

YEAR 6

	EFYS	YEAR 1	YEAR 2	YEAR 3	YEAR 4	YEAR 5	YEAR 6
Programming B – Sensing Movement	<p>The three Prime ELGS of Communication and Language, PSED and Physical Development provide the foundations of which all other learning is built upon.</p> <p>No Specific ELG links.</p>	To be introduced to on-screen programming. To explore sprites and backgrounds, using programming blocks to use, modify and create programs. To be introduced to program design through algorithms.	To understand that sequences of commands have an outcome. To use and modify designs to create and evaluate a project.	To explore the links between events and actions. To move sprites in four directions. To introduce programming extensions, changing the size and colour. To design and code their own program.	To explore the concept of repetition in programming. To look at similarities and differences between count controlled and infinite loops. To modify, design and create a game using repetition, applying stages of programming design throughout.	To develop knowledge of selection by revisiting how conditions can be used in programming. To create algorithms and programs using this understanding. To design, write and evaluate a quiz program.	To use the four programming constructs (sequence, repetition, selection and variables) whilst using a physical device. To design, build and test a simple program and transfer to device.

Programming B – Sensing Movement COMPOSITES

Design, write, and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

Use sequence, selection, and repetition in programs; work with variables and various forms of input and output

Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information

COMPONENTS

	1	2	3	4	5	6	End Point
	How do I create a program to run on a controllable device?	Can I explain that selection can control the flow of a program?	How do I update a variable with a user input?	Can I use a conditional statement to compare a variable to a value?	Can I design a project that uses inputs and outputs on a controllable device?	Can I develop a program to use inputs and outputs on a controllable device?	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 pupils 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 pupils to build in and test within the new programming environment, before transferring it to their micro:bit. Pupils then take on three new projects in Lessons 2, 3, and 4, with each lesson adding more depth.
CONCEPTS Link to concept map	6 - Programming B - Sensing movement	6 - Programming B - Sensing movement	6 - Programming B - Sensing movement	6 - Programming B - Sensing movement	6 - Programming B - Sensing movement	6 - Programming B - Sensing movement	
SKILLS	Apply my knowledge of programming to a new environment Test my program on an emulator Transfer my program to a controllable device	Use a variable in an 'if, then, else' statement to select the flow of a program Determine the flow of a program using selection	Use a condition to change a variable Experiment with different physical inputs	Use a comparison operator (e.g. <=>) in an if, then statement Modify a program to achieve a different outcome	Design the algorithm for my project Design the program flow for my project	Create a program based on my design Test my program against my design Know and use a range of approaches to find and fix bugs	In the final lesson, pupils will apply their knowledge of the programming constructs and use their design to create their own micro:bit-based step counter.
KNOWLEDGE		Identify examples of conditions in the real world	Explain that checking a variable doesn't change its value	Explain the importance of the order of conditions in else, if statements	Use knowledge to decide what variables to include in a project	Know and use a range of approaches to find and fix bugs	Pupils will conclude and apply their knowledge on programming.
LESSON LINK	4 - Data and information – Spreadsheets	4 - Data and information – Spreadsheets	4 - Data and information – Spreadsheets	4 - Data and information – Spreadsheets	4 - Data and information – Spreadsheets	4 - Data and information – Spreadsheets	
PROGRESSIVE VOCABULARY	micro:bit, makecode, input, process, output, flashing, usb, trace	selection, condition, if then else, variable, random	input, selection, condition, variable, sensing, accelerometer, value	compass, direction, variable, navigation	micro:bit, design, task, algorithm, variable, step counter	plan, create, code, test, debug	Children will be able to understand, articulate and use the vocabulary
CURRICULUM EXPERIENCES					Creation of Spreadsheet		

<p>END POINT</p>	<p>Pupils will be introduced to the micro:bit as an input, process, output device that can be programmed. Pupils will familiarise themselves with the device itself and the programming environment, before creating their own programs. They will then run their programs on the device.</p>	<p>Pupils 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.</p>	<p>Pupils 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 and using the accelerometer to sense motion. Finally, they will learn that a variable's value remains the same after it has been checked by the program.</p>	<p>Pupils will apply their understanding of the importance of order in programs. They will then use comparison operators in selection to determine the flow of a program. Pupils will then modify 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 a basic compass.</p>	<p>Pupils 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. In the main activity, Pupils will design the algorithm and program flow for their step counter project.</p>	<p>Pupils 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. Pupils will test and debug their code, using the emulator and then the physical device. To successfully complete this project, Pupils will need to demonstrate their understanding of all the programming lessons they've had so far.</p>	
-------------------------	---	--	---	--	--	--	--

