**Year 3 Computing Curriculum – Spring Term 1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Theme: Sequencing Sounds** | | | | | | | | |
| **Curriculum objectives** | | | **Vocabulary** | | | | | **Links across the curriculum** |
| - Design, write, and debug programmes that accomplish specific goals, including controlling or simulating physical systems; Solve problems by decomposing them into smaller parts  - Use sequence, selection, and repetition in programmes; 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 an create a range of programs, systems and content that accomplished given goals, including collecting, analysing, evaluating and presenting data and information | | | **Keyword** | Definition | sequences | a pattern or process in which one thing follows another. | | [**Computing**](https://assets.publishing.service.gov.uk/media/5a7c576be5274a1b00423213/PRIMARY_national_curriculum_-_Computing.pdf)   * 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 | |
| design | to think up and plan out in the mind | directions | guidance on where to go | |
| test | a means of finding out the nature, quality, or value of something | code | How it is done | |
| debug | to fix | plan | an action you want to take | |
| commands | to order or instruct | program | a plan of what will be done | |
| task | What is needed | 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 | | | | | **Future Knowledge:**  Year 3 - Sequencing Sounds  Year 4 – Repetition in Sounds to modify a count-controlled. Year 5 - control a simple circuit connected to a computer. Year 6 - To choose how to improve a game by using variables | | | |
| **Lesson Sequence** | | **Key Knowledge** | | | | | **Key Skills** | |
| 1. Introduction to Scratch | | This lesson introduces learners to a new programming environment: Scratch. Learners will begin by comparing Scratch to other programming environments they may have experienced, before familiarising themselves with the basic layout of the screen. | | | | | To explore a new programming environment   * I can identify the objects in a Scratch project (sprites, backdrops) * I can explain that objects in Scratch have attributes (linked to) * I can recognise that commands in Scratch are represented as blocks | |
| 1. Programming sprites | | * In this lesson, learners will create movement for more than one sprite. In doing this, they will design and implement their code, and then will create code to replicate a given outcome. Finally, they will experiment with new motion blocks. | | | | | To identify that commands have an outcome   * I can create a program following a design and understand that each sprite is controlled by the commands I choose * I can predict the coding blocks used to move a sprite * I can match coding blocks to their actions | |
| 1. Sequences | | * In this lesson, learners will be introduced to the concept of sequences by joining blocks of code together. They will also learn how event blocks can be used to start a project in a variety of different ways. In doing this, they will apply principles of design to plan and create a project. | | | | | To explain that a program has a start   * I can start a program in different ways * I can create a sequence of connected commands * I can explain that the objects in my project will respond exactly to the code | |
| 1. Ordering commands | | * This lesson explores sequences, and how they are implemented in a simple program. Learners have the opportunity to experiment with sequences where order is and is not important. They will create their own sequences from given designs. | | | | | To recognise that a sequence of commands can have an order   * I can explain what a sequence is * I can combine sound commands * I can order notes into a sequence | |
| 1. Looking good | | * This lesson develops learners’ understanding of sequences by giving them the opportunity to combine motion and sounds in one sequence. They will also learn how to use costumes to change the appearance of a sprite, and backdrops to change the appearance of the stage. They will apply the skills in Activity 1 and 2 to design and create their own project, including sequences, sprites with costumes, and multiple backdrops. | | | | | To change the appearance of my project   * 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 | |
| 1. Making an instrument | | * In this lesson, learners will create a musical instrument in Scratch. They will apply the concept of design to help develop programs and use programming blocks — which they have been introduced to throughout the unit. They will learn that code can be copied from one sprite to another, and that projects should be tested to see if they perform as expected. | | | | | To create a project from a task description   * I can identify and name the objects I will need for a project * I can relate a task description to a design * I can implement my algorithm as code | |
| **Themes and links** | | | | | | | | |
| **Computing themes** | **Where these are covered:** | | | | | | | |
| **Technology around us**  Autumn 1 | * Scratch links to the real world and computer games the children know. | | | | | | | |
| **Digital painting**  Autumn 2 | * Understanding the need for coding and algorithms | | | | | | | |
| **Programming A**  Spring 1 | * Programming the Scratch | | | | | | | |
| **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 Scratch | | | | | | | |
| **Programming B**  Summer 2 | * Using Scratch to implement an algorithm as a code | | | | | | | |