

Year 3. Coding

This unit further refines the children's planning process behind their algorithms, taking them from a simpler storyboard format used in KS1 (Coding Unit 2.1) to studying the more organised structure of flowcharts. It develops their skills in using timers (which they were introduced to in Coding Unit 2.1) to include more complex sequences of events, which requires an understanding of how to 'nest' their lines of code within one another. The children are introduced to the idea of repetition in programming, and are shown how it may be used to save re-writing many lines of code. They are given a more complex set of debugging challenges (than they encountered in Coding Unit 2.1), which incorporate the new skills they have learned (such as nesting). Their final lesson takes them further from the closed, highly structured activities of year 2 to a more open-ended challenge, where they are to plan and build their own interactive scene; an activity which encourages all students to employ the new skills they have learned in this unit.

Adaptive teaching – Recommended tried and tested practise in these lessons:

Due to the nature of Computing lessons and the common 'firefighting' required in lessons with technical issues or children who are individually confused over what may be considered small stumbling blocks, it is recommended that teachers put time and effort into reinforcing the use of school-wide learning ethos and systems (for example, the use of the 4B's – brain, board, buddy, boss) to encourage children to try to solve their problems in the right way rather than going straight to the teacher, or lessons will quickly become unmanageable. It is also recommended to follow our school's use of the Rosenshine model of mastery teaching strategy and teach lessons in a series of stages, with short, simple tasks being set, and children who finish early can be given encouragement to support and 'lifeguard' other students. This has twofold benefits; the process of teaching others cements and secures pupils' own mastery as well as ensuring the teacher isn't overwhelmed with personal requests for help. If one-to-one support is unavailable for high need children, it is a good idea to make sure these children are sat next to or near a higher ability student who may act as peer support for them, and these children are monitored carefully throughout lessons to ensure they stay on task and understand each objective given. Ensure that any hardware support that is required for individual's SEND requirements are adjusted appropriately and are tied to those students' individual log ins (e.g. Zoom tools or screen tints for students with visual impairments etc). Teachers should have a knowledge of Pupil Premium children, particularly those who have been seen to struggle in Computing lessons, and these children should be monitored alongside those with SEND during lesson time to help ensure they access lessons along with the rest of the class.

Knowledge covered:

NC	Ladders Objectives
(2.1) Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts	Y3 - CS1 - I understand that usually a problem can be broken up into smaller parts. Y3 - CS3 - I know when to test a program and when to debug it if it doesn't work. Y3 - CS6 - I can predict some of the things that may happen BEFORE I run my program. Y3 - CS7 - I can read and explain a flowchart for a computer program
(2.2) Use sequence, selection, and repetition in programs; work with variables and various forms of input and output	Y3 - CS4 - I understand that repeating instructions can save time and make a program simpler.
(2.3) Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs	Y3 - CS2 (KPI) - I can create an algorithm to make something happen successfully. I can talk about it. Y3 - CS5 - I can explain how a simple program works (you may like to use the words input and output)
(2.4) Understand computer networks including the Internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration.	
(2.5) Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	
(2.6) 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.	
(2.7) Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.	

Medium Term Lesson Plan:

Lesson 1	Using Flowcharts L.I I can review my previous coding knowledge. L.I I understand what a flowchart is and how flowcharts are used in computer programming. Y3 - CS1 Y3 - CS2 (KPI) Y3 - CS3 Y3 - CS5 Y3 - CS6 Y3 - CS7	S.C: I can read and explain a flowchart. I can use a flowchart to create a computer program. I can create a computer program that uses click events and timers.	<u>Key Questions</u> *Prior knowledge* Which of the terms in this quiz have you come across before? (Perhaps show of hands per term before starting) Where have you seen a flowchart before? Can you predict what will happen when this program is run?	Vocabulary knowledge review Children create a program to match actions shown on a flowchart, then develop it.
Lesson 2	Using Timers L.I I understand that there are different types of timers. L.I I am able to select the right type of timer for a purpose. Y3 - CS1 Y3 - CS2 (KPI) Y3 - CS3 Y3 - CS5 Y3 - CS6 Y3 - CS7	S.C: I can create a program that uses a timer-after command. I can create a program that uses a timer-every command. I understand there can be different ways to solve a problem	<u>Key Questions</u> *Prior knowledge* What is a timer? When have you used a timer before? What is the difference between an 'after' and an 'every' timer? Can you predict what will happen if we don't 'nest' this line of code?	Children use nesting to create sequences of events in 3 structured activities.
Lesson 3	Using Repeat L.I I understand how to use the repeat command. Y3 - CS1 Y3 - CS2 (KPI) Y3 - CS3 Y3 - CS4 Y3 - CS5 Y3 - CS6 Y3 - CS7	S.C: I understand how the turtle object moves. I can use the repeat command with an object. I can create a computer program that includes use of the repeat command.	<u>Key Questions</u> *Prior knowledge* What does the word 'repeat' mean? What do you think it means in programming? How many degrees are in a right angle? How many times will we need to repeat this set of commands?	Children use repetition to program a turtle to draw a square. Task is expanded to develop their programs by changing backgrounds, pen colours or turtle appearance, and writing these changes into their programs.
Lesson 4	Code, Test and Debug L.I I can use coding knowledge to create a range of programs. L.I I understand the importance of nesting. Y3 - CS1 Y3 - CS2 (KPI) Y3 - CS3 Y3 - CS4 Y3 - CS5 Y3 - CS6 Y3 - CS7	S.C: I can create computer programs using prior knowledge. I can run, test and debug my programs. I can consider nesting when debugging my programs.	<u>Key Questions</u> *Prior knowledge* What is 'nesting'? What could it mean in terms of programming? What will happen when we run this code with an instruction embedded compared to one without?	Children to progress through three structured activities designed to show how timers and nesting can be used to create a sequence of animated events. These skills will be needed for the next two lessons.
Lesson 5 & 6	Code, Test and Debug L.I I can design and create an interactive scene. Y3 - CS1 Y3 - CS2 (KPI) Y3 - CS3 Y3 - CS4 Y3 - CS5 Y3 - CS6 Y3 - CS7	S.C: I can use the properties table to set the properties of objects. I can plan their scene and algorithms before I create my program. I can confidently make several different things happen in a program.	<u>Key Questions</u> *Prior knowledge* What did you like about this scene? How could it be improved? Can you predict how show/hide will work in this program? Could you include an alert/nesting/timers to refine and develop your scene?	Children evaluate a number of interactive scenes prior to creating one of their own.