

Computational thinking unplugged

Lesson at a glance

Open by giving a deliberately vague instruction such as “Go over there” near the door so pupils see why a robot needs precise steps. Model one fail-then-fix algorithm on the demo floor grid, then groups write sequences with only forward, turn left and turn right on desk mats using command cards and a token. They record the working list and the bug fix in the Investigation Journal, rebuild a different path on the algorithm-sequencer interactive, and close by naming decomposition and debugging from their own runs.

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

Tape or lay one demo floor grid at the front and one 4-by-4 desk mat per group of 3–4 with start, goal, one wall square, a robot token and command cards (forward, turn left, turn right only). Fix the start facing on every mat. Clear bags and chair legs from floor-grid edges before the bell.

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; step inside taped squares so lifted edges do not trip. Keep bags and chairs clear. Desk tokens stay on the table.

Teaching moves

- **Getting Started:** Stand near the door and say only “Go over there.” Ask why a robot would get stuck. Collect two or three ideas about missing direction, step count or stop point. Keep it light — no grid yet.
- **Today's words: algorithm and bug:** Project only algorithm and bug. Point clearly to start, goal (two ahead, one left) and path on the demo floor grid before asking for any step. Think aloud one full fail-then-fix cycle: predict a sequence, deliberately omit the turn, hit the wall, name the missing turn, re-run successfully. Allow only forward, turn left and turn right — block “go to the star” as one step.
- **Guide the class robot:** Assign writer, reader, robot and checker (combine roles in threes). Every group works its own desk mat — floor grid is demo/share only. Cards laid before any move; robot obeys literally; on failure freeze, name the step, change only what is needed, re-run from start. Leave working sequences laid out for the journal. Optional: one group replays on the floor grid while the class predicts the next step.
- **Record your algorithm:** Straight after the desk runs, send pupils to the Investigation Journal while cards are still out. Check: only the three commands in order; bug named as a specific step; fix written as the changed or inserted command. If a first run worked first time, have them deliberately bug and fix a second challenge on the same mat.
- **Order the steps on the board:** Drive algorithm-sequencer on the IWB. Say once the screen is 5-by-5 with a wall — different from the desk 4-by-4, so desk sequences will not copy across. Start col 0 row 2 facing east; goal col 4 row 2; wall col 3 row 2. Run a deliberate all-forwards bug into the wall, then rebuild a working route together from pupil calls. Prompt: smallest change that could work?
- **Where did it go wrong?:** Draw two or three desk-run stories, then pin the labels: breaking the journey into small steps was decomposition; finding and fixing the wrong step was debugging. Surface that vague steps cannot be debugged and that debugging is normal programmer work.

What it should show

Working desk algorithms use only forward / turn left / turn right and reach the goal without entering the wall square; bugs are usually a missing or late turn, an extra forward, or a turn the wrong way. On the board, a straight run of forwards hits the wall at col 3 — the robot must go around north or south then return to the goal row. A “surprising” success often means the robot pupil guessed instead of obeying the cards; re-run with literal moves only.



Misconceptions & interventions

- Pupils write one vague step such as “go to the star” or “go around the wall.” — Point at the three command cards only. Ask: which exact word would the robot hear? Have them replace the vague line with forward / turn left / turn right cards before any run.
- When it fails, pupils scrap the whole sequence instead of changing the one wrong step. — Freeze the token at the fail square. Ask which single card caused that move, change only that card, and re-run from the start.
- The robot pupil “helps” by turning or stopping even when the cards do not say so. — Remind the robot job: do exactly what is read, nothing kinder. A wrong card should walk into the wall — that is how the checker finds the bug.



Differentiation

Emerging	Developing	Proficient
• Give a simpler desk path: three forwards and one turn, with writer and checker combined beside you for the first run. • Allow the group to place cards while you act as reader so they only judge whether each move matches.	• Standard path with one wall; after one success, rotate jobs and re-run the same goal. • Prompt: did you change one thing at a time, or rewrite the whole plan?	• Add a second turn or a longer route around the wall; fast finishers write a second algorithm for a new goal square on the same mat. • In the share, ask them to critique whether another group’s cards could be followed by a stranger with no explaining.



Cross-curricular hook

Link to English procedural writing — numbered steps must be clear enough for someone else to follow exactly.