**Data handling**

**Lesson 4: Data conditions & selection**

**Introduction**

In this lesson, students are introduced to how data collected by the micro:bit’s sensors can be used as a condition in programs. They explore the effect of changing the values and use this knowledge to plan, program and debug a micro:bit as a temperature warning system similar to those found in vehicles. You will ideally need physical micro:bits to complete this lesson, although if you do not have these you can still use the simulator.

**Time:** @60 minutes

**Learning objectives**

* To know that data can be used as a condition in selection
* To explore the effects of changing the value of data in programs
* To write programs that use data as a condition

**Materials needed:** printouts of *Algorithms and programs* (**slide 3**), printouts of *Debugging with sensors* (**slide 10**), printouts of *LED planner* (**slide 12)**, large sheets of paper to write algorithms on, hex files for ‘street light’, ‘street light support’, ‘temperature sensor support’ and ‘debugging with sensors’, lesson presentation, micro:bits and associated hardware.

**Lesson summary**

1. Introduction: Comparing programs and algorithms (10 minutes)
2. Tinkering with sensors (20 minutes)
3. Using temperature sensors (25 minutes)
4. Debugging with sensors (5 mins)

**Introduction: Comparing programs and algorithms (10 minutes)**

* Show students **slide 3**, and give a printout of the slide, explain that it shows the algorithm they used last lesson which shows how an automated street light could work and that the program is a representation of this algorithm using the micro:bit as an automated light.
* Ask students to discuss in pairs where the different parts of the algorithms are represented in the program and to annotate their printout to show this (a completed example of this is included on **slide 4**). After sufficient discussion, invite students to share their ideas. Ask students to explain what the terms **data**, **selection** and **repetition** mean and where they are used in the program.

**Tinkering with sensors (20 minutes)**

* Using a micro:bit with the program **microbit-street-light** running on it, explain to the students that the program shown on **slide 6** is on the device. If you do not have access to micro:bit hardware, a modified version can be completed using the MakeCode simulator. To test their programs, students should change the light level using the input on the simulator.
* Invite suggestions from students as to what will happen when the lights in the classroom are turned off. Discuss the student’s predictions and their use of logical reasoning when predicting the program’s output.
* Turn the lights off and identify that nothing has happened. Return to the program (**slide 7**) and identify that the condition is - *is light level greater than zero*. Identify that the micro:bit’s LEDs will only come on if the light level goes below zero.
* Invite suggestions from students as to how the program could be modified so that the LEDs come on when it starts getting dark. Discuss students’ ideas but do not test out.
* Explain to the students that you want them to test out their ideas to find their own solution to the problem.
* After the students have modified their programs, ask them to transfer them to their micro:bits and test them out by turning the lights out in the classroom: the micro:bit LEDs should turn on.

**Using temperature sensors (25 minutes)**

* Use the [frost indicator link](https://www.google.com/search?q=ice+warning+light+car&safe=strict&rlz=1C1GCEA_enGB852GB852&source=lnms&sa=X&ved=0ahUKEwiR6vDiv-biAhWHMBQKHUKLALoQ_AUICygA&biw=1366&bih=608&dpr=1) in the presentation to show students information on frost warning lights on cars. Invite suggestions as to why it might be useful for a driver to know that there are frosty conditions.
* Ask students to suggest how the sensor might be used (the sensor checks the temperature - if the temperature is lower than 5 degrees Celsius it displays the frost warning light - a frost symbol).
* Give out large sheets of paper and the LED planner (**slide 12**). Ask students to work with a partner to represent this as an algorithm and to identify where data, selection and repetition have been used (see example on **slide 13**).
* Explain to students that they are going to turn the algorithm into a program for the micro:bit using the MakeCode editor and give students time to write and test their programs by using the temperature slider in the simulator.
* Depending on the time of the year, it might be possible to test the program by placing the micro:bits outdoors, otherwise you could put them in a see-through container and place them in a fridge. Allow sufficient time for the required temperature to be reached, then open the fridge to observe if the frost symbol is being displayed.

**Debugging with sensors (5 mins)**

* Give out copies of **slide 10** and use the link on the image to open the MakeCode file [debugging-with-sensor](https://makecode.microbit.org/#pub:_HgbTTvTCU0J5)s. Use the temperature slider to change the temperature and ask students to observe what happens (nothing - the image in the program isn’t displayed).
* Ask students to work in pairs and annotate their copy of the program to identify where the errors are and how they could be fixed. An explanation of how this program should be debugged is contained in the *speaker notes* section of the slide.
* If you wish, use **slide 11** to review the learning outcomes of the session.

**Extension ideas**

Students could research the sensors in smartphones and the apps that make use of these senses and report their findings back to the class.

**Differentiation**

**Support:** For the streetlight activity**,** students could use **micro-street-light-support hex file** which provides all the blocks needed for the tinkering activity and on-screen instructions.   
For the temperature sensor activity, students could use the [**temperature-sensor-support hex file**](https://makecode.microbit.org/#pub:_CHcR4WV0cDca)**.**

**Stretch & challenge:** Students could annotate their programs (right-click on the block and select ‘add comment’) to show where repetition, selection and data have been used in their programs.

**Opportunities for assessment**

* Informal assessment of students’ understanding of how data can be used in programs through tinkering.
* More formal assessment of students’ algorithms and programs.