**Year 5 Computing Curriculum – Summer Term 2**

|  |
| --- |
| Theme: Selection in Physical Computing |
| **Curriculum objectives** | **Vocabulary** | **Links across the curriculum** |
| - 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 | **Keyword** | Definition | sequences | a pattern or process in which one thing follows another. | [**Science – Electricity (Year 4)**](https://www.gov.uk/government/publications/national-curriculum-in-england-science-programmes-of-study/national-curriculum-in-england-science-programmes-of-study)* 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)**](https://www.gov.uk/government/publications/national-curriculum-in-england-design-and-technology-programmes-of-study)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
 |
| design |  to think up and plan out in the mind | decompose | Break down into smaller and manageable ‘chunks’ |
| Count-controlled loop | A count-controlled loop is a form of repetition in which a set of commands are carried out a specific number of times. | Condition-controlled loop | A condition-controlled loop is a form of repetition in which a set of commands stop being carried out when a condition is met.  |
| debug | to fix  | conditions | Conditions are statements that need to be met for a set of actions to be carried out |
| commands | to order or instruct | program | a plan of what will be done |
| Infinite loop | An infinite loop is a loop that commands the instruction/set of instructions to repeat forever. | algorithms | a determined and finite procedure for solving a problem |
| **Prior Knowledge:**EYFS – To follow two step instructions. Year 1 – Commands for a robot. Year 2 – plan and debug algorithm Year 3 - Sequencing Sounds Year 4 – repetition in Shapes | **Future Knowledge:**Year 5 – Selection in Quizzes; Year 6 - To choose how to improve a game by using variables |
| **Lesson Sequence** | **Key Knowledge** | **Key Skills** |
| 1 Connecting Crumbles | In this lesson, your learners will become familiar with the Crumble controller and the programming environment used to control it. Learners will connect a Sparkle to a Crumble and then program the Crumble to make the Sparkle flash different colour patterns. Learners will also use infinite loops, which were introduced to the learners in the previous school year.  | To control a simple circuit connected to a computer * I can create a simple circuit and connect it to a microcontroller
* I can program a microcontroller to make an LED switch on
* I can explain what an infinite loop does
 |
| 2 Combining output components | In this lesson, learners will connect a Sparkle and a motor to the Crumble controller. Learners will design sequences of actions for these components. They will then apply their understanding of repetition by using count-controlled loops when implementing their design as a program. | To write a program that includes count-controlled loops* I can connect more than one output component to a microcontroller
* I can use a count-controlled loop to control outputs
* I can design sequences that use count-controlled loops
 |
| 3 Controlling with conditions | In this lesson, learners will be introduced to conditions, and how they can be used in programs to control their flow. They will identify conditions in statements, stating if they are true or false. Learners will be introduced to a Crumble switch, and learn how it can provide the Crumble controller with an input that can be used as a condition. They will explore how to write programs that use an input as a condition. | To explain that a loop can stop when a condition is met* I can explain that a condition is either true or false
* I can design a conditional loop
* I can program a microcontroller to respond to an input
 |
| 4 Starting with selection  | In this lesson, learners will develop their understanding of how the flow of actions in algorithms and programs can be controlled by conditions. They will be introduced to selection and then represent conditions and actions using the ‘if…then…’ structure. Learners will create algorithms that include selection. They will use their algorithms to guide their program writing. Learners will see that infinite repetition is required to repeatedly check if a condition has been met. | To explain that a loop can be used to repeatedly check whether a condition has been met* 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) to direct the flow of a program
 |
| 5 Drawing designs | In this lesson, learners will apply their understanding of microcontrollers and selection when designing a project to meet the requirements of a given task. To support their understanding, learners will identify how selection might be used in real-world situations, then they will consider how they can apply this knowledge to design their project. Learners will produce design sketches to show how their model will be made and how they will connect the microcontroller to its components. | To design a physical project that includes selection* I can identify a real-world example of a condition starting an action
* I can describe what my project will do
* I can create a detailed drawing of my project
 |
| 6 Writing and testing algorithms | In this final lesson of the unit, learners will develop Crumble programs to control the model of a fairground ride they built in Lesson 5. First, learners will identify how they are going to use selection before writing an algorithm to meet the requirements of the given task. They will then implement their algorithms as code. Learners will run their programs to identify any bugs, and then return to the code or algorithm to debug it where necessary. Finally, to conclude the unit, learners will evaluate their designs. | To create a program that controls a physical computing project* I can write an algorithm that describes what my model will do
* I can use selection to produce an intended outcome
* I can test and debug my project
 |
|  |
| **Themes and links** |
| **Computing themes** | **Where these are covered:** |
| **Technology around us** Autumn 1  | * Crumble links to the real world and computer games the children know.
 |
| **Digital painting** Autumn 2  | * Design, make and evaluate process
 |
| **Programming A** Spring 1  | * the concept of selection in programming using the Crumble programme
 |
| **Data /information** Spring 2  | * Storing the commands and the effect on language on the outcome of your commands.
 |
| **Creating media** Summer 1  | * Your own designs of Crumble
 |
| **Programming B** Summer 2  | * Using crumble to implement an algorithm as a code
 |