HIGH LITTLETON CHURCH OF ENGLAND PRIMARY SCHOOL
COMPUTING MEDIUM TERM PLAN TERM 5
2025-2026

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Hedgehog (Y1)	**Exploring the keyboard** Learners will familiarise themselves with a word processor and think about how they might use this application in the future. The learners will also identify and find keys, before adding text to their page by pressing keys on a keyboard.	**Adding and removing text** Learners will continue to familiarise themselves with word processors and how they can interact with the computer using a keyboard. The learners will focus on adding text and will explore more of the keys found on a keyboard. Finally, they will begin to use the Backspace key to remove text from the computer.	**Exploring the toolbar** Learners will begin to explore the different tools that can be used in word processors to change the look of the text. Learners will use the Caps Lock key to add capital letters to their writing and will begin thinking about how to use this successfully. Learners will match simple descriptions to the related keys. Finally, learners will begin exploring the different buttons available on the toolbar in more detail, and use these to change their own text.	**Making changes to text** Learners will begin to understand when it is best to change the look of their text and which tool will achieve the most appropriate outcome. The learners will begin to use their mouse cursor to select text to enable them to make more efficient changes. They will explore the different fonts available to them and change the font for their lost toy poster.	**Explaining my choices** Learners will begin to justify their use of certain tools when changing text. The learners will decide whether the changes that they have made have improved their writing and will begin to use 'Undo' to remove changes. They will begin to consolidate their ability to select text using the cursor, through double-clicking and clicking and dragging. The learners will be able to explain what tool from the toolbar they have used to change their writing.	**Pencil or keyboard?** Learners will make comparisons between using a computer for writing and writing on paper. The learners will discuss how the two methods are the same and different and think of examples to explain this. They will demonstrate making changes to writing using a computer to compare the two methods. Finally, the learners will begin to explain which they like best and think about which method would be the best method to use in different situations.	POP task
Fox (Y2)	**How music makes us feel** Learners will listen to and compare two pieces of music from *The Planets*	**Rhythms and patterns** Learners will explore rhythm. They will create patterns and use those patterns as	**How music can be used** Learners will explore how music can be used in different	**Notes and tempo** Learners will develop their understanding of music. They will use a computer to	**Creating digital music** Learners will choose an animal and create a piece of	**Reviewing and editing music** Learners will retrieve and review their work. They	POP task

	by Gustav Holst. They will then use a musical description word bank to describe how this music generates emotions, i.e. how it makes them feel.	rhythms. They will use untuned percussion instruments and computers to hear the different rhythm patterns that they create.	ways to express emotions and to trigger their imaginations. They will experiment with the pitch of notes to create their own piece of music, which they will then associate with a physical object — in this case, an animal.	create and refine musical patterns.	music using the animal as inspiration. They will think about their animal moving and create a rhythm pattern from that. Once they have defined a rhythm, they will create a musical pattern (melody) to go with it.	will spend time making improvements and then share their work with the class.	
Badger (Y3)	**Introduction to Scratch** Learners are introduced 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.	**Programming sprites** 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.	**Sequences** 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.	**Ordering commands** Learners have the opportunity to experiment with sequences where order is and is not important. They will create their own sequences from given designs.	**Looking good** Learners develop an 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.	**Making an instrument** 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.	**POP task**

Otter (Y4)	Programming a screen turtle	Programming letters	Patterns and repeats	Using loops to create shapes	Breaking things down	Creating a program	POP task
	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.	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.	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.	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.	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.	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.	
Robin	The drawing tools	Creating images	Making effective drawings	Layers and objects	Manipulating objects	Create a vector drawing	POP task

(Y5)	Learners are introduced to vector drawings and begin to understand that they are made up of simple shapes and lines. They use the main drawing tools within the Google Drawings application to create their own vector drawings. Learners discuss how vector drawings differ from paper-based drawings.	Learners begin to identify the shapes that are used to make vector drawings. They are able to explain that each element of a vector drawing is called an object. Learners create their own vector drawing by moving, resizing, rotating, and changing the colours of a selection of objects. They also learn how to duplicate the objects to save time.	Learners increase the complexity of their vector drawings and use the zoom tool to add detail to their work. They are shown how grids and resize handles can improve the consistency of their drawings. Learners also use tools to modify objects to create a new image.	Learners gain an understanding of layers and how they are used in vector drawings. They discover that each object is built on a new layer and that these layers can be moved forwards and backwards to create effective vector drawings.	Learners find out how to select and duplicate multiple objects at a single time. They develop this skill further by learning how to group multiple objects to make them easier to work with. Learners then use this knowledge to group and ungroup objects, in order to make changes to and develop their vector drawings.	Learners use the skills they have gained in this unit to create a vector drawing for a specific purpose. They reflect on the skills they have used to create the vector drawing and think about why they used the skills they did. Learners then begin to compare vector drawings to freehand paint program drawings.	
Deer (Y6)	**Introducing variables** Learners are introduced to variables. They see examples of real-world variables (score and time in a football match) before they explore them in a Scratch project. Learners then design and make their own project that includes variables. Finally, learners identify that variables are named and that they can be letters (strings) as well as numbers.	**Variables in programming** Learners understand that variables are used in programs, and that they can only hold a single value at a time. They complete an unplugged task that demonstrates the process of changing variables. Then, learners explore why it is important to name variables and apply their learning in a Scratch project in which they make, name, and update variables.	**Improving a game** Learners apply the concept of variables to enhance an existing game in Scratch. They predict the outcome of changing the same change score block in different parts of a program, then they test their predictions in Scratch. Learners also experiment with using different values in variables, and with using a variable	**Designing a game** Learners work at the 'design' level of abstraction, where they create their artwork and algorithms. Learners first design the sprites and backgrounds for their project, then they design their algorithms to create their program flow.	**Design to code** Learners implement the algorithms that they created in Lesson 4. In doing this, they identify variables in an unfamiliar project and learn the importance of naming variables. They also have the opportunity to add another variable to enhance their project.	**Improving and sharing** Learners build on the project that they created in Lesson 5. They consider how they could improve their own projects and make small changes to achieve this. Learners then have the opportunity to add a variable independently. Finally, learners evaluate each other's projects; they identify features that they liked and	**POP task**

			elsewhere in a program. Finally, they add comments to their project to explain how they have met the objectives of the lesson.			features that could be improved.	