Fellside Community Primary School Computing Curriculum

Year 2 – Programming A – Robot algorithms

Unit introduction

This unit develops pupils' understanding of instructions in sequences and the use of logical reasoning to predict outcomes. Pupils will use given commands in different orders to investigate how the order affects the outcome. Pupils will also learn about design in programming. They will develop artwork and test it for use in a program. They will design algorithms and then test those algorithms as programs and debug them.

There are two Year 2 programming units:

- Programming A – Robot algorithms
- Programming B – Programming quizzes

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

Overview of lessons

Lesson	Brief overview	Learning objectives
1 Giving instructions	In this lesson, pupils will follow instructions given to them and give instructions to others. Pupils will consider the language used to give instructions and how that language needs to be clear and precise. Pupils will combine several instructions into a sequence that can then be	To describe a series of instructions as a sequence • I can follow instructions given by someone else

	issued to another pupil to complete. Pupils will then consider this clear and precise set of instructions in relation to an algorithm, and they will think about how computers can only follow clear and unambiguous instructions.	• I can choose a series of words that can be enacted as a sequence • I can give clear and unambiguous instructions
2 Same but different	This lesson focuses on sequences, and guides pupils to consider the importance of the order of instructions within a sequence. Pupils will create several short sequences using the same commands in different orders. They will then test these sequences to see how the different orders affect the outcome.	To explain what happens when we change the order of instructions • I can create different algorithms for a range of sequences (using the same commands) • I can use an algorithm to program a sequence on a floor robot • I can show the difference in outcomes between two sequences that consist of the same commands
3 Making predictions	In this lesson, pupils will use logical reasoning to make predictions. They will follow a program step by step and identify what the outcome will be.	To use logical reasoning to predict the outcome of a program (series of commands) • I can follow a sequence • I can predict the outcome of a sequence • I can compare my prediction to the program outcome
4 Mats and routes	In this lesson, pupils will design, create, and test a mat for a floor robot. This will introduce the idea that design in programming not only includes code and algorithms, but also artefacts related to the project, such as artwork and audio.	To explain that programming projects can have code and artwork • I can explain the choices I made for my mat design

		• I can identify different routes around my mat • I can test my mat to make sure that it is usable
5 Algorithm design	In this lesson, pupils will design algorithms to move their robot around the mats that they designed in Lesson 4. As part of the design process, pupils will outline what their task is by identifying the starting and finishing points of a route. This outlining will ensure that pupils clearly understand what they want their program to achieve.	To design an algorithm • I can explain what my algorithm should achieve • I can create an algorithm to meet my goal • I can use my algorithm to create a program
6 Debugging	In this lesson, pupils will take on a larger programming task. They will break the task into chunks and create algorithms for each chunk. This process is known as 'decomposition' and is covered further in key stage 2. Pupils will also find and fix errors in their algorithms and programs. This is known as 'debugging'.	To create and debug a program that I have written • I can plan algorithms for different parts of a task • I can test and debug each part of the program • I can put together the different parts of my program

Progression

In advance of the lessons in this Year 2 unit, pupils should have had some experience of creating short programs and predicting the outcome of a simple program. This unit progresses students' knowledge and understanding of algorithms and how they are implemented as programs on digital devices. Pupils

will spend time looking at how the order of commands affects outcomes. Pupils will use this knowledge and logical reasoning to trace programs and predict outcomes.

Please see the learning graph for this unit for more information about progression.

Curriculum links

National curriculum links

- Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs
- Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

Assessment

Formative assessment

Assessment opportunities are detailed in each lesson plan. The learning objective and success criteria are introduced in the slide deck at the beginning of each lesson and then reviewed at the end. Pupils are invited to assess how well they feel they have met the learning objective using thumbs up, thumbs sideways, or thumbs down.

Subject knowledge

This unit focuses on developing pupils' understanding of computer programming. It highlights that algorithms are a set of clear, precise, and ordered instructions, and that a computer program is the implementation of an algorithm on a digital device. The unit also introduces reading 'code' to predict what a program will do. Pupils will engage in aspects of program design, including outlining the project task and creating algorithms.

When programming, there are four levels that can help describe a project, known as 'levels of abstraction'. Research suggests that this structure can support pupils in understanding how to create a program and how it works:

- Task — what is needed
- Design — what it should do
- Code — how it is done
- Running the code — what it does

Spending time at the task and design levels before engaging in writing code aids pupils in assessing the achievability of their programs and reduces the cognitive load for pupils during programming.

Pupils will move between the different levels throughout the unit, and this is highlighted within each lesson plan.

Enhance your subject knowledge to teach this unit through the following training opportunities:

Online training courses

- Raspberry Pi Foundation online training courses

Face-to-face courses

- National Centre for Computing Education face-to-face training courses