Seamer and Irton CP School – Computing (H.Griffiths)

Topic – Repetition in Shapes	**Year 4** **Spring 1**	**Strand – Programming**

Prior Learning	**Key Knowledge I need to understand**
In **Year 3 – Summer 2 - Events and Action** Learners explored the links between events and actions, while consolidating prior learning relating to sequencing. Learners began by moving a sprite in four directions (up, down, left, and right). They then explored movement within the context of a maze, using design to choose an appropriately sized sprite. The unit also introduced programming extensions, through the use of **Pen** blocks. Learners were given the opportunity to draw lines with sprites and change the size and colour of lines. The unit concluded with learners designing and coding their own maze-tracing program.	**I need to understand that:** **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.** Learners will create programs by planning, modifying, and testing commands to create shapes and patterns. They will use Logo, a text-based programming language.

How I will show what I have learned

To identify that accuracy in programming is important	- 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
To create a program in a text-based language	- 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
To explain what 'repeat' means	- 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
To modify 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
To decompose a task into small steps	- I can identify 'chunks' of actions in the real world - I can use a procedure in a program - I can explain that a computer can repeatedly call a procedure
To create a program that uses count-controlled loops to produce a given outcome	- 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

What vocabulary I need to know	**What's next**
Program, turtle, commands, code snippet, algorithm, design, debug, pattern, repeat, repetition, count-controlled loop, value, trace, decompose, procedure The following Glossary may be useful https://icompute-uk.com/ewExternalFiles/iCompute-Glossary.pdf	In **Year 4 – Repetition in Games – Summer 2 l**earners will explore the concept of repetition in programming using the Scratch environment. The unit begins with a Scratch activity similar to that carried out in Logo in Programming unit A, where learners can discover similarities between two environments. Learners look at the difference between count-controlled and infinite loops and use their knowledge to modify existing animations and games using repetition. Their final project is to design and create a game which uses repetition, applying stages of programming design throughout.

Please access resources at Teach Computing Curriculum - https://teachcomputing.org/curriculum

Assessment

National Curriculum Computing 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

Assessment

Formative assessment opportunities are highlighted in each of the lesson plan documents. The learning objective and success criteria will be introduced at the beginning of each lesson and then reviewed at the end. Learners should assess how well they feel they have met the learning objective using the teacher's chosen method.

Summative assessment document included - multiple choice questions. This should be used, alongside teacher judgement, to complete summative assessment on ScholarPack
https://teachcomputing.org/curriculum/key-stage-2/programming-a-repetition-in-shapes

Teacher Subject Knowledge
You can use either a tablet, desktop or laptop computer for this unit. Logo software is accessible online at Turtle Academy turtleacademy.com/playground. Pupils do not need to log in but I have set up a school log in as Username: seamer Password: turtle if required.

You will need to be able to access and demonstrate the version of Logo that you are using. You will also need to be aware of the Logo commands used in this unit. You can find these attached, and in the glossary which is part of Lesson 3 of this unit. This unit focuses on repetition, where actions or commands in programming are repeated. The repeating commands can also be placed into a loop. Loops can be repeated indefinitely, or a set number of times — the latter are called 'count-controlled loops'.
Pupils will encounter Parson's Problems, which are programming puzzles where the pupil is given the correct code, but the commands have been split and mixed up. Pupils will also carry out code tracing, where they will read through the code line by line and say exactly what each command will make happen when it runs.
In Lesson 5, pupils will look at decomposition and procedures. They will decompose code snippets, breaking them down to make them easier to plan and work with. They will use these broken down chunks to help recognise patterns in their programming.
Pupils will create and call procedures in Logo. Procedures are code snippets that are named and can be reused in their programming. When creating a procedure, the word 'TO' is typed, followed by the procedure name, eg TO SQUARE.

Note: A subroutine is a sequence of commands to perform a specific task with an identifiable name. In programming there are a number of constructs which are variations of a subroutine; these include subprogram, **procedures** and functions.

.

Glossary of Logo commands

FD — forwards. FD is always followed by a space and then a number of steps, eg FD 50

BK — backwards. BK is always followed by a space and then a number of steps, eg BK 50

LT — left. LT is always followed by a space and then a number of degrees to turn, eg LT 90

RT — right. RT is always followed by a space and then a number of degrees to turn, eg RT 90

CS — clear screen. This command clears any pen marks on your screen and gets the turtle back to the home position in the centre of the screen.

PU — pen up. This command will stop the turtle from leaving a pen trail. It is not followed by any numbers.

PD — pen down. This command will make the turtle start leaving a pen trail again, so it needs to be used before you want to draw. It is not followed by any numbers.

home — typing this command returns the turtle to the centre of your screen, pointing forwards

repeat — this command is used to repeat 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. In this example, Logo will repeat FD 100 LT 90 four times: repeat 4 [FD 100 LT 90]

setpc — setpencolour. This command changes the colour of the turtle pen. Colours are numbered 0–15 (0 is black, 1 is blue, etc), eg setpc 4