Year 1 – Computing systems and networks – Technology around us

Unit introduction

In this unit, learners will develop their understanding of technology and how it can help us. They will start to become familiar with the different components of a computer by developing their keyboard and mouse skills. Learners will also consider how to use technology responsibly.

Overview of lessons

Lesson	Brief overview	Learning objectives
Technology around us	Technology is all around us, and we use it regularly throughout daily life. In this lesson, learners will become familiar with the term 'technology'. Referring to objects in their own school or classroom, they will learn to classify what is and what is not technology, plus they'll practice explaining how it helps us.	To identify technology • I can explain technology as something that helps us • I can locate examples of technology in the classroom • I can explain how these technology examples help us
Using technology	In this lesson, learners will get to know the main parts of a desktop or laptop computer. They will apply this knowledge to use a computer to complete a mouse-based task.	To identify a computer and its main parts • I can name the main parts of a computer

		• I can switch on and log into a computer • I can use a mouse to click and drag
Developing mouse skills	This lesson builds on the basic mouse skills introduced in lesson 2. Learners will have the opportunity to apply mouse skills to a more open-ended, creative task.	To use a mouse in different ways • I can use a mouse to open a program • I can click and drag to make objects on a screen • I can use a mouse to create a picture
Using a computer keyboard	In this lesson, learners will experience using another input device: the computer keyboard. They will combine the use of the keyboard with use of the mouse to create a digital image with text.	To use a keyboard to type • I can tell you that writing on a computer is called typing • I can type my name on a computer • I can save my work to a file
Developing keyboard skills	In this lesson, learners will become more familiar with the keyboard. They will recap skills introduced in the previous lesson and develop them further by using a greater range of keys on the keyboard.	To use the keyboard to edit text • I can open my work from a file • I can use the arrow keys to move the cursor • I can delete letters
Using a computer responsibly	In this lesson, learners will be introduced to the concept of using	To create rules for using technology

	computers safely, within the context of a school setting. They will explore why we have rules in school and how those rules help us, and then apply that to rules needed for using computer technology safely.	responsibly • I can identify rules to keep us safe and healthy when we are using technology in and beyond the home • I can give examples of some of these rules • I can discuss how we benefit from these rules

Progression

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

This unit progresses students' knowledge and understanding of technology and how they interact with it in school. Learners will build their knowledge of parts of a computer and develop the basic skills needed to effectively use a computer keyboard and mouse.

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

Curriculum links

National curriculum links

- Recognise common uses of information technology beyond school
- Use technology purposefully to create, organise, store, manipulate, and retrieve digital content
- 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.

[Education for a Connected World links]

Health, well-being and lifestyle

- I can identify rules that help keep us safe and healthy in and beyond the home when using technology
- I can give some simple examples

Copyright and ownership

- I know that the work I create belongs to me
- I can name my work so that others know it belongs to me

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 the definition of technology and offers learners the opportunity to explore computer technology by using a computer mouse and keyboard with an online paint application.

For this unit, you require a secure knowledge of how technology is defined and what is and what is not technology.

You should be familiar with:

- Paintz.app online paint application
- Functions of a computer mouse
- Functions of a computer keyboard

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

Online training courses

- Raspberry Pi online training courses

Face-to-face courses

- NCCE face-to-face training courses

Resources are updated regularly — please check that you are using the latest version.

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 direction command buttons, as well as buttons to clear memory and run programs.	To explain what a given command will do • 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	To act out a given word • I can follow an instruction • I can recall words that can be acted

	this lesson, be related to the floor robot that was introduced in the last lesson.	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 chosen highlight this and ensure that the learners' knowledge builds in a suitably paced way.	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
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 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.	To find more than one solution to a problem • I can identify several possible solutions • 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

Resources are updated regularly — please check that you are using the latest version.

Year 1 – Programming B – Programming animations

Unit introduction

This unit introduces learners to on-screen programming through ScratchJr. Learners will explore the way a project looks by investigating sprites and backgrounds. They will use programming blocks to use, modify, and create programs. Learners will also be 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 B, which should be delivered after unit A.

All the lessons in this unit require access to ScratchJr.

