



## Year 4 – Programming A – Repetition in shapes

### Unit introduction

Learners will create programs by planning, modifying, and testing commands to create shapes and patterns. They will use Logo, a text-based programming language.

This unit is the first of the two programming units in Year 4, and looks at repetition and loops within programming

There are two Year 4 programming units:

- Programming A – Repetition in shapes
- Programming B – Repetition in games

This is unit A, which should be delivered before unit B.

You can use either a tablet, desktop or laptop computer for this unit. Logo software should be installed or accessible online, for example:

- You can use Turtle Academy online at [turtleacademy.com/playground](https://turtleacademy.com/playground)
- You can download FMSLogo from [fmslogo.sourceforge.net](https://fmslogo.sourceforge.net)

Note: The activities will be easier to complete on a laptop or desktop computer as there is more screen area available.

### Overview of lessons

Lesson	Brief overview	Learning objectives
1 Programming a screen turtle	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	To identify that accuracy in programming is important

	screen. Pupils will learn the basic Logo commands, and will use their knowledge of them to read and write code.	<ul style="list-style-type: none"> <li>• I can program a computer by typing commands</li> <li>• I can explain the effect of changing a value of a command</li> <li>• I can create a code snippet for a given purpose</li> </ul>
2 Programming letters	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.	<p>To create a program in a text-based language</p> <ul style="list-style-type: none"> <li>• I can use a template to draw what I want my program to do</li> <li>• I can write an algorithm to produce a given outcome</li> <li>• I can test my algorithm in a text-based language</li> </ul>
3 Patterns and repeats	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 <code>repeat</code> command within Logo to program squares the 'short' way.	<p>To explain what 'repeat' means</p> <ul style="list-style-type: none"> <li>• I can identify repetition in everyday tasks</li> <li>• I can identify patterns in a sequence</li> <li>• I can use a count-controlled loop to produce a given outcome</li> </ul>
4 Using loops to create shapes	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.	<p>To modify a count-controlled loop to produce a given outcome</p> <ul style="list-style-type: none"> <li>• I can identify the effect of changing the number of times a task is repeated</li> </ul>

		<ul style="list-style-type: none"> <li>• I can predict the outcome of a program containing a count-controlled loop</li> <li>• I can choose which values to change in a loop</li> </ul>
5 Breaking things down	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.	<p>To decompose a task into small steps</p> <ul style="list-style-type: none"> <li>• I can identify ‘chunks’ of actions in the real world</li> <li>• I can use a procedure in a program</li> <li>• I can explain that a computer can repeatedly call a procedure</li> </ul>
6 Creating a program	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.	<p>To create a program that uses count-controlled loops to produce a given outcome</p> <ul style="list-style-type: none"> <li>• I can design a program that includes count-controlled loops</li> <li>• I can make use of my design to write a program</li> <li>• I can develop my program by debugging it</li> </ul>

## Progression

This unit progresses students’ knowledge and understanding of programming. It progresses from the sequence of commands in a program to using count-controlled loops. Pupils will create algorithms and then implement those algorithms as code.

## Curriculum links

### National curriculum links

- 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

Resources are updated regularly — the latest version is available at: [ncce.io/tcc](https://ncce.io/tcc).

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](https://ncce.io/ogl).