

Year 1. Coding (Unit 1.7)

This unit builds on children's understanding that an algorithm is a precise, step by step set of instructions to complete a task (building on knowledge gained in Unit 1.4 and 1.5) and continue to develop the understanding that this is at the core of coding. In this unit, children revisit the knowledge that a program is an algorithm for a computer (as previously learnt in Unit 1.4 and 1.5) and are introduced to the term 'computer programming' and 'coding,' and begin to understand how a programmer uses code to make a computer program. This builds on from experience gained in unit 1.5 where they began to use directional arrows and numbers (rather than whole sentences) to make screen sprites move, and is developed further as children are introduced to code blocks from the 2Code programming app. They explore how these code blocks can be joined together to create commands and further joined to create algorithms, whilst being exposed to vocabulary associated with programming in 2Code such as 'objects, actions, event, commands, design view, code view, run, execute...' Pupils begin to develop skills in reading lines of code and are encouraged to make guesses of what might happen when the code is run. When running a code does not produce the desired outcome (as with the Code Monkey Activities throughout this unit), children build on prior knowledge gained in Unit 1.5 to recognise that the code must be wrong. In these instances, they try to fix their code and are reminded of the term 'debugging' and 'bugs' as introduced in units 1.4 and 1.5 previously. By the end of the unit, children will have designed and programmed their own simple program.

Vocabulary: Algorithm, Precise, Instructions, Bug, Debug, Sprite, Object, Program, Computer Programming/ Coding, Programmer/Coder, Test, Code, Run, Execute, Save, Code blocks, Action blocks, Object Blocks, Event Blocks – when clicked, Command, Click, Scene, Background, Properties, Scale, Design view, Code View

Knowledge covered:

NC	Ladders Objectives
Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.	Y1- CS1 I understand that an algorithm is a precise, step-by-step set of instructions to complete a task Y1- CS2 I know that an algorithm written for a computer is called a program
To create and debug simple code	Y1- CS3 (KPI) I can design a simple program to control the movements of a screen sprite Y1-CS4 I can say that if something does not work how it should, it is because my code is incorrect Y1-CS5 I can try and fix my code if it isn't working properly
Use logical reasoning to predict the behaviour of simple programs.	Y1 – CS6 I can make good guesses of what is going to happen in a program. For example, where the sprite might go
Use technology purposefully to create, organise, store, manipulate and retrieve digital content	Y1- CS3 (KPI) I can design a simple program to control the movements of a screen sprite

Medium Term Lesson Plan:

Lesson 1 Unplugged	L.I: I understand that computer programs work by following instructions called code *Carry out pre-assessment learning* (Follow unit plan lesson 1)	S.C: Give and follow instructions Draw symbols to represent instructions Arrange code blocks to create a set of instructions	<u>Key Question (from KWL)</u> *Prior knowledge* What is a program? (Y1-CS2) *Prior knowledge* What is an algorithm? (Y1-CS1)
Lesson 2	L.I: I can design a simple program to control the movements of a screen sprite (Y1-CS3 KPI) (Follow unit plan lesson 2)	S.C: Use object and action blocks Fix code if it isn't working properly (debug) (Y1-CS5)	<u>Key Question (from KWL)</u> *Prior knowledge* What is a program? (Y1-CS2) *Prior knowledge* What is an algorithm? (Y1-CS1) Why might a program not work correctly? (Y1-CS4) What should you do if a program does not work correctly? (Y1-CS5)
Lesson 3	L.I: I can design a simple program to control the movements of a screen sprite (Y1-CS3 KPI) (Follow unit plan - lesson 3)	S.C: Use event, object and action code blocks Make good guesses of what is going to happen in a program (Y1-CS6)	What do you think will happen in this program? (Y1-CS6) *Prior knowledge* Why might a sprite not move where you wanted it to? (Y1-CS4) *Prior knowledge* What should you do if a sprite does not move where you wanted it to? (Y1-CS5)
Lesson 4	L.I: I can design a simple program to control the movements of a screen sprite (Y1-CS3 KPI) (follow unit plan – lesson 4)	SC: Use event, object and action code blocks Notice when the code executes when a program is run (it turns orange) Debug by making good guesses of what is going to happen in a program (Y1-CS4; CS5; CS6)	<u>Key Question (from KWL)</u> Can you predict where the sprite will go? (Y1-CS6) *Prior knowledge* Why might a sprite not move where you wanted it to? (Y1-CS4) *Prior knowledge* What should you do if a sprite does not move where you wanted it to? (Y1-CS5)
Lesson 5	L.I: I can edit a scene by adding, deleting and moving objects	SC: Understand what backgrounds and objects are Use properties table to scale Use bin to delete Drag mouse to move	

Year 1 Coding (Unit 1.7) Medium Term Plan

Lesson 6	L.I: I can design a simple program to control the movements of a screen sprite (Y1-CS3 KPI) *Carry out end of unit assessment*	SC: Program the objects to reflect the plan Read and predict what will happen before code is executed (Y1-CS6) Fix code if it is not working properly (Y1-CS5)	Why might a program not work correctly? (Y1-CS4) What should you do if a program does not work correctly? (Y1-CS5) Can you predict where the sprite will go? (Y1-CS6)
----------	---	--	--