technology

Computational thinking unplugged

Lesson at a glance

Open by giving a deliberately vague instruction such as “Go over there” near the classroom door, then name why a robot would get stuck. Model one fail-then-fix cycle on the demo floor grid using only forward, turn left and turn right. Groups write and debug card sequences on desk mats with a robot token and one wall square, then record the working algorithm and the bug fix in the Investigation Journal. Close by sequencing a different 5-by-5 path together on the algorithm-sequencer interactive and pinning the words decomposition and debugging to the class’s own stories.

Learning objectives

- Break a problem into smaller steps and write a precise sequence of instructions
- Spot where an algorithm went wrong and fix the step that caused the bug

Before the bell — prep

Night before: tape a short demo floor grid and lay out one desk mat per group of 3–4 with a robot token, start/goal markers, one wall square and command cards (forward, turn left, turn right only). Fix the robot’s start facing on every mat. Clear bags and chairs from the floor-grid area.

Lesson 25

Materials

Item	Qty	Per	Source	Low-cost substitute
masking tape for one demo floor grid	1	class	classroom	chalk on the yard or rope laid in a grid outdoors if weather allows
desk grid mats (simple 4-by-4 grid on paper or card)	1	group	classroom	grid drawn in pencil on scrap paper or whiteboard per group
instruction cards (forward, turn left, turn right)	1	group	classroom	pupils write the three commands on scrap paper strips at the start of the activity
robot tokens (cube, rubber, or small toy)	1	group	classroom	a board rubber, counter, or folded paper arrow that shows facing direction
start and goal markers (paper squares, cones, or coloured counters)	1	group	classroom	two different coloured books, sticky notes, or beanbags
obstacle marker for the wall square	1	group	classroom	a rubber, block, or folded card placed on one square

Safety watch-point

No running on the floor grid; tape edges can trip — step inside squares. Keep bags and chairs clear. Desk tokens stay on the table.

Teaching moves

- **Getting Started:** Stand near the door and give one vague instruction such as "Go over there." Collect two or three pupil ideas about what was missing — direction, number of steps, when to stop — then frame today's job as writing steps precise enough for a robot that cannot guess.
- **Today's words: algorithm and bug:** Write the two definitions on the board and leave them up. Anchor the path on the demo floor grid so every pupil sees the same short journey, then think aloud through one full fail-then-fix cycle: deliberately omit the turn, name the bug, insert the missing turn and re-run. Allow only forward, turn left and turn right — reject "go to the star" as a step.
- **Guide the class robot:** Assign writer, reader, robot and checker jobs out loud (combine roles in threes). Writers lay cards before anyone moves the token; reader says one command at a time; robot moves literally. On failure, freeze — checker names the step, change only what is needed, re-run from the start. Leave working sequences laid out; do not stop mid-run for journal writing.
- **Record your algorithm:** Move straight to the Investigation Journal while cards are still on the desks. Check that steps are only the three allowed commands, the bug is named as a specific step, and the fix is the changed or inserted command. If a group succeeded first time, ask them to record a deliberate bug-and-fix on the same mat.
- **Order the steps on the board:** You drag the blocks on the algorithm-sequencer; pupils call the order. State once that the screen is a 5-by-5 with a wall — different from the desk 4-by-4 mats, so desk sequences will not simply copy across. Run once with a deliberate straight-line bug into the wall, then rebuild a working route together, pressing for the smallest change.
- **Where did it go wrong?:** Draw out two or three group stories from the desk runs. After one clear story, pin the labels: breaking the journey into small steps was decomposition; finding and fixing the wrong step was debugging. Prompt with "Which step caused the bug?" and "Did fixing one step create a new bug elsewhere?"

What it should show

Working desk algorithms use only forward, turn left and turn right and reach the goal without entering the wall square. Typical first-run bugs are a missing or late turn, or facing the wrong way after a turn. A sequence that "works" by sliding the token rather than obeying each card is a setup error — re-run with the robot moving literally one command at a time.

Lesson 25

Misconceptions & interventions

- **Pupils write "go to the star" or "go around the wall" as a single step.** — Hold up the three command cards only and ask: which of these would a robot understand? Have them replace the vague step with forward / turn left / turn right before running again.
- **When it fails, pupils rewrite the whole plan instead of naming the one wrong step.** — Freeze the token at the failure square and ask "which card got us here?" Change only that card, then re-run from the start so they see one-step debugging work.
- **Pupils think a bug means they failed, so they hide the first attempt.** — Point back to your own deliberate miss on the demo grid and say bugs are normal programmer work — the journal wants the wrong step named, not a perfect first try.

Differentiation

Emerging	Developing	Proficient
• Give a simpler path of three forwards and one turn, and sit briefly at the teacher table to help the writer lay the first card sequence before the group runs it. • Pair the checker role with a peer so naming the failed step is shared aloud.	• After one success, add a second turn or a longer route around the same wall on the desk mat. • Prompt: "Did you change one thing at a time, or rewrite the whole plan?"	• Fast finishers write a second algorithm for a new goal square on the same mat, or deliberately insert a bug and record the fix. • On the board run, ask them to justify the smallest change that clears the wall rather than rebuilding the whole sequence.

Cross-curricular hook

Link to English procedural writing — numbered steps must be precise enough for someone else to follow without asking what you meant.