Year 2 – Programming A – Robot algorithms

Unit introduction

This unit develops learners' understanding of instructions in sequences and the use of logical reasoning to predict outcomes. Learners will use given commands in different orders to investigate how the order affects the outcome. They 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.

This unit includes references relating to Bee-Bot and Blue-Bot floor robots, however, other educational floor robots are available. Learners should be given access to a device with a limited range of functions that is designed for young learners. Before delivering this unit, ensure that you are familiar with your school's floor robots, including charging or battery requirements. You should also know how to switch the devices on and off, as well as key functions such as clearing the memory. It is advisable to use the robots on the floor if possible, as this can reduce damage caused by dropping.

Overview of sessions

Session	Brief overview	Learning objectives
1 Giving instructions	Learners will follow instructions given to them and give instructions to others. They will consider the language used to give instructions, and how that language needs to be clear and precise. Learners will combine several instructions into a sequence that can then be issued to another learner to complete. They will then consider a clear and precise set of instructions in relation to an algorithm, and will think about how computers can only follow clear and unambiguous instructions.	To describe a series of instructions as a sequence • I can follow instructions given by someone else • I can choose a series of words that can be acted out as a sequence • I can give clear instructions
2 Same but different	Learners will focus on sequences, and consider the importance of the order of instructions within a sequence. They will create sequences using the same instructions 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 use the same instructions to create different algorithms • 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 instructions
3 Making predictions	Learners 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 • I can follow a sequence

	Note: Learners may need to be encouraged to think through their predictions and understand that they are reasoned decisions rather than guesses.	• I can predict the outcome of a sequence • I can compare my prediction to the program outcome
4 Mats and routes	Learners 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. **Note:** The designs in this session can be changed to suit a topic or theme that the class is learning about. The ideas included in the slides are examples.	To explain that programming projects can have code and artwork • I can explain the choices that 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	Learners will design an algorithm to move their robot around the mat that they designed in Session 4. As part of the design process, learners will outline what their task is by identifying the starting and finishing points of a route. This outlining will ensure that learners 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 Break it down	Learners 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. Learners will also find and fix errors in their algorithms	To create and debug a program that I have written • I can test and debug each part of the program

	and programs. They will understand this process to be 'debugging'.	• I can plan algorithms for different parts of a task • I can put together the different parts of my program

Progression

In advance of the sessions in this Year 2 unit, learners should have had some experience of creating short programs using floor robots and predicting the outcome of a simple program. This unit progresses learners' knowledge and understanding of algorithms and how they are implemented as programs on digital devices. Learners will spend time looking at how the order of commands affects outcomes. Learners 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

Assessment

Formative assessment opportunities are provided in each of the session plan documents, and the learning objectives and success criteria can be used to observe learners' progress for summative assessment.

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 writing code aids learners in assessing the achievability of their programs and reduces the cognitive load for learners during programming.

Learners will move between the different levels throughout the unit.

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

Online training courses

If you are a teacher in England, you should access our online courses via the teachcomputing.org website:

- Get Started Teaching Computing in Primary Schools: Preparing to teach 5- to 11-year-olds
- Teaching Programming to 5- to 11-year-olds

If you are not a teacher in England, you can access our online courses via the FutureLearn platform:

- Get Started Teaching Computing in Primary Schools: Preparing to teach 5- to 11-year-olds

- Teaching Programming to 5- to 11-year-olds

Face-to-face courses

- National Centre for Computing Education face-to-face training courses (filter the courses to select from face-to-face or live remote training)

Resources are updated regularly — the latest version is available at: ncce.io/tcc.