- Download ScratchJr App for tablets (iPad or Android), or install ScratchJr for computers (https://jfo8000.github.io/ScratchJr-Desktop/) before the lesson

Overview of lessons

Lesson	Brief overview	Learning objectives
Comparing tools	During this lesson learners will become accustomed to the ScratchJr programming environment. They will discover that they can move characters on-screen using commands, and compare ScratchJr to the Bee-Bots used in the previous unit.	To choose a command for a given purpose • I can find the commands to move a sprite • I can use commands to move a sprite • I can compare different programming tools

Joining blocks	During this lesson learners will discover that blocks can be joined together in ScratchJr. They will use a **Start** block to run their programs. They will also learn additional skills such as adding backgrounds and deleting sprites. Learners will follow given algorithms to create simple programs.	To show that a series of commands can be joined together • I can use more than one block by joining them together • I can use a **Start** block in a program • I can run my program
Make a change	During this lesson learners will discover that some blocks in ScratchJr have numbers underneath them. They will learn how to change these values and identify the effect on a block of changing a value.	To identify the effect of changing a value • I can find blocks that have numbers • I can change the value • I can say what happens when I change a value
Adding sprites	During this lesson learners will be taught how to add and delete sprites in ScratchJr. They will discover that each sprite has its own programming area, and learn how to add programming blocks to give instructions to each of the sprites.	To explain that each sprite has its own instructions • I can show that a project can include more than one sprite • I can delete a sprite • I can add blocks to each of my sprites
Project design	During this lesson learners will choose appropriate backgrounds and sprites for a ‘Space race’ project. They will decide how each sprite will move, and create an algorithm based on the blocks available in ScratchJr that reflects this.	To design the parts of a project • I can choose appropriate artwork for my project • I can decide how each sprite will move • I can create an algorithm for each sprite

Following my design	During this lesson learners will use their project designs from the previous lesson to create their projects on-screen in ScratchJr. They will use their project design, including algorithms created in the previous lesson, to make programs for each of their rocket sprites. They will test whether their algorithms are effective when their programs are run.	To use my algorithm to create a program • I can use sprites that match my design • I can add programming blocks based on my algorithm • I can test the programs I have created

Progression

This unit progresses learners' knowledge and understanding of programming and follows on from 'Programming A – Moving a robot', where children will have learned to program a floor robot using instructions.

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

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

The 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.

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

Year 2 – Computing systems and networks - Information Technology around us

Unit introduction

In this unit, learners will look at information technology at school and beyond, in settings such as shops, hospitals, and libraries. Learners will investigate how information technology improves our world, and they will learn about using information technology responsibly.

Overview of lessons

Lesson	Brief overview	Learning objectives
1 What is information technology?	This lesson develops learners' understanding of what information technology (IT) is. They will identify devices which are computers and consider how IT can help us both at school and at home.	To recognise the uses and features of information technology • I can identify examples of computers • I can describe some uses of computers • I can identify that a computer is a part of information technology
2 Where have we seen information technology	This lesson encourages learners to consider common uses of information technology in a context that they are familiar with beyond	To identify information technology in the home

in the home?	school.	• I can explain the purpose of information technology in the home • I can open a file • I can move and resize images
3 Where have we seen information technology in the world?	Having considered the use of information technology in the familiar context of the home, learners will explore IT in other environments that they may have experienced.	To identify information technology beyond school • I can find examples of information technology • I can talk about uses of information technology • I can compare types of information technology
4 How does IT improve our world?	In the previous lesson, learners looked at where IT is likely to be found and considered where it is less likely to be found. This lesson focuses on the specific use of IT in a shop.	To explain how information technology benefits us • I can demonstrate how information technology is used in a shop • I can recognise that information technology can be connected • I can explain how information technology helps people
5 Demonstrate safe use of information technology	In this lesson, learners will consider how they use different forms of information technology safely, in a range of different environments.	To show how to use information technology safely

		• I can list different uses of information technology • I can recognise how to use information technology responsibly • I can say how those rules/guides can help me
6 Using information technology responsibly	In this lesson, learners will think about the choices that are made when using information technology, and the responsibility associated with those choices.	To recognise that choices are made when using information technology • I can identify the choices that I make when using information technology • I can explain simple guidance for using information technology in different environments and settings • I can enjoy a variety of activities

Progression

Learners should have an understanding of what technology is and where it is used in a school context. They should also be familiar with the technology available in their own school setting.

