

Year 2	Computer Science: Computational Thinking	Unit 1: Robot algorithms
CS2.1 I can write algorithms for everyday tasks CS2.2 I can use logical reasoning to predict the outcome of algorithms. CS2.3 I can understand decomposition is breaking objects/processes down. CS2.4 I can debug algorithms. CS2.5 I can implement simple algorithms on digital devices (Bee Bots, Apps, Daisy the Dino).		
<u>National curriculum links</u> • Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions • Create and debug simple programs • Use logical reasoning to predict the behaviour of simple programs • Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.		
1	<u>Giving instructions</u> To describe a series of instructions as a sequence • I can follow instructions given by someone else • I can choose a series of words that can be enacted as a sequence • I can give clear and unambiguous instructions In this lesson, pupils will follow instructions given to them and give instructions to others. Pupils will consider the language used to give instructions and how that language needs to be clear and precise. Pupils will combine several instructions into a sequence that can then be issued to another pupil to complete. Pupils will then consider this clear and precise set of instructions in relation to an algorithm, and they will think about how computers can only follow clear and unambiguous instructions.	
2	<u>Same but different</u> To explain what happens when we change the order of instructions • I can create different algorithms for a range of sequences (using the same commands) • I can use an algorithm to program a sequence on a floor robot • I can show the difference in outcomes between two sequences that consist of the same commands This lesson focuses on sequences, and guides pupils to consider the importance of the order of instructions within a sequence. Pupils will create several short sequences using the same commands in different orders. They will then test these sequences to see how the different orders affect the outcome.	
3	<u>Making predictions</u> To use logical reasoning to predict the outcome of a program (series of commands) • I can follow a sequence • I can predict the outcome of a sequence • I can compare my prediction to the program outcome In this lesson, pupils will use logical reasoning to make predictions. They will follow a program step by step and identify what the outcome will be.	

4	<u>Mats and routes</u> To explain that programming projects can have code and artwork ● I can explain the choices I made for my mat design ● I can identify different routes around my mat ● I can test my mat to make sure that it is usable In this lesson, pupils will design, create, and test a mat for a floor robot. This will introduce the idea that design in programming not only includes code and algorithms, but also artefacts related to the project, such as artwork and audio.
5	<u>Algorithm design</u> To design an algorithm ● I can explain what my algorithm should achieve ● I can create an algorithm to meet my goal ● I can use my algorithm to create a program In this lesson, pupils will design algorithms to move their robot around the mats that they designed in Lesson 4. As part of the design process, pupils will outline what their task is by identifying the starting and finishing points of a route. This outlining will ensure that pupils clearly understand what they want their program to achieve.
6	<u>Debugging</u> To create and debug a program that I have written ● I can plan algorithms for different parts of a task ● I can test and debug each part of the program ● I can put together the different parts of my program In this lesson, pupils will take on a larger programming task. They will break the task into chunks and create algorithms for each chunk. This process is known as 'decomposition' and is covered further in key stage 2. Pupils will also find and fix errors in their algorithms and programs. This is known as 'debugging'.