Computing Progression - Algorithms: Being able to comprehend, design, create and evaluate algorithms.

Key Stage 1	Lower Key Stage 2	Upper Key Stage 2
Across KS1 pupils will:	**In lower KS2 pupils will:**	**In upper KS2 pupils will:**
Match a command to an outcome. Predict the outcome of, and run, a command on a device. Follow an instruction. Give directions. Follow a sequence. Predict the outcome of a sequence and compare their prediction to the program outcome. Choose the order of commands in a sequence. Explain what their program should do. Plan two different programs and use them to get to the same place. Add programming blocks based on their algorithm. Use sprites that match their design. Choose a series of words that can be enacted as a sequence. Show the difference in outcomes between two sequences that consist of the same commands.	Identify and name the objects needed for a project. Implement their algorithm as code. Relate a task description to a design. Create a code snippet for a given purpose. Explain the effect of changing a value of a command. Program a computer by typing commands. Identify everyday tasks that include repetition as part of a sequence, eg brushing teeth, dance moves. Identify patterns in a sequence. Use a count-controlled loop to produce a given outcome. Explain that a computer can repeatedly call a procedure. Identify ‘chunks’ of actions in the real world. Use a procedure in a program. Choose when to use a count-controlled and/or an infinite loop. Modify loops to produce a given outcome.	Identify conditions in a program. Modify a condition in a program. Recall how conditions are used in selection. Create a program with different outcomes using selection. Identify the condition and outcomes in an 'if... then… else...' statement. Use selection in an infinite loop to check a condition. Design the flow of a program which contains ‘if… then… else…’. Explain that program flow can branch according to a condition. Show that a condition can direct program flow in one of two ways.

Use an algorithm to program a sequence on a floor robot. Use the same instructions to create different algorithms. Create an algorithm to meet a goal. Explain what their algorithm should achieve and use their algorithm to create a program. Plan algorithms for different parts of a task. Put together the different parts of a program and test and debug each part.	Recognise that some programming languages enable more than one process to be run at once.	