This unit progresses students' knowledge and understanding of technology and how they interact with it beyond school. Learners will also build on their knowledge of using technology safely and responsibly, and begin to consider the implications of the choices that they make.

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

Curriculum links

National curriculum links

- Use technology purposefully to create, organise, store, manipulate and retrieve digital content
- Recognise common uses of information technology beyond school
- 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.

Education for a Connected World links

Health, well-being and lifestyle

- I can identify rules that help keep us safe and healthy in and beyond the home when using technology.
- I can give some simple examples.

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

Lesson 1: You should have a clear understanding of devices which can be described as information technology (IT). For younger learners, IT can be seen as computers, devices with computers inside, or things made to work with computers.

Lesson 2: You should have a clear understanding of devices which can be described as IT. For younger learners, IT can be seen as computers, devices with computers inside, or things made to work with computers.

Lesson 3: You will need to understand where technology can be found in shops and how it can be used. You should also know which devices can work together, for example:

- Barcode, barcode scanner, till
- Bank card, chip and PIN card reader, till
- Traffic light, crossing button, crossing signal

Lesson 4: You can find some useful information and a short video about barcodes at www.waspbarcode.com/buzz/barcode

Lesson 5: You should know your school's rules regarding the safe use of technology and be familiar with Education for a Connected World.

Lesson 6: You will need to be familiar with the Digital 5 a Day: www.childrenscommissioner.gov.uk/our-work/digital/5-a-day

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

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 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.	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 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	To explain what happens when we change the order of instructions

	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.	• 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	To create and debug a program that I have written

	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’.	• 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

Year 2 – Programming quizzes

Unit introduction

This unit initially recaps on learning from the Year 1 ScratchJr unit 'Programming B – Programming animations'. Learners begin to understand that sequences of commands have an outcome, and make predictions based on their learning. They use and modify designs to create their own quiz questions in ScratchJr, and realise these designs in ScratchJr using blocks of code. Finally, learners evaluate their work and make improvements to their programming projects.

There are two Year 2 programming units:

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

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

Overview of lessons

Lesson	Brief overview	Learning objectives
ScratchJr recap	During this lesson, learners will recap what they know already about the ScratchJr app. They will begin to identify the start of sequences in real-world scenarios, and learn that sequences need to be started in ScratchJr. Learners will create programs and run them in full-screen mode using the **Green flag**.	To explain that a sequence of commands has a start • I can identify the start of a sequence • I can identify that a program needs to be started • I can show how to run my program

Outcomes	During this lesson, learners will discover that a sequence of commands has an 'outcome'. They will predict the outcomes of real-life scenarios and a range of small programs in ScratchJr. Learners will then match programs that produce the same outcome when run, and use a set of blocks to create programs that produce different outcomes when run.	To explain that a sequence of commands has an outcome • I can predict the outcome of a sequence of commands • I can match two sequences with the same outcome • I can change the outcome of a sequence of commands
Using a design	During this lesson, learners will be taught how to use the **Start on tap** and **Go to page** (**Change background**) blocks. They will use a predefined design to create an animation based on the seasons. Learners will then be introduced to the task for the next lesson. They will predict what a given algorithm might mean.	To create a program using a given design • I can work out the actions of a sprite in an algorithm • I can decide which blocks to use to meet the design • I can build the sequences of blocks I need
Changing a design	During this lesson, learners will look at an existing quiz design and think about how this can be realised within the ScratchJr app. They will choose backgrounds and characters for their own quiz projects. Learners will modify a given design sheet and create their own quiz questions in ScratchJr.	To change a given design • I can choose backgrounds for the design • I can choose characters for the design • I can create a program based

		on the new design
Designing and creating a program	During this lesson, learners will create their own quiz question designs including their own choices of question, artwork, and algorithms. They will increase the number of blocks used within their sequences to create more complex programs.	To create a program using my own design • I can choose the images for my own design • I can create an algorithm • I can build sequences of blocks to match my design
Evaluating	During this lesson, learners will compare their projects to their designs. They will think about how they could improve their designs by adding additional features. They will modify their designs and implement the changes on their devices. Learners will find and correct errors in programs (debug) and discuss whether they debugged errors in their own projects.	To decide how my project can be improved • I can compare my project to my design • I can improve my project by adding features • I can debug

Progression

This unit progresses learners' knowledge and understanding of instructions in sequences and the use of logical reasoning to predict outcomes.

