

Computing Progression

Concept: Programming

YR	Y1	Y2	Y3	Y4	Y5	Y6
<ul style="list-style-type: none"> -I can explain my journey through the maze. -I can talk about which directions and turns I need to take, and explain the thinking behind my decisions. -I can order or sequence movements needed to navigate through the maze. -I can break down the route into steps, helping me to learn and remember the sequence. -I can create a set of very simple instructions to make pumpkin soup. (algorithm) -I can debug designs and make improvements. - I can plan routes to get to different items. -I can use symbols in my routes. -I can consider where objects are in relation to each other within the sequence. - I can make 	<ul style="list-style-type: none"> - I can match a command to an outcome - I can predict the outcome of a command on a device - I can run a command on a device - I can follow an instruction - I can give directions - I can recall words that can be acted out - I can compare forwards and backwards movements - I can predict the outcome of a sequence involving forwards and backwards commands - I can start a sequence from the same place - I can compare left and right turns - I can experiment with turn and move commands to move a robot - I can predict the outcome of a sequence involving up to four commands 	<ul style="list-style-type: none"> - I can choose a series of words that can be enacted as a sequence - I can follow instructions given by someone else - I can give clear and unambiguous instructions - I can create different algorithms for a range of sequences (using the same commands) - I can show the difference in outcomes between two sequences that consist of the same commands - I can use an algorithm to program a sequence on a floor robot - I can compare my prediction to the program outcome - I can follow a sequence - I can predict the outcome of a sequence - I can explain the choices I made for my mat design - I can identify different routes around my mat 	<ul style="list-style-type: none"> - I can explain that objects in Scratch have attributes (linked to) - I can identify the objects in a Scratch project (sprites, backdrops) - I can recognise that commands in Scratch are represented as blocks - I can choose a word which describes an on-screen action for my plan - I can create a program following a design - I can identify that each sprite is controlled by the commands I choose - I can create a sequence of connected commands - I can explain that the objects in my project will respond exactly to the code - I can start a program in different ways - I can combine sound commands 	<ul style="list-style-type: none"> - I can create a code snippet for a given purpose - I can explain the effect of changing a value of a command - I can program a computer by typing commands - I can test my algorithm in a text-based language - I can use a template to create a design for my program - I can write an algorithm to produce a given outcome - I can identify everyday tasks that include repetition as part of a sequence, eg brushing teeth, dance moves - I can identify patterns in a sequence - I can use a count-controlled loop to produce a given outcome - I can choose which values to change in a loop - I can identify the effect of changing the number of times a task is repeated - I can predict the outcome of a program 	<ul style="list-style-type: none"> - I can create a simple circuit and connect it to a microcontroller - I can explain what an infinite loop does - I can program a microcontroller to make an LED switch on - I can connect more than one output component to a microcontroller - I can design sequences that use count-controlled loops - I can use a count-controlled loop to control outputs - I can design a conditional loop - I can explain that a condition is either true or - I can program a microcontroller to respond to an input - I can explain that a condition being met can start an action - I can identify a condition and an action in my project - I can use selection (an 'if...then...' statement) 	<ul style="list-style-type: none"> - I can explain that the way that a variable changes can be defined - I can identify examples of information that is variable - I can identify that variables can hold numbers or letters - I can explain that a variable has a name and a value - I can identify a program variable as a placeholder in memory for a single value - I can recognise that the value of a variable can be changed - I can decide where in a program to change a variable - I can make use of an event in a program to set a variable - I can recognise that the value of a variable can be used by a program - I can choose the artwork for my project - I can create algorithms for my project - I can explain my design choices



