YEAR 4

	EYFS	YEAR 1	YEAR 2	YEAR 3	YEAR 4	YEAR 5	YEAR 6
Programming – Repetition in shapes	**The three Prime ELGS of Communication and Language, PSED and Physical Development provide the foundations of which all other learning is built upon.** **No Specific ELG links.**	To introduce early programming concepts, exploring commands and algorithms.	To develop understanding of instructions in sequences, using commands in different orders to investigate how this affects the outcome. To design, test and debug algorithms.	To explore the concept of sequencing, using motion, sound and event blocks to create their own programs with sequences. To begin to apply stages of program design.	To create programs by planning, modifying and testing commands. To understand repetition and loops within programming	To explore the concept of selection in programming. To apply knowledge of repetition and conditions to the concept of selection and write algorithms and programs using this concept. To apply stages of programming design.	To explore the concept of variables in programming. To design, modify and improve their own project.

COMPOSITES

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

COMPONENTS

	1	2	3	4	5	6	End Point
	How can I identify that accuracy in programming is important?	Can I create a program in a text-based language?	Can I explain what 'repeat' means?	How can I modify a count-controlled loop to produce a given outcome?	How can I decompose a task into small steps?	Can I create a program that uses count-controlled loops to produce a given outcome?	This unit progresses children's knowledge and understanding of programming. It progresses from the sequence of commands in a program to using count-controlled loops. Children will create algorithms and

							then implement those algorithms as code.
CONCEPTS **Link to concept map**	Programming – Repetition in shapes Information Technology Computer Science	Programming – Repetition in shapes Information Technology Computer Science	Programming – Repetition in shapes Information Technology Computer Science	Programming – Repetition in shapes Information Technology Computer Science	Programming – Repetition in shapes Information Technology Computer Science	Programming – Repetition in shapes Information Technology Computer Science	Programming – Repetition in shapes Information Technology Computer Science
SKILLS	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	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	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	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	I can identify 'chunks' of actions in the real world I can use a procedure in a program	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	Children will create programs by planning, modifying, and testing commands to create shapes and patterns. They will use Logo, a text-based programming language.
KNOWLEDGE	I know what a command is. I know what a snippet is.	I know what algorithm is.	I know what a count-controlled loop is.	I know what a value is.	I can explain that a computer can repeatedly call a procedure	I can explain that a computer can repeatedly call a procedure	This unit is the first of the two programming units in Year 4, and looks at repetition and loops within programming. Children will develop their knowledge of snippet, count-controlled

							loops, values and algorithms.
LESSON LINK	Programming A – Repetition in shapes	Programming A – Repetition in shapes	Programming A – Repetition in shapes	Programming A – Repetition in shapes	Programming A – Repetition in shapes	Programming A – Repetition in shapes	Programming A – Repetition in shapes
PROGRESSIVE VOCABULARY	program turtle — an arrow or turtle image on screen that draws a line as it is programmed commands code snippet — this could be the same as a program; it can have several sets of commands in one program	algorithm — the part of the design of the program that is precise instructions to be implemented as code design debug — the process of finding and correcting errors in your code logo commands as detailed in the 'glossary' handout	pattern, repeat, repetition, count-controlled loop, algorithm, value	repeat, repetition, count-controlled loop, trace, value	repeat count-controlled loop decompose — break something down into smaller parts procedure — a named code snippet that can be run multiple times	count-controlled loop procedure — a named code snippet that can be run multiple times debug — the process of finding and correcting errors in your code program — the entire solution to the task, and an implementation of the algorithm as code	The children will be able to understand, articulate and use the vocabulary
CURRICULUM EXPERIENCES						Design their own wrapping paper using computer programming.	
END POINT	This lesson will introduce children to programming in Logo. Logo is a text-based programming language where children type commands that are then drawn on screen. Children will learn the basic Logo commands, and will use their knowledge of them to read and write code.	In this lesson, children will create algorithms (a set of instructions in the correct order) 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, children 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	In this lesson, children 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	In this lesson, children 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 this lesson, children 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	

			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.	will be drawn, and they will modify existing code by changing values within the code snippet.		programs against the original brief.	