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

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

Year 3 – Connecting computers

Unit introduction

During this unit, learners develop their understanding of digital devices, with an initial focus on inputs, processes, and outputs. They also compare digital and non-digital devices. Following this, learners are introduced to computer networks, including devices that make up a network's infrastructure, such as wireless access points and switches. The unit concludes with learners discovering the benefits of connecting devices in a network.

Overview of lessons

Lesson	Brief overview	Learning objectives
1 How does a digital device work?	This lesson introduces the concepts of input, process, and output. These concepts are fundamental to all digital devices.	To explain how digital devices function • I can explain that digital devices accept inputs • I can explain that digital devices produce outputs • I can follow a process
2 What parts make up a digital device?	In this lesson, learners will develop their knowledge of input, process, and output and apply it to devices and parts of devices that they will be familiar with in their everyday surroundings.	To identify input and output devices • I can classify input and output devices • I can model a simple process • I can design a digital device

3 How do digital devices help us?	In this lesson, learners will apply their learning from lessons 1 and 2 by using programs in conjunction with inputs and outputs on a digital device. They will create two pieces of work with the same focus, using digital devices to create one piece of work, and non-digital tools to create the other. Learners will then compare and contrast the two approaches.	To recognise how digital devices can change the way we work • I can explain how I use digital devices for different activities • I can recognise similarities between using digital devices and non-digital tools • I can suggest differences between using digital devices and non-digital tools
4 How am I connected?	Many digital devices are now connected to other digital devices, e.g. computers through wires, tablets through WiFi, and smartphones through mobile phone networks. The benefit of connecting digital devices is that it allows information to be shared between users and systems. This lesson introduces the concept of connections and moving information between connected devices. Learners will learn to explain how and why computers are joined together to form networks.	To explain how a computer network can be used to share information • I can recognise different connections • I can explain how messages are passed through multiple connections • I can discuss why we need a network switch
5 How are computers connected?	This lesson introduces key network components, including a server and wireless access points. Learners will examine each device's functionality and look at the benefits of networking computers.	To explore how digital devices can be connected • I can recognise that a computer network is made up

		of a number of devices • I can demonstrate how information can be passed between devices • I can explain the role of a switch, server, and wireless access point in a network
6 What does our school network look like?	In this lesson, learners will further develop their understanding of computer networks. They will see examples of network infrastructure in a real-world setting and relate them to the activities in the last lesson.	To recognise the physical components of a network • I can identify how devices in a network are connected with one another • I can identify networked devices around me • I can identify the benefits of computer networks

Progression

This unit progresses students' knowledge and understanding of technology by focussing on digital and non-digital devices, and introducing the concept of computers connected together as a network. Following this unit, learners will explore the internet as a network of networks.

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

Curriculum links

National curriculum links

Computing

- Use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- Understand computer networks including the internet; how they can provide multiple services, such as the World Wide Web; and the opportunities they offer for communication and collaboration
- 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

Maths (lesson 1)

- **Number and place value:** Solve number problems and practical problems

Art (lesson 3)

- To improve their mastery of art and design techniques, including drawing, painting and sculpture with a range of materials

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.

Summative assessment

There are ten multiple choice questions in the quiz at the end of the unit. The questions are drawn from all six lessons. Please see the assessment question and answer documents for this unit.

Subject knowledge

Lesson 1: You will need an understanding of digital and non-digital devices. The key difference is that a digital device is capable of some processing, i.e. it has functions beyond being either on or off. You will also need to be familiar with the concept of input, process, output (IPO) which underpins all digital devices. There are cross-curricular links with maths for IPO which can be referenced during this lesson.

Lesson 2: You will need to be familiar with the inputs and outputs of a range of digital devices and you will need an understanding that devices can have one input which leads to several outputs (e.g. starting a video leads to outputs from the screen and the speaker) and that many inputs can lead to one output (e.g. using a mouse and keyboard to produce a document).

