



Computing 'Children Build Skills' Overview: Programming, Coding, Controlling

Y1	- Explore a range of control toys and devices such as sound recording devices, music players, digital recording devices - Explore outcomes when individual buttons are pressed on a programmable device - Explore an on-screen character (or BeeBot) and navigate it around a course or grid. While navigating around a course on a computer, predict what will happen once the next command is entered. - Solve simple problems by following instructions to move objects on screen or devices in the classroom - Create a series of instructions to move their peers/toys around a course using simple planning aids e.g. a series of cards used to remember and recall the order of instructions (code) - Talk about how devices need instructions to work and talk about common devices in school and in the home
Y2	- Talk about and demonstrate how everyday devices can be controlled through the use of remote control e.g. TV, DVD, cameras, projectors, automated doors, and screens etc. - Use a series of cards or written instructions to plan and/or record the sequence of instructions, understand the need for precise language e.g. forward, backward, right, left, turn, angle - Through different cross curricular opportunities create a series of instructions to program objects to move, to solve specific problems: - Understand that this is coding - Talk about what each part of the code does - Ensure that by testing, any bugs in the code are resolved (debugging) - Discuss devices that have been programmed and need code such as domestic appliances, games, apps in order to operate successfully
Y3	- Solve open ended problems with a floor robot or an on-screen program - Investigate how everyday devices are controlled using inputs and outputs e.g. automatic doors, kettle, traffic lights, microwave oven - Draw flow diagrams, (algorithms), to show how everyday devices work (movement, sound, light etc.) - Explore loops and repetitions to shorten the code - Plan, create and debug more complex sequences of instructions to achieve a specific outcome - Explore conditional If . . . statements e.g. program an object to move until it touches or hits something then stops - Begin to explore variables such as values (time, change in length, angle, count, scoring system) - Understand that outputs can be controlled using code
Y4	- Create, test, edit, save their own code enabling the onscreen object to carry out a specific task by using conditional If, repeats and loops - Build sequences of instructions to create code to solve specific problems being aware of the need for accuracy - Draw more complex patterns using repeats and loops by shortening the code to create procedures Debug my own code and the code of others - Explore changing values and using variables (time, change in length, angle, count, scoring system) - Investigate conditionals in coding by looking at examples such as, "if input 1 is off then turn on output 3" - Understand how code is used to control physical systems in the real world e.g. bar codes, cash machines, TV control and drink dispensers
Y5	- Find errors and improve given code (debug) - Create flow diagrams to explain what is happening and illustrate how control impacts on our lives - Build code to control a device or create a game which includes inputs and outputs and make use of: - Sub-procedures - Physical inputs such as a sensor and outputs - Values and/or variables - If . . . then conditional commands - Refine procedures to improve desired outcomes through the use of loops or repeats - Evaluate, test, and debug the code explaining the processes

Y6	• Undertake creative projects using procedures and variables to achieve specific goals – E.g. control a device linked to work created in Design and Technology • Build a sequence of instructions to control a device, simulation, App or game with inputs and outputs which includes: ○ Sub-procedures ○ Physical inputs and outputs ○ Values, including random numbers ○ If . . . then conditional commands ○ Variables • Explain the purpose and function of the code in the project • Compare and contrast different coding languages e.g. Flowol, Scratch, Espresso Coding recognising similarities and differences
----	---