

LESSON 9

Microbit Showcase

60 minutes (2 periods)

Project Application and Showcase

WHAT THIS LESSON DOES

Get ready to create an exciting project with your Microbit! Plan your ideas, code using the skills you've learned, debug any issues, and finalise your work with a neat description. Showcase your creativity step by step.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop a comprehensive plan for a Microbit project, including sketching ideas and outlining necessary steps.
- Apply coding skills to create and test a Microbit project, ensuring functionality and desired outcomes.
- Identify and rectify errors in the code through effective debugging techniques.
- Finalise the project by ensuring clean, organised code and providing a concise description of the project's purpose and functionality.
- Optional: Download and implement the project on a Microbit device, demonstrating understanding of the MakeCode editor.

BEFORE THE BELL

- Open makecode.microbit.org on the school network and confirm students can reach it and start a new project.
- If physical micro:bits will be used, count one per student or pair, with USB cables, and check they charge or connect the day before.
- Have a simple working sample ready to show on screen (for example a forever loop that plots a single LED) so the class can see a finished shape.

Microbit Showcase

HOW THE LESSON RUNS

Planning Your Project

Coding Your Project

Debugging Your Project

Finalizing Your Project

TEACHING MOVES

- 1. Planning Your Project. Give five quiet minutes for a sketch and a short step list before anyone opens the editor. Ask two students to say aloud what their micro:bit will do and the first three steps.
- 2. Coding Your Project. Set a rule: test after every small change. Point to the sample forever-and-plot pattern only as a start, then circulate and ask what each student is testing right now.
- 3. Debugging Your Project. When something fails, have them say what they expected versus what happened. Send them back to their own earlier working blocks before adding anything new.
- 4. Finalizing Your Project. Stop coding with time left. Ask for tidy blocks and a two-sentence description of purpose and behaviour. Optional: guide Download only for pairs who already have a working simulator result.

WHAT TO WATCH FOR

- Students treat a program that runs without an error message as finished, even when the micro:bit behaviour is wrong. What to do: Ask them to state the expected behaviour first, then match it against what the simulator or device actually does.
- Students pile on new blocks when stuck instead of isolating the broken part. What to do: Have them disable or remove the newest change and re-test until the last working version is clear, then fix one thing at a time.
- Students assume the micro:bit remembers where it is or what happened last run without code for it. What to do: Prompt them to point to the blocks that set starting state, and add them if they are missing.

DIFFERENTIATION

- Emerging: Sit with them on the plan only: one clear behaviour and three steps before any code. Pair them with a neighbour whose simulator already shows a working fragment and have them rebuild that fragment first.
- Developing: Ask them to test after each block change and say one thing that improved. Have them read their own code aloud to you and mark the spot where behaviour diverges from the plan.
- Proficient: Challenge them to tidy names, remove unused blocks, and tighten the project description so another student could use it. If devices are out, have them download, demo on the micro:bit, and note one difference between simulator and hardware.