Lesson 3: You will need to be familiar with your chosen digital paint program, including the draw, fill, edit and undo functions.

Lesson 4: You will need a basic understanding of how information (data) flows around a computer network, and how this benefits us. You will also need to know that a network switch manages the way in which data moves around a network.

Lesson 5: This lesson requires further knowledge of a simple school network. It introduces a server, which for the purposes of this lesson is defined as a location to store files (they can, and usually do, carry out more functions than this). It also introduces wireless access points, which send and receive wireless signals from wireless devices such as tablets or laptops. There is likely to be a wireless access point in most classrooms or communal areas in school. It will be connected to the network via a physical cable.

Lesson 6: You will need to be familiar with how the main devices (network switch, server, wireless access point, router, printer/copier) on your school's network are connected with one another and where the devices are located. You may wish to discuss this with the IT technician prior to the lesson.

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

Year 3 – Desktop publishing

Unit introduction

Learners will become familiar with the terms ‘text’ and ‘images’ and understand that they can be used to communicate messages. They will use desktop publishing software and consider careful choices of font size, colour and type to edit and improve premade documents. Learners will be introduced to the terms ‘templates’, ‘orientation’, and ‘placeholders’ and begin to understand how these can support them in making their own template for a magazine front cover. They will start to add text and images to create their own pieces of work using desktop publishing software. Learners will look at a range of page layouts thinking carefully about the purpose of these and evaluate how and why desktop publishing is used in the real world.

The suggested application for this unit is Adobe Spark. To use Spark accounts are needed for learners. A guide to creating accounts is provided in lesson 2. Adobe Spark is web based and can be used on tablets, desktops and laptops.

Overview of lessons

Lesson	Brief overview	Learning objectives
1. Words and pictures	In this lesson, learners will become familiar with the terms ‘text’ and ‘images’ and understand that text and images need to be used carefully to communicate messages clearly. Learners will be able to give advantages and disadvantages of using text, images, or both text and images to communicate messages effectively.	To recognise how text and images convey information • I can explain the difference between text and images • I can recognise that text and images can communicate messages clearly • I can identify the advantages and disadvantages of using text and images
2. Can you edit it?	This lesson will build on last week’s lesson, in which we looked at using	To recognise that text and layout can be

	images and text to communicate a message effectively. In this lesson we will look at desktop publishing. Learners will think about how to make careful choices regarding font size, colour, and type in an invitation. The use of the Return, Backspace, and Shift keys will be explored and learners will be taught how to type age-appropriate punctuation marks. This will build on the typing skills learned in the Year 1 'Digital painting' unit. Learners will understand that once content has been added, it can be rearranged on the page.	edited • I can change font style, size, and colours for a given purpose • I can edit text • I can explain that text can be changed to communicate more clearly
3. Great template!	Learners will be introduced to the terms 'templates', 'orientation', and 'placeholders' within desktop publishing software. The learners will create their own magazine template, which they will add content to during the next lesson. This lesson has been designed on a laptop using Adobe Spark and this is reflected in the screenshots and videos. Teachers may decide to use the Adobe Spark app, or other software such as Canva or Microsoft Publisher.	To choose appropriate page settings • I can define the term 'page orientation' • I can recognise placeholders and say why they are important • I can create a template for a particular purpose
4. Can you add content?	In this lesson, learners will add their own content (text and images) to the magazine templates they created in lesson 3. They will copy the information for the front of their magazine from a prewritten document and paste it into the chosen place on their magazine cover. Images will be added from within the search facility in Adobe Spark. Teachers could ask learners to gather copyright-free images from http://www.pixabay.com if using a different application.	To add content to a desktop publishing publication • I can choose the best locations for my content • I can paste text and images to create a magazine cover • I can make changes to content after I've added it
5. Lay it out	In this lesson, learners will think about the different ways information can be laid out on a page. They will look at a range of page layouts such as letters and newspapers, and begin to think about the purpose of each of these.	To consider how different layouts can suit different purposes • I can identify different layouts • I can match a layout to a purpose • I can choose a suitable layout for a given purpose

6. Why desktop publishing?	In this lesson, learners will explain what desktop publishing means in their own words. They will think about how desktop publishing is used in the wider world and consider the benefits of using desktop publishing applications.	To consider the benefits of desktop publishing • I can identify the uses of desktop publishing in the real world • I can say why desktop publishing might be helpful • I can compare work made on desktop publishing to work created by hand

Progression

This unit progresses learners' knowledge and understanding of using digital devices to combine text and images building on work from the following units; Digital Writing Year 1, Digital painting Year 1, and Digital Photography Year 2.

