COMPUTING – Computer Science (Discovery Education Coding)

	Year 1 (Level 1)	Year 2 (Level 2)	Year 3 (Level 3)	Year 4 (Level 4)	Year 5 (Level 5 and 6)	Year 6 (HTML and Python)
Statutory Information (NC)	**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**		**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**			
Programming	To understand that when a computer does something, it is following instructions called 'code'. To give instructions to make objects on the screen move when the program starts. To use code to make objects move when they are clicked on. To use code to write a computer program where objects in a space scene move when they are clicked on. To combine start events and click events to make a simple game. To combine start events and click events in code	To write code that makes an object move around the screen when keys are pressed. To make objects perform different actions when keys are pressed on the keyboard. To write code that makes an object change direction when different keys on the keyboard are pressed. To write code that makes an object change direction when the pointer is pressed and released. To write code where different inputs can be used to make objects move and disappear. To write code where buttons can be used to make an object move around the screen. To write the code for a simple game where buttons are used to move an object around.	To write a computer program where different pieces of code execute in a particular sequence. To create a program that uses sequences for two different objects moving on the screen. To write code that uses a timer to create a sequence of events. To use 'hit events' to program a maze game in which an object reacts to particular conditions. To use conditional hit events to control the movement of an object on the screen. To make a simple game that uses conditional hit events to check if one object has hit another. To program a simple game where conditional events are used to check whether objects have collided.	To understand how a variable can be used to keep track of the score in a game. To use variables to keep track of the score in a game that uses conditional events. To learn how to use multiple different variables and to set the value of a variable. To use a variable to keep track of the score in a game where the score increases, decreases or resets when different conditions are met. To use a loop to do something repeatedly in a program. To write code that uses nested loops. To use the concepts of loops, regular or infinite repetition, and 'if statement' blocks. To use loops, a variable and if statements to create an animated scene involving a repeating pattern.	**Level 5 objectives** To set values in code to control the speed of an object. To use object properties (speed, heading and angle) to create a driving simulation. To create a simulation game where an object's position on the screen is controlled by making changes to its co-ordinates. To write code including 'if' statements to make an object rotate, and combine this with conditional events to make a game. **Level 6 objectives (Applying the use of complex variables)** To write code that prompts the user to input the value of a variable, and use this to create an interactive block chart. To use my knowledge of variables to make a balloon pop game that gets harder as users score more points. To write the code for a shopping till using variables to store and calculate values. To create a stopwatch with stop, start, and reset buttons, and both digital and analogue displays.	**Introduction to Python** To input information in Python and generate an output by running the code. To use Python to make simple calculations and recognise symbols for multiplication (*) and division (/). To use inputs and variables in Python, including 'if' commands. To use variables in order to store values in Python. **Introduction to HTML** To get started with HTML by adding paragraphs of text to a page. To add images to a web page using HTML. To create a web page using headings, paragraphs and images.
Debugging and Reasoning	To use knowledge of coding to fix mistakes in a variety of algorithms. To predict the effect of a given instruction	To use knowledge of algorithms involving buttons and clickers to debug a given code. To explain why a sequence needs to be changed To predict and test the outcome of a given algorithm	To debug a program written block commands To use visual programming language to explain what a given program does To explain how an algorithm can solve a problem in real life context.	Test and improve given programs which involve variables. To explain simple algorithms which use loops and selection.	To debug algorithms involving speed, direction and co-ordinates. To test and improve code which features more complex variables. Use understanding of variables create more complex sequences and events.	To debug commands in Python. To debug commands in HTML. To explain how Python and HTML can be used in real life context.
Provision	Beebots/BeeBot App Dicsovery Eduction Coding	BeeBots/BeeBot App Discovery Education	Discover Education Coding	Discover Education Coding	Discover Education Coding	Discover Education Coding
Vocabulary	*code, instructions, run, up, down, left, right, direction, object, action, click event, program, decompose, stop, start*	*object, key press, control, action, algorithm, input device, event, key, run, execute, direction, clockwise, anti-clockwise, pointer, pointer press, pointer release, button, program, direction, run, control, click, debug*	*sequence, run, before, after, between, execute, algorithm, order, action, timer event, debug, condition, conditional statement, background, direction, hit event, collide, input*	*variable, score, start, click, time, alert, conditional event, value, hit event, execute, negative, collide, set, condition, repetition, loop, action, efficient, nesting, repeat, always, object, event, timer, if statement, direction*	*Level 5 - object, action, speed, property, value, accelerate, decelerate, debug, angle, speed, heading, value, iteratively, object properties, simulation, decomposition, angle, co-ordinates, condition, negative numbers, y-axis, x-axis, position, conditional event, if statement* *Level 6 - input, variable, property, background, grid, pixel, block, convert, value, alignment, unit, scale, variable, condition, event, random, loop, if statement, discount, calculate, total, percentage, Boolean, analogue, digital, loop*	**Children should be secure with vocabulary linked to block coding.* *Python - input, variable, if command, generate, output, pair programming, quotes, string* *HTML (Hyper Text Mark-up Language) – opening and closing tags, paragraph tags, angle brackets, paragraphs, headings, jpgs, graphics, PNG (portable network graphics), content*