

Subject: Computing	Year: 4 – Spring 1 – <b>Programming A – Repetition in Shapes</b>
<p><b>National Curriculum objectives</b></p> <ul style="list-style-type: none"> <li>• Design, write, and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts;</li> <li>• Use sequence, selection, and repetition in programs; work with variables and various forms of input and output;</li> <li>• Use logical reasoning to explain how some simple algorithms work, and to detect and correct errors in algorithms and programs;</li> <li>• 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.</li> </ul>	
<p><b>To begin this unit, the children should have already learnt:</b></p> <p><u>Year 1 &amp; 2</u> Programming is when we make a set of instructions for computers to follow. Robots, such as floor robots like Bee-bots, are one type of machine that can follow programs. We can use algorithms (a set of guidelines to perform a task) to program floor robots along routes and correct ‘debug’ mistakes in algorithms.</p> <p><i>ScratchJr</i> is a programming application.</p> <p><u>Year 3</u> <i>ScratchJr</i> is a programming environment with three main areas: The Blocks Palette; Code Area; and the Stage with Sprite. <i>ScratchJr</i> can be used to create sequences using sounds.</p>	<p><b>The learning in this unit will prepare the children to learn these things in the future:</b></p> <p><u>Year 5</u> Microcontrollers control real-life objects (like LEDs and motors ) through the construction of programs. Conditions are a means of controlling the flow of actions in a program. The children will make use of their knowledge of repetition and conditions when introduced to the concept of selection (through the ‘if...then...’ structure) and write algorithms and programs that utilise this concept.</p> <p><u>Year 6</u> A variable is something that is changeable. A variable can be set and changed throughout the running of a program. Programmers will apply the Use-Modify-Create model: learners will experiment with variables in an existing project, then modify them, before they create their own project.</p>
<p><b>Key Enquiry Question</b> What happens if you change the value of a command in <i>Logo</i>? What do you notice about everyday tasks you have been shown? Why would a programmer use a ‘count-controlled’ loop rather than an ‘infinite’ loop? What are the ‘chunks’ involved in your procedure? What successful is your program and how could it be improved?</p>	<p><b>The Big Idea:</b> <i>Logo</i> is a text-based programming language, where we can type commands which are then drawn on the screen. Instead of typing in the code to create each individual shape, we can save time by repeating a sequence of instructions. We use the ‘repeat’ function and create ‘infinite’ or ‘count-controlled’ loops.</p>

**To achieve ARE, pupils will need to be secure in the following knowledge:**

**By the end of this unit, children will know:**

- The meaning of 'repeat';
- Everyday tasks that include repetition as part of a sequence, e.g. brushing teeth, dance moves;
- We can use a loop command in a program to repeat instructions;
- Patterns and loops in a sequence;
- In programming there are indefinite loops and count-controlled loops;
- An indefinite loop will run until the program is stopped;
- Loops can be programmed to stop after a certain number;
- When to use a loop and when not to;
- The importance of instruction order in a loop;
- Recognise that not all tools enable more than one process to be run at once.

**Vocabulary:**

Programmed; algorithm; button; direction; forward; backward; robot; left; right; route; design; chunking; error; debugging (introduced in KS1).

Scratch; blocks; commands; code; sprite; stage; costume; backdrop; debugging (introduced in Y3).

**Logo; codes; infinite loop; count-controlled loop.**

**By the end of this unit, children will be able to do:**

- list an everyday task as a set of instructions including repetition;
- Use an infinite loop to produce a given outcome;
- Use a count-controlled loop to produce a given outcome;
- Plan a program that includes appropriate loops to produce a given outcome;
- recognise tools that enable more than one process to be run at the same time (concurrency);
- Create two or more sequences that run at the same time.

**Useful Resources:**

**Online training courses**

[Raspberry Pi online training courses](#)

*Bee-bot* floor robots

*Logo* software is also available online: Turtle Academy at [turtleacademy.com/playground](https://turtleacademy.com/playground).



