Fellside Community Primary School Computing Curriculum

Year 1 – Programming A – Moving a robot

Unit introduction

This unit introduces learners to early programming concepts. Learners will explore using individual commands, both with other learners and as part of a computer program. They will identify what each floor robot command does and use that knowledge to start predicting the outcome of programs. The unit is paced to ensure time is spent on all aspects of programming and builds knowledge in a structured manner. Learners are also introduced to the early stages of program design through the introduction of algorithms.

There are two year 1 programming units:

- Programming A - Moving a robot
- Programming B - Programming animations

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

Overview of lessons

Lesson	Brief overview	Learning objectives
1. Buttons	This lesson introduces the learners to floor robots. Learners will talk about what the buttons might do and then try the buttons out. Time will be spent linking an outcome to a button press. Learners will consider the	To explain what a given command will do

	direction command buttons, as well as buttons to clear memory and run programs.	• I can predict the outcome of a command on a device • I can match a command to an outcome • I can run a command on a device
2. Directions	During this lesson, learners will think about the language used to give directions and how precise it needs to be. Learners will also work with a partner, giving and following instructions. This real-world activity should, at suitable points during this lesson, be related to the floor robot that was introduced in the last lesson.	To act out a given word • I can follow an instruction • I can recall words that can be acted out • I can give directions
3. Forwards and backwards	In this lesson, learners will focus on programming the floor robot to move forwards and backwards. They will see that the robot moves forwards and backwards a fixed distance. This highlights the idea that robots follow a clear (fixed) command in a precise and repeatable way. Learners will think about starting the robot from the same place each time. Using the same start position with fixed commands will allow learners to predict what a program will do. **Note**: This lesson focuses specifically on forwards and backwards movement only. This is to ensure that learners are developing a depth of knowledge in the concepts surrounding programming, as well as increasing their ability to make the robot move. The success criteria	To combine forwards and backwards commands to make a sequence • I can compare forwards and backwards movements • I can start a sequence from the same place • I can predict the outcome of a sequence involving forwards and backwards commands

	chosen highlight this and ensure that the learners' knowledge builds in a suitably paced way.	
4. Four directions	In this lesson, learners will use left and right turn commands along with forwards and backwards commands. Doing this will allow learners to develop slightly more complex programs. Learners will create their programs in this lesson through trial and error before moving onto planning out their programs in the next lesson. In the last activity, learners will predict where given programs will move the robot. Learners will make their predictions by 'stepping through' the commands and matching the program steps to movements.	To combine four direction commands to make sequences • I can compare left and right turns • I can experiment with turn and move commands to move a robot • I can predict the outcome of a sequence involving up to four commands
5. Getting there	In this lesson, learners will decide what their program will do. They will then create their program and test it on the robot. Where needed, learners will also debug their programs.	To plan a simple program • I can explain what my program should do • I can choose the order of commands in a sequence • I can debug my program
6. Routes	This lesson encourages learners to plan their routes before they start to write their programs. The activities also introduce the concept of there being more than one way to solve a problem. This concept applies to a lot of programming activities: the same outcome can be achieved	To find more than one solution to a problem • I can identify several possible solutions

	through a number of different approaches, and there isn't necessarily a 'right' way. The lesson also introduces the idea of program design, in which learners need to plan what they want their program to achieve before they start programming.	• I can plan two programs • I can use two different programs to get to the same place

Progression

As this is a Year 1 unit, no prior knowledge is assumed.

This unit progresses students' knowledge and understanding of giving and following instructions. It moves from giving instructions to each other to giving instructions to a robot by programming it.

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
- Recognise common uses of information technology beyond school

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. Learners 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 learners' 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. Learners 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 learners 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 code writing aids learners in assessing the achievability of their programs and reduces a learner's cognitive load during programming.

Learners 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