LESSON 6

Microbit Compass and Thermometer

60 minutes (2 periods) Intermediate Projects

WHAT THIS LESSON DOES

Follow this guide to program your Microbit to function as a compass and thermometer. You'll create a new project, set up variables for direction, code buttons to display direction and temperature, and test everything on your device.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand and utilise the compass and temperature sensor features of the Microbit.
- Develop proficiency in creating and setting variables in a Microbit project.
- Apply conditional logic to program Microbit buttons for specific functions.
- Test and debug code using the simulator before transferring to the Microbit.
- Interpret and display data from the Microbit's sensors in a user-friendly format.

BEFORE THE BELL

- Confirm makecode.com loads on the school network and that students can create a new micro:bit project.
- Count out micro:bits and USB cables (one set per student or pair) the day before and check they power on.
- Have a simple compass rose ready to project or sketch, marked 0, 90, 180 and 270 degrees, for the direction variable step.

Microbit Compass and Thermometer

HOW THE LESSON RUNS

Create a new Microbit project
Create the 'direction' variable
Set the 'direction' variable
Program the A button
Check the direction and display it
Program the B button
Test your code
Send your code to your Microbit

TEACHING MOVES

- 1. Create a new Microbit project. Open makecode.com together and start a new micro:bit project. Name it clearly so students can find it again later in the session.
- 2. Create the 'direction' variable. In Variables, choose Make a Variable and name it direction. If needed, show how a compass maps 0 to 360 degrees onto north, east, south and west.
- 3. Set the 'direction' variable. Place a forever loop and set direction to the compass heading input. Explain that forever keeps the reading live as the micro:bit turns.
- 4. Program the A button. Add on button A pressed, then an if then else block. Click the plus twice so you have three tests and a final else, ready for the four directions.
- 5. Check the direction and display it. Fill each branch: N if direction is at least 315 or under 45, E if under 135, S if under 225, otherwise W. Show N, E, S or W as a string.
- 6. Program the B button. On button B pressed, show the number from the temperature sensor. One short handler is enough beside the compass logic.
- 7. Test your code. In the simulator, drag the direction and temperature controls, then press A and B. Confirm the display matches before any download.
- 8. Send your code to your Microbit. Download to the micro:bit. Students face different ways, press A for direction and B for temperature, and compare readings with a neighbour.

WHAT TO WATCH FOR

- Students treat North like the other branches and miss that it wraps around 0 degrees, so headings near 0 never show N. Write the North edges on the board: at least 315 or less than 45. Have them test simulator headings of 10 and 350.
- Students set direction once outside a loop, so the heading never updates when they turn the device. Ask what the display would do after a single set if they then turn. Point back to forever as what keeps the variable current.
- Students read a wrong letter and assume the program failed, when one degree boundary in the if else chain is simply off. Stay in the simulator. Pick one heading, walk the chain top to bottom with them, and stop at the first true condition.

DIFFERENTIATION

- Emerging: Pair them with someone whose direction variable already updates inside forever. Give them the North condition first and only add East once N displays correctly.
- Developing: Ask them to predict the A display for two simulator headings before they click. If B works but A does not, have them read each condition aloud against their current heading.
- Proficient: Ask them to explain why North needs an or across 0 while the other branches do not. Have them compare two micro:bit temperature readings side by side and say what might cause a small gap.