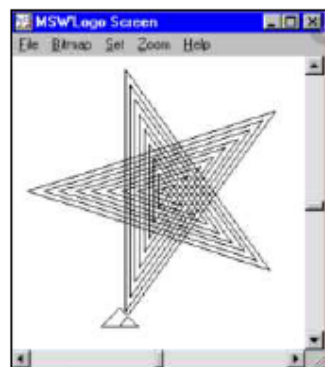
# COMPUTING: PROGRAMMING

## KNOWLEDGE ORGANISER

Y4



### Overview



#### Repetition in Shapes

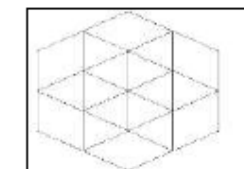
- **Programming** is when we make a set of instructions for computers to follow.
- **Logo** is a text-based program that we can use in order to create shapes and patterns.
- We use **algorithms** (a set of instructions to perform a task) which we can plan, model and test, in order to create accurate and imaginative shapes and patterns.

### Programming Patterns

- **Patterns:** Patterns are things that repeat in a logical way. In everyday life, patterns are everywhere!

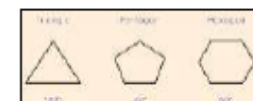


- **Patterns in Logo:** Instead of typing in the code to create each individual shape, we can save time by repeating a sequence of instructions. We use the 'repeat' function.

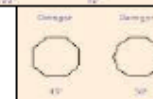


- **Repeat:** Type the command 'repeat' — this repeats commands a set number of times. The number following repeat is the number of times to repeat the code, and the code to be repeated is in square brackets, e.g. repeat 4 [FD 100 LT 90]

The above code will repeat FD 100 LT 90 four times.

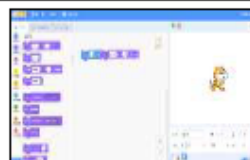


- **Creating Shapes and Loops:** To make shapes, we need to know the angles of corners of different shapes (see right). Using the repeat function with shapes can help us to make spirals.

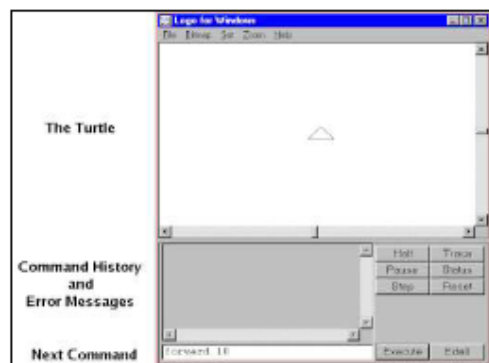


### The Basics of FMS Logo

- **What is FMS Logo?** Logo is a text-based programming language, where we can type commands which are then drawn on the screen.
- Logo helps us to learn how to use programming language, whilst also being creative and using problem-solving skills.



#### The Display:



#### Basic Commands:



- **FD:** Forwards. Always followed by a space and the number of steps, e.g. FD 50
- **BK:** Backwards. As above, e.g. BK 50
- **LT:** Left turn. Always followed by a space and then the degrees to turn, e.g. LT 90
- **RT:** Right turn. As above, e.g. RT 90
- **CS:** Clears any pen marks on your screen and gets the turtle back to the centre.
- **PU:** Stops turtle from leaving a pen trail.
- **PD:** Makes turtle leave a pen trail again.

### Sequencing and Algorithms

- A **sequence** is a pattern or process in which one thing follows another.

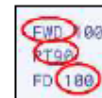
- We design **algorithms** (sets of instructions for performing a task) to help us program the sequence that we require to achieve our desired outcomes.



- **Programming** is the process of keying in the code recognized by the computer (using your algorithm).

### Trialling and Debugging

- Programmers do not put their computer programs straight to work. They **trial** them first to find any errors:



- **Sequence errors:** An instruction in the sequence is wrong or in the wrong place.

- **Keying errors:** Typing in the wrong code.

- **Logical errors:** Mistakes in plan/thinking.

- If your algorithm does not work correctly the first time, remember to **debug** it.

### Important Vocabulary

Programming

Logo

Turtle

Commands

Code

Cursor

Algorithm

Pattern

Sequence

Debugging