

Year 6 – Programming B - Sensing

Unit introduction

This unit is the final KS2 programming unit and brings together elements of all the four programming constructs: sequence from Year 3, repetition from Year 4, selection from Year 5, and variables (introduced in Year 6 – ‘Programming A’). It offers learners the opportunity to use all of these constructs in a different, but still familiar environment, while also utilising a physical device — the micro:bit. The unit begins with a simple program for learners to build in and test in the programming environment, before transferring it to their micro:bit. Learners then take on three new projects in Lessons 2, 3, and 4, with each lesson adding more depth.

Design features prominently in this unit. A design template is introduced in Lesson 3, initially scaffolded to give learners the opportunity to create code from a given design. In Lesson 4 that scaffolding is gradually reduced, then in Lesson 5, learners create their own design, using the same template. In the final lesson, learners will apply their knowledge of the programming constructs and use their design to create their own micro:bit-based step counter.

There are two Year 6 programming units:

- Programming A – Variables in games
- Programming B – Sensing

This is unit B, which should be delivered after unit A.

Lesson	Brief overview	Learning objectives and Success Criteria
The micro:bit	In this lesson, learners will be introduced to the micro:bit as an input, process, output device that can be programmed. Learners will familiarise themselves with the device itself and the programming environment, before creating their own programs. They will then flash their programs to the device. https://teachcomputing.org/curriculum/key-stage-2/programming-b-sensing/lesson-1-the-micro-bit	To create a program to run on a controllable device <ul style="list-style-type: none"> • I can apply my knowledge of programming to a new environment • I can test my program on an emulator • I can transfer my program to a controllable device
Go with the flow	In this lesson, learners will explore how if, then, else statements are used to direct the flow of a program. They will initially relate if, then, else statements to real-world situations, before creating programs in MakeCode. They will apply their knowledge of if, then, else statements to create a program that features selection influenced by a random number to create a micro:bit fortune teller project. https://teachcomputing.org/curriculum/key-stage-2/programming-b-sensing/lesson-2-go-with-the-flow	To explain that selection can control the flow of a program <ul style="list-style-type: none"> • I can identify examples of conditions in the real world • I can use a variable in an if, then, else statement to select the flow of a program • I can determine the flow of a program using selection
Sensing inputs	In this lesson, learners will initially use the buttons to change the value of a variable using selection. They will then develop their programs to update the variable by moving their micro:bit using the accelerometer to sense motion. Finally, they will learn that a variable can be displayed after it is updated or in response to an input. https://teachcomputing.org/curriculum/key-stage-2/programming-b-sensing/lesson-3-sensing-inputs	To update a variable with a user input <ul style="list-style-type: none"> • I can use a condition to change a variable • I can experiment with different physical inputs • I can explain that if you read a variable, the value remains
Finding your way	In this lesson, learners will initially work at code level by applying their knowledge from the previous lesson to make their micro:bit perform the function of a compass. They will then design a program which will enable the micro:bit to be used as a navigational device. To code this, they will adapt the code they completed to make the compass. https://teachcomputing.org/curriculum/key-stage-2/programming-b-sensing/lesson-4-finding-your-way	To use an conditional statement to compare a variable to a value <ul style="list-style-type: none"> • I can explain the importance of the order of conditions in else, if statements • I can use an operand (e.g. <=>) in an if, then statement • I can modify a program to achieve a different outcome
Designing a step counter	In this lesson, learners will be working at the design level. They will pick out features of a step counter, a piece of technology with which they are likely to be familiar. They will then relate those features to the sensors on a micro:bit. Having seen a simulated example of a micro:bit step counter, learners will pick out features which they will be able to include in their design. In the main activity, learners will design the algorithm for their step counter project. Finally, they will connect the battery pack to their micro:bit to set it up as a portable device. https://teachcomputing.org/curriculum/key-stage-2/programming-b-sensing/lesson-5-designing-a-step-counter	To design a project that uses inputs and outputs on a controllable device <ul style="list-style-type: none"> • I can decide what variables to include in a project • I can design the algorithm for my project • I can design the program flow for my project
Making a step counter	In this lesson, learners will use the design that they have created in Lesson 5 to make a micro:bit-based step counter. First they will review their plans, followed by creating their code. Depending on their level of confidence, they can use a scaffolded or part-complete project, otherwise they can	To develop a program to use inputs and outputs on a controllable device <ul style="list-style-type: none"> • I can create a program based on my design • I can test my program against my design

	<p>start a new project. Learners will test and debug their code, using the emulator and then the physical device. To successfully complete this project, learners will need to use all four programming constructs: sequence, repetition, selection, and variables.</p> <p>https://teachcomputing.org/curriculum/key-stage-2/programming-b-sensing/lesson-6-making-a-step-counter</p>	<ul style="list-style-type: none">• I can use a range of approaches to find and fix bugs
--	--	--

Subject knowledge

This unit focuses on developing pupils' understanding of variables in a different programming environment and using a physical device. It also enables pupils to combine their knowledge and understanding of programming constructs introduced in previous years. This unit continues to advance pupils' understanding of design in programming, using the approach outlined below.

When programming, there are four levels that can help describe a project (known as 'levels of abstraction'). Research suggests that this structure can support pupils in understanding how to create a program and how it works:

- Task — what is needed
- Design — what it should do
- Code — how it is done
- Running the code — what it does

Spending time at the 'task' and 'design' levels before engaging in writing code can aid pupils in assessing the 'do-ability' of their programs. It also reduces the cognitive load for pupils during programming.

Pupils will move between the different levels throughout the unit, and this is highlighted within each lesson plan:

- Lesson 3 - pupils work at the 'code' and 'running the code' levels from a given design
- Lesson 4 - pupils move from 'design' to 'code', to 'running the code' with some scaffolding
- Lesson 5 - pupils work at the 'design' level with increasing independence
- Lesson 6 - pupils work at the 'code' and 'running the code' levels, using their own design