Computing Progression

<p>suggest improvements or different ways of doing things.</p>	<ul style="list-style-type: none"> - I can choose the order of commands in a sequence - I can debug my program - I can explain what my program should do - I can identify several possible solutions - I can plan two programs - I can use two different programs to get to the same place - I can compare different programming tools - I can find which commands to move a sprite - I can use commands to move a sprite - I can run my program - I can use a Start block in a program - I can use more than one block by joining them together - I can change the value - I can find blocks that have numbers - I can say what happens when I change a value - I can add blocks to each of my sprites - I can delete a sprite - I can show that a project can include more than one sprite 	<ul style="list-style-type: none"> - I can test my mat to make sure that it is usable - I can create an algorithm to meet my goal - I can explain what my algorithm should achieve - I can use my algorithm to create a program - I can plan algorithms for different parts of a task - I can put together the different parts of my program - I can test and debug each part of the program - I can identify that a program needs to be started - I can identify the start of a sequence - I can show how to run my program - I can change the outcome of a sequence of commands - I can match two sequences with the same outcome - I can predict the outcome of a sequence of commands - I can build the sequences of blocks I need 	<ul style="list-style-type: none"> - I can explain what a sequence is - I can order notes into a sequence - I can build a sequence of commands - I can decide the actions for each sprite in a program - I can make design choices for my artwork - I can identify and name the objects I will need for a project - I can implement my algorithm as code - I can relate a task description to a design - I can choose which keys to use for actions and explain my choices - I can explain the relationship between an event and an action - I can identify a way to improve a program - I can choose a character for my project - I can choose a suitable size for a character in a maze - I can program movement - I can choose blocks to set up my program - I can consider the real world when making design choices 	<ul style="list-style-type: none"> containing a count-controlled loop - I can explain that a computer can repeatedly call a procedure - I can identify 'chunks' of actions in the real world - I can use a procedure in a program - I can design a program that includes count-controlled loops - I can develop my program by debugging it - I can make use of my design to write a program - I can list an everyday task as a set of instructions including repetition - I can modify a snippet of code to create a given outcome - I can predict the outcome of a snippet of code - I can choose when to use a count-controlled and an infinite loop - I can modify loops to produce a given outcome - I can recognise that some programming languages enable more than one process to be run at once - I can choose which action will be repeated for each object 	<ul style="list-style-type: none"> to direct the flow of a program - I can create a detailed drawing of my project - I can describe what my project will do - I can identify a real-world example of a condition starting an action - I can test and debug my project - I can use selection to produce an intended outcome - I can write an algorithm that describes what my model will do - I can identify conditions in a program - I can modify a condition in a program - I can recall how conditions are used in selection - I can create a program with different outcomes using selection - I can identify the condition and outcomes in an 'if... then... else...' statement - I can use selection in an infinite loop to check a condition - I can design the flow of a program which contains 'if... then... else...' 	<ul style="list-style-type: none"> - I can choose a name that identifies the role of a variable - I can create the artwork for my project - I can test the code that I have written - I can extend my game further using more variables - I can identify ways that my game could be improved - I can share my game with others - 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 - I can determine the flow of a program using selection - 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 experiment with different physical inputs - I can explain that if you read a variable, the value remains
--	--	--	--	--	---	--



Computing Progression

	<ul style="list-style-type: none">- I can choose appropriate artwork for my project- I can create an algorithm for each sprite- I can decide how each sprite will move- I can add programming blocks based on my algorithm- I can test the programs I have created- I can use sprites that match my design	<ul style="list-style-type: none">- I can decide which blocks to use to meet the design- I can work out the actions of a sprite in an algorithm- I can choose backgrounds for the design- I can choose characters for the design- I can create a program based on the new design- I can build sequences of blocks to match my design- I can choose the images for my own design- I can create an algorithm- I can compare my project to my design- I can debug my program- I can improve my project by adding features	<ul style="list-style-type: none">- I can use a programming extension- I can build more sequences of commands to make my design work- I can choose suitable keys to turn on additional features- I can identify additional features (from a given set of blocks)- I can match a piece of code to an outcome- I can modify a program using a design- I can test a program against a given design- I can evaluate my project- I can implement my design- I can make design choices and justify them	<ul style="list-style-type: none">- I can evaluate the effectiveness of the repeated sequences used in my program- I can explain what the outcome of the repeated action should be- I can explain the effect of my changes- I can identify which parts of a loop can be changed- I can re-use existing code snippets on new sprites- I can develop my own design explaining what my project will do- I can evaluate the use of repetition in a project- I can select key parts of a given project to use in my own design- I can build a program that follows my design- I can evaluate the steps I followed when building my project- I can refine the algorithm in my design	<ul style="list-style-type: none">- I can explain that program flow can branch according to a condition- I can show that a condition can direct program flow in one of two ways- I can identify the outcome of user input in an algorithm- I can outline a given task- I can use a design format to outline my project- I can implement my algorithm to create the first section of my program- I can share my program with others- I can test my program- I can extend my program further- I can identify the setup code I need in my program- I can identify ways the program could be improved	<ul style="list-style-type: none">- I can use a condition to change a variable- I can explain the importance of the order of conditions in else, if statements- I can modify a program to achieve a different outcome- I can use an operand (e.g. <>=) in an if, then statement- 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- I can create a program based on my design- I can test my program against my design- I can use a range of approaches to find and fix bugs
--	---	--	--	--	---	--