Subject: Computing	Year: 5 – Spring 1 – **Programming A – Selection in Physical Computing**

National Curriculum objectives

- 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;
- Select, use, and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems, and content that accomplish given goals, including collecting, analysing, evaluating, and presenting data and information

Science – Electricity (Year 4)

- Construct a simple series electrical circuit, identifying and naming its basic parts, including cells, wires, bulbs, switches, and buzzers.

Design and Technology (Key stage 2)

Design:

- Generate, develop, model, and communicate their ideas through discussion, annotated sketches, cross-sectional and exploded diagrams, prototypes, pattern pieces, and computer-aided design.

Make:

- Select from and use a wider range of tools and equipment to perform practical tasks [for example, cutting, shaping, joining, and finishing], accurately;
- Select from and use a wider range of materials and components, including construction materials, textiles, and ingredients, according to their functional properties and aesthetic qualities.

Evaluate:

- Evaluate their ideas and products against their own design criteria and consider the views of others to improve their work;
- Technical knowledge;
- Understand and use electrical systems in their products [for example, series circuits incorporating switches, bulbs, buzzers, and motors];
- Apply their understanding of computing to program, monitor, and control their products.

To begin this unit, the children should have already learnt:	**The learning in this unit will prepare the children to learn these things in the future:**
Year 1 & 2 Programming is when we make a set of instructions for computers to follow. Robots, such as floor robots like Bee-bots, are one type of machine that can follow programs. We can use algorithms (a set of guidelines to perform a task) to program floor robots along routes and correct 'debug' mistakes in algorithms. *ScratchJr* is a programming application. Year 3 *ScratchJr* is a programming environment with three main areas: The Blocks Palette; Code Area; and the Stage with Sprite. *ScratchJr* can be used to create sequences using sounds.	Year 6 A variable is something that is changeable. A variable can be set and changed throughout the running of a program. Programmers will apply the Use-Modify-Create model: learners will experiment with variables in an existing project, then modify them, before they create their own project.

Year 4

Instead of typing in the code to create each individual step, we can save time by repeating a sequence of instructions. We use the ‘repeat’ function and create ‘infinite’ or ‘count-controlled’ loops.

Key Enquiry Question

What is vital to make the LED switch on? What does an infinite loop do? How does a count-controlled loop differ from an infinite loop? Why are ‘conditions’ useful in programming? Why is infinite repetition essential? Can you suggest a real-world example of a condition starting an action? Were there any bugs in your final program? How did you/will you debug it?

The Big Idea:

Microcontrollers control real-life objects (like LEDs and motors) through the construction of programs. Conditions are a means of controlling the flow of actions in a program. The children will make use of their knowledge of repetition and conditions when introduced to the concept of selection (through the ‘if...then...’ structure) and write algorithms and programs that utilise this concept.

To achieve ARE, pupils will need to be secure in the following knowledge:

By the end of this unit, children will know:

- A condition can only be true or false;
- A count-controlled loop contains a condition;
- How to compare a count-controlled loop with a condition-controlled loop;
- A condition-controlled loop will stop when a condition is met;
- When a condition is met, a loop will complete a cycle before it stops;
- selection can be used to branch the flow of a program;
- A loop can be used to repeatedly check whether a condition has been met;
- The importance of instruction order in ‘if...then...else...’ statements (and explain it).

Vocabulary:

Programmed; algorithm; button; direction; forward; backward; robot; left; right; route; design; chunking; error; debugging (introduced in KS1).

Scratch; blocks; commands; code; sprite; stage; costume; backdrop; debugging (introduced in Y3).

Logo; codes; infinite loop; count-controlled loop (introduced in Y4).

Programming; circuit; electricity; microcontroller; LED; condition-controlled loop.

By the end of this unit, children will be able to do:

- Create a condition-controlled loop;
- Use a condition in an ‘if...then…’ statement to start an action;
- Use selection to switch the program flow in one of two ways;
- Use a condition in an ‘if...then...else…’ statement to produce given outcomes.

