rodo codo

Computing Skills Progression Document

EYFS & KS1 | Lower KS2 | Upper KS2

Primary Computing Skills Progression

Rodocodo follows the national computing curriculum and ensures that through using the program, pupils will develop an understanding of the core coding concepts and develop key programming skills. By the end of primary school, children will know and understand these key concepts:

Key programming skills:	Computational thinking components:
• **Sequencing:** creating a set of actions performed in the correct order to achieve something • **Debugging:** the process of correcting errors or 'bugs' in code • **Creating Loops:** writing a sequence of instructions that is repeated until a certain condition is reached • **Creating Functions:** writing a section of a program that performs a specific task than can be used multiple times • **Using Selection:** selection is how a computer program makes decisions, and that those decisions are based on conditions • **Using Variables:** variables help computers remember values that can change	• **Algorithms:** developing or following a set of step-by-step instructions to solve a problem • **Pattern Recognition:** looking for similarities among and within problems • **Decomposition:** breaking down a complex problem into smaller, more manageable parts • **Abstraction:** focusing on the important information only, ignoring irrelevant detail

This document outlines what pupils should know and be able to do by the end of each year group through progressive objectives and outcomes. These are used to support planning and the ongoing assessments of pupil's work.

Computing Skills Progression: EYFS & KS1

	Reception	Year 1	Year 2
Objective	• To understand that a sequence is a set of actions performed in the correct order to achieve something • To understand and explain what the commands do • To understand that there can be more than 1 solution to a problem • To understand that bugs are errors in code • To understand that debugging is the process of correcting bugs	• To understand and explain what each command will do • To understand that Loops are used when you want to repeat actions • To identify that using Loops create more optimal solutions • To understand that a Function is a section of a program that performs a specific task and can be used multiple times	• To explain the Run-Step-Fix method of debugging • To identify that using Loops create more optimal solutions • To use pattern recognition to identify Loops • To understand that a Function is a section of a program that performs a specific task and can be used multiple times
Outcome	• To act out given commands by adding them to the program and running the program • To use the movement command to create a basic sequence • To use both movement and rotation to navigate the environment • To create simple sequences using movement, rotation and pick-up commands • To identify the effect of changing certain commands and writing an incorrect sequence • To find more than 1 solution to a problem • To identify bugs in pre-written programs • To debug simple programs with 1 or 2 bugs	• To act out a given command • To create simple programs using a variety of commands • To debug simple programs with 1 or 2 bugs • To use Loops within the program to control the characters movement • To create programs that use ready-made Functions	• To create programs using up to 6 different commands • To debug programs using the Run-Step-Fix method • To create simple programs using Loops • To use Loops to control the characters movement and rotation • To create programs that use ready-made Functions • To debug programs which include Functions • To debug programs which include Loops

Computing Skills Progression: Lower KS2

	Year 3	Year 4
Objective	• To explain the Run-Step-Fix method of debugging • To identify that using Loops create more optimal solutions • To use pattern recognition to identify Loops • To identify that using Functions creates more optimal solutions • To understand that decomposition is the process of breaking down larger problems into smaller, more manageable parts	• To understand that decomposition is the process of breaking down larger problems into smaller, more manageable parts • To understand that Loops can be placed within other Loops, and that these are called Nested Loops • To understand that Selection is how a computer program makes decisions, and that those decisions are based on conditions
Outcome	• To use Loops to control the characters movement and rotation • To complete ready-made Functions that are not fully complete, and use them in sequences • To debug programs which include Functions • To debug programs which include Loops • To create a Function and use it in a program • To use decomposition to break down more complex problems into multiple parts e.g. writing out the main program then writing out the Function, etc. • To evaluate their coding skills using the course assessment	• To use pattern recognition to identify Loops • To use Loops in more complex programs which include Functions • To create a Function and use it in a program • To debug programs which include both Functions and Loops • To use Nested Loops within simple programs • To use Selection within simple programs • To write complex programs using Selection and Loops (basic and nested) • To write complex programs using Selection, Loops and Functions

Computing Skills Progression: Upper KS2

	Year 5	Year 6
Objective	• To understand that Selection is how a computer program makes decisions, and that those decisions are based on conditions • To understand the difference between Counting Loops and Conditional Loops • To understand that Conditional Loops repeat until a certain condition has been reached • To understand that variables help computers remember values that can change	• To understand that Selection is how a computer program makes decisions, and that those decisions are based on conditions • To understand the difference between Counting Loops and Conditional Loops • To understand that Conditional Loops repeat until a certain condition has been reached • To understand that variables help computers remember values that can change
Outcome	• To debug programs which include both Functions and Loops • To create programs using Functions and Loops • To write complex programs using Selection, Loops and Functions • To write complex programs using Conditional Loops, Functions and Selection • To use Variables within simple programs • To write complex programs that include Variables, Loops and Functions	• To write complex programs using Conditional Loops, Functions and Selection • To use the If Else command to write more complex Selection chains that instruct the character to perform different actions • To use Variables within simple programs • To write complex programs that include Variables and Loops (basic and nested) • To write complex programs that include Variables, Loops and Functions