Emerging	Developing	Proficient
• Offer a partly ordered sticky-note path with one deliberate gap to fill, and allow numbered arrows instead of full sentences on the journal page. • Pair at the teacher table for the dry-run: adult reads each note while the pupil finger-walks the desk grid sketch.	• After a successful dry-run, challenge the group to find a second shorter path to the same target using the same three commands only.	• Ask them to critique the shared class algorithm: name one step that could be removed or swapped and predict what the robot would do, then test on the dry-run grid.

Cross-curricular hook

Link to English procedural writing—same precise ordered steps as a recipe or directions from the school gate.