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

Curriculum links

National curriculum links

Computing

- Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- 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

English programmes of study links

- Pupils should be taught to draft and write by: in non-narrative material, using simple organisational devices [for example, headings and subheadings]
- Evaluate and edit by assessing the effectiveness of their own and others' writing and suggesting improvements

- Proofread for spelling and punctuation errors

Education for a Connected World links

Managing online information

- I can use key phrases in search engines
- I can use search technologies effectively

Copyright and ownership

- When searching on the internet for content to use, I can explain why I need to consider who owns it and whether I have the right to reuse it
- I can demonstrate the use of search tools to find and access online content which can be reused by others

Assessment

Summative assessment rubric

Please see the assessment rubric document for this unit.

Subject knowledge

This unit focuses on desktop publishing.

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

Resources are updated regularly — please check that you are using the latest version.

Year 3 – Programming B – Events and actions in programs

Unit introduction

This unit explores the links between events and actions, while consolidating prior learning relating to sequencing. Learners begin by moving a sprite in four directions (up, down, left, and right). They then explore movement within the context of a maze, using design to choose an appropriately sized sprite. This unit also introduces programming extensions, through the use of **Pen** blocks. Learners are given the opportunity to draw lines with sprites and change the size and colour of lines. The unit concludes with learners designing and coding their own maze-tracing program.

There are two Year 3 programming units:

- Programming A – Sequencing sounds
- Programming B – Events and actions in programs

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

Overview of lessons

Lesson	Brief overview	Learning objectives
1 Moving a sprite	In this lesson, learners will investigate how characters can be moved using 'events'. They will analyse and improve an existing project, and then apply what they have learned to their own projects. They will then extend their learning to control multiple sprites in the same project.	To explain how a sprite moves in an existing project • I can explain the relationship between an event and an action • I can choose which keys to use for actions and explain my choices • I can identify a way to improve a program
2 Maze movement	In this lesson, learners will program a sprite to move in four directions: up, down,	To create a program to move a sprite in

	left, and right. They will begin by choosing a sprite and sizing it to fit in with a given background. Learners will then create the code to move the sprite in one direction before duplicating and modifying it to move in all four directions. Finally, they will consider how their project could be extended to prove that their sprite has successfully navigated a maze.	four directions • I can choose a character for my project • I can choose a suitable size for a character in a maze • I can program movement
3 Drawing lines	This lesson will introduce learners to extension blocks in Scratch using the **Pen** extension. Learners will use the pen down block to draw lines, building on the movement they created for their sprite in Lesson 2. Learners will then decide how to set up their project every time it is run.	To adapt a program to a new context • I can use a programming extension • I can consider the real world when making design choices • I can choose blocks to set up my program
4 Adding features	In this lesson, learners will be given the opportunity to use additional **Pen** blocks. They will predict the functions of new blocks and experiment with them, before designing features to add to their own projects. Finally, they will add these features to their projects and test their effectiveness.	To develop my program by adding features • I can identify additional features (from a given set of blocks) • I can choose suitable keys to turn on additional features • I can build more sequences of commands to make my design work
5 Debugging movement	This lesson explores the process of debugging, specifically looking at how to identify and fix errors in a program. Learners will review an existing project against a given design and identify bugs within it. They will then correct the errors, gaining independence as they do so. Learners will also develop their projects by considering which new setup blocks to use.	To identify and fix bugs in a program • I can test a program against a given design • I can match a piece of code to an outcome • I can modify a program using a design

6 Making a project	In this lesson, learners will design and create their own projects. Using a template (which can be blank or partially completed), learners will complete projects to move a sprite around a maze, with the option to leave a pen trail showing where the sprite has moved. Ideally, projects will include setup blocks to position the sprite at the start of the maze and clear any lines already on the screen.	To design and create a maze-based challenge • I can make design choices and justify them • I can implement my design • I can evaluate my project

Progression

This unit assumes that learners will have some prior experience of programming. The key stage 1 National Centre for Computing Education units focus on floor robots and ScratchJr, however experience of other languages or environments may also be useful. The Year 3 — Programming A unit introduces the Scratch programming environment and the concept of sequences.