Useful Resources:

Online training courses

Crumble App. The *Crumble* (app) ‘Getting Started’ guide: redfernelectronics.co.uk/crumble-getting-started

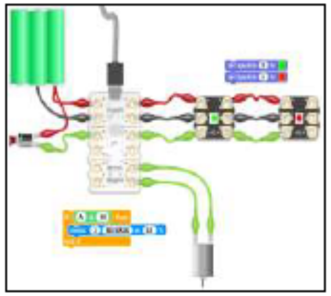
The unit has been designed to make use of the components provided in the **Crumble starter kit.**

Y5

COMPUTING: PROGRAMMING KNOWLEDGE ORGANISER

Overview

Selection in Physical Computing



-Programming is when we make and input a set of instructions for computers to follow.

-Microcontrollers are devices that can be programmed to control output devices that are connected to them.

-We use algorithms which we can plan, model, trial and debug, in order to create accurate command sequences, involving multiple output devices (e.g. LEDs and motors).

Microcontrollers, LEDs and Motors

-**Microcontrollers:** A microcontroller is a small device that can be programmed to control devices that are connected to it.

-One brand of widely used microcontroller is called a Crumble controller, which can be used to control many things, e.g. LEDs and motors.

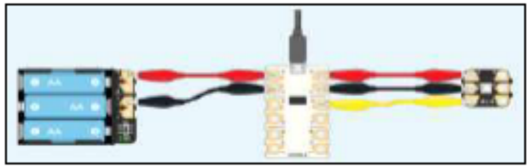
LEDs:

-LEDs are output devices that are emit light. When electricity is passed through an LED it produces light. One type of LED light, controlled by a Crumble controller, is called a Sparkle.

Motors:

-Motors are another output device. A motor can start, stop, spin forwards, spin backwards, and go at different speeds.

Creating Circuits:



-The USB port connects the microcontroller to a computer. Crocodile clips pass electricity and data through to the LED/motor.

-The + and - power pads on the Crumble should be connected with the + and - power pads on the Sparkle and battery box. The D pads on the Crumble and Sparkle should also be connected.

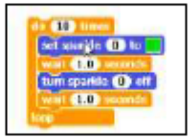
Programming Commands

-For programming, we should use the microcontroller software.

-Crumble uses command blocks (like Scratch).

-**Adding/Removing Commands:** To add a command block, drag it from the menu towards the program. When the grey arrow appears, the command will snap into the program. To remove a command block, drag it away from the program and back to the menu.

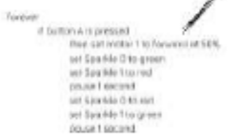
-**Modifying Commands:** Clicking on the colour square in the command block allows us to change the Sparkle's colour. To change the time of commands, click on the value. Delete the current value and type in the new value. Press enter after completed.



-**Count Controlled Loops:** These allow us to put programs on a loop. Count Controlled Loops are found in the 'Control' options. Drag the desired program into the Count Controlled Loop command block. 'Do until' loops allow commands to happen until a condition is met.

Sequencing and Algorithms

-A **sequence** is a pattern or process in which one thing follows another.



```
forever
    if button A is pressed
        then set motor 1 to forward at 50%
        set Sparkle 0 to green
        set Sparkle 1 to red
        pause 1 second
        set Sparkle 0 to red
        set Sparkle 1 to green
        pause 1 second
```

-We design **algorithms** (sets of instructions for performing a task) to help us program sequences involving multiple output devices (e.g. LEDs and motors).

-**Programming** is the process of keying in the code recognized by the computer into the software (using your algorithm).

Trialling and Debugging

-Programmers do not put their computer programs straight to work. They **trial** them first to find any errors:

-Sequence errors: An instruction in the sequence is wrong or in the wrong place.

-Keying errors: Typing in the wrong code.

-Logical errors: Mistakes in plan/thinking.

-If your algorithm does not work correctly the first time, remember to **debug** it.

Important Vocabulary

Programming · Circuit · Electricity · Microcontroller · Code · LED · Algorithm · Motor · Modify · Debugging