

technology

Program a micro:bit: follow precise instructions

Lesson at a glance

Open by holding up a micro:bit (or the MakeCode simulator) and asking what happens if instructions miss a step or name the wrong button. Pairs name a shared input–process–output plan (button A/B → heart or letter), then watch a live demo with one deliberate bug fixed on the IWB. They build and debug the same program on paired devices, and close by writing input, process, output and one bug-fix on the Investigation Journal page.

Learning objectives

- Program a micro:bit to show an output on a button-press input
- Frame the program as input, process and output and debug errors as they arise
- Explain that a micro:bit runs exactly the algorithm it is given

Before the bell – prep

Charge and count micro:bits and USB cables the day before; bookmark MakeCode (editor.makecode.com → micro:bit) on every tablet/laptop and the IWB. Test one download path yourself. If boards are short, pairs share or use the simulator only.

Materials

Item	Qty	Per	Source	Low-cost substitute
micro:bit (with USB lead)	1	pair	school devices	run the lesson on the IWB using the MakeCode simulator

Safety watch-point

Handle micro:bits and USB cables gently; keep all drinks and food away from device tables; count boards back in at the end.

**Teaching moves**

- **Name the input, process and output:** Point to buttons A/B and the LED grid on a real board or large photo. Voice the shared plan (button A → show heart → heart on LEDs), then drive a short think-pair-share so pairs only pick button and output from those options. Harvest three or four plans on the board; pairs may lightly sketch IPO on the Investigation Journal page before devices open.
- **Watch the first working program:** On the IWB, drag on button A pressed + show icon (heart), think aloud in IPO language, pause for a class prediction, then run it. Deliberately switch to button B (or strip the show block), press A so nothing useful happens, name it a bug, fix it live, and say the fix is debugging.
- **Program and debug on the micro:bit:** Leave the IPO checklist and starter block sequence on the IWB. Circulate asking 'What is your input? Which blocks are the process? What output?' When a pair is stuck, have them read the blocks aloud before you touch the keyboard. Switch any failed hardware pair to the simulator; celebrate a clear bug-fix as much as a first success.
- **Record your steps:** Same Investigation Journal page as the earlier sketch: four plain notes — input, process, output, and one real bug with how it was fixed. Offer spoken stems for strugglers. If a pair truly had no bug, ask them to plant a small error, watch it fail, fix it, and record that.
- **What the micro:bit taught us:** Take three quick pair shares, then revoice: the device does not guess; it runs the steps it is given. Spoken-only prompt: 'What was the trickiest part, and what fixed it?' Count boards and cables back in.

What it should show

On a correct button press matching the on-button block, the chosen heart or letter appears on the LED grid. Nothing useful usually means wrong button coded, missing show block, failed download, or power — re-check the block matches the button pressed and re-download. Simulator pairs should show the same IPO behaviour.

Misconceptions & interventions

- **Pupils think the micro:bit 'knows' what they meant, so a near-right program should still work.** — Point at the deliberate demo bug: wrong button or missing show block. Stress it runs exactly the steps given — change one block, re-run, and watch the output change.
- **Pupils treat a first-run failure as 'I'm bad at coding' rather than normal debugging work.** — Name the stuck moment as a bug out loud and celebrate the fix as much as a clean first run. Have them read the blocks in order before you touch the keyboard.
- **Pupils blur input, process and output into one vague 'it shows a heart'.** — Point to the physical button, the blocks on screen, then the LEDs in turn while they say each part in plain words before they download.

Differentiation

Emerging	Developing	Proficient
• Stay on the shared starter (button A → heart) with the block sequence left on the IWB; use spoken stems for the Journal notes. • Pair at the teacher table or switch straight to the MakeCode simulator if hardware confuses them.	• Choose their own button and icon/letter, still as clear IPO, and record one real bug-fix on the Journal page.	• Add a second button with a different output, still framed as precise IPO, and note both plans plus one debug on the Journal page.

Cross-curricular hook

Link to English instructional writing — the Journal algorithm must be clear enough that another pair could follow it exactly.