See the learning graph for this unit for more information about progression.

Curriculum links

National curriculum 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
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.

Summative assessment
See the assessment questions and solutions for this unit.

We recommend the use of teacher and learner accounts in Scratch to help with assessment throughout this unit. For guidance on setting up teacher accounts, visit scratch.mit.edu/educators/faq. A teacher account enables you to manage learners' accounts and organise projects into studios. If you are unable to use teacher and learner accounts, work can be saved offline to local devices.

Subject knowledge

This unit focuses on the links between 'events' and 'actions' in programming, while also developing learners' understanding of sequencing. It highlights that events cause actions, and that the order of those actions can have an impact on the outcome of a program. This unit also further develops learners' understanding of design in programming, using the approach outlined below.

When programming, there are four levels that help to describe the stages of 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 — this is what is needed
- Design — this is what it should do
- Code — this is how it is done
- Running the code — this is what it does

Spending time at the Task and Design levels before engaging in code writing aids learners in assessing the 'do-ability' of their programs and reduces a learner's cognitive load during programming.

Learners will move between the different levels throughout the unit. 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

Year 4 – The internet

Unit introduction

During this unit learners will apply their knowledge and understanding of networks, to appreciate the internet as a network of networks which need to be kept secure. They will learn that the World Wide Web is part of the internet, and be given opportunities to explore the World Wide Web for themselves to learn about who owns content and what they can access, add, and create. Finally they will evaluate online content to decide how honest, accurate, or reliable it is, and understand the consequences of false information.

Overview of lessons

Lesson	Brief overview	Learning objectives
1. Connecting networks	Learners will explore how a network can share messages with another network to form the internet. They will consider some of the network devices involved in this, such as routers, and then discuss what we should keep in and out of a network to keep safe.	To describe how networks physically connect to other networks • I can describe the internet as a network of networks • I can demonstrate how information is shared across the internet • I can discuss why a network needs protecting
2. What is the internet made of?	Learners will describe parts of a network and how they connect to each other to form the internet. They will use this to help explain how the internet lets us view the World Wide Web and recognise that the World Wide Web is part of the internet which contains websites and web	To recognise how networked devices make up the internet • I can describe the different networked devices and how

	pages.	they connect • I can explain how the internet allows us to view the World Wide Web • I can recognise that the World Wide Web is the part of the internet that contains websites and web pages
3. Sharing information	Learners will explore what can be shared on the World Wide Web and where websites are stored. They will also explore how the World Wide Web can be accessed on a variety of devices.	To outline how websites can be shared via the World Wide Web • I can explain the types of media that can be shared on the World Wide Web (WWW) • I can describe where websites are stored when uploaded to the WWW • I can describe how to access websites on the WWW
4. What is a website?	Learners will analyse the contents of websites, before designing their own website, offline. They will consider the content they would like to include on a website of their own, and then decide how they could create that content. They will then use an existing website to create some of their own content online, using tools introduced in Year 2.	To describe how content can be added and accessed on the World Wide Web • I can create media which can be found on websites • I can recognise that I can add content to the WWW

		• I can explain that new content can be created online
5. Who owns the web?	Learners will explore who owns the content on websites. They will explore a variety of websites, investigating what they can and cannot do with the content on them. They will also relate this to principles of ownership and sharing in the real world.	To recognise how the content of the WWW is created by people • I can explain that websites and their content are created by people • I can suggest who owns the content on websites • I can explain that there are rules to protect content
6. Can I believe what I read?	In this lesson, learners will gain an appreciation of the fact that not everything they see on the internet is true, honest, or accurate. They will review images and decide they may not be real, before conducting a web search which will return ambiguous and sometimes misleading results, looking for why this is the case. Finally, learners will complete a practical activity, demonstrating how quickly information can spread, beyond your own control.	To evaluate the consequences of unreliable content • I can explain that not everything on the World Wide Web is true. • I can explain why some information