**Year 4 Computing Curriculum – Spring Term 2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Theme: Repetition in Shapes** | | | | | | | | |
| **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 bracket including Internet services bracket on a range of digital devices to design and create a range of programmes, 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. | | [**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 | decompose | Break down into smaller and manageable ‘chunks’ | |
| 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  Year 3 - Sequencing Sounds | | | | | **Future Knowledge:**  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 Programming a screen turtle | | This lesson will introduce pupils to programming in Logo. Logo is a text-based programming language where pupils type commands that are then drawn on screen. Pupils will learn the basic Logo commands, and will use their knowledge of them to read and write code. | | | | | To identify that accuracy in programming is important   * I can program a computer by typing commands * I can explain the effect of changing a value of a command * I can create a code snippet for a given purpose | |
| 2 Programming letters | | In this lesson, pupils will create algorithms (a precise set of ordered instructions, which can be turned into code) for their initials. They will then implement these algorithms by writing them in Logo commands to draw the letter. They will debug their code by finding and fixing any errors that they spot. | | | | | To create a program in a text-based language   * I can use a template to draw what I want my program to do * I can write an algorithm to produce a given outcome * I can test my algorithm in a text-based language | |
| 3 Patterns and repeats | | In this lesson, pupils will first look at examples of patterns in everyday life. They will recognise where numbers, shapes, and symbols are repeated, and how many times repeats occur. They will create algorithms for drawing a square, using the same annotated diagram as in Lesson 2. They will use this algorithm to program a square the ‘long’ way, and recognise the repeated pattern within a square. Once they know the repeated pattern, they will use the repeat command within Logo to program squares the ‘short’ way. | | | | | To explain what ‘repeat’ means   * I can identify repetition in everyday tasks * I can identify patterns in a sequence * I can use a count-controlled loop to produce a given outcome | |
| 4 Using loops to create shapes | | In this lesson, pupils will work with count-controlled loops in a range of contexts. First, they will think about a real-life example, then they will move on to using count-controlled loops in regular 2D shapes. They will trace code to predict which shapes will be drawn, and they will modify existing code by changing values within the code snippet. | | | | | To modify a count-controlled loop to produce a given outcome   * I can identify the effect of changing the number of times a task is repeated * I can predict the outcome of a program containing a count-controlled loop * I can choose which values to change in a loop | |
| 5 Breaking things down | | In this lesson, pupils will focus on decomposition. They will break down everyday tasks into smaller parts and think about how code snippets can be broken down to make them easier to plan and work with. They will learn to create, name, and call procedures in Logo, which are code snippets that can be reused in their programming. | | | | | To decompose a task into small steps   * I can identify ‘chunks’ of actions in the real world * I can use a procedure in a program * I can explain that a computer can repeatedly call a procedure | |
| 6 Creating a program | | In the final lesson, pupils will apply the skills that they have learnt in this unit to create a program containing a count-controlled loop. Over the course of the lesson, they will design wrapping paper using more than one shape, which they will create with a program that uses count-controlled loops. They will begin by creating the algorithm, either as an annotated sketch, or as a sketch and algorithm, and then implement it as code. They will debug their work throughout, and evaluate their programs against the original brief. | | | | | To create a program that uses count-controlled loops to produce a given outcome   * I can design a program that includes count-controlled loops * I can make use of my design to write a program * I can develop my program by debugging it | |
| **Themes and links** | | | | | | | | |
| **Computing themes** | **Where these are covered:** | | | | | | | |
| **Technology around us**  Autumn 1 | * Logo 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 Logo | | | | | | | |
| **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 Logo | | | | | | | |
| **Programming B**  Summer 2 | * Using Logo to implement an algorithm as a code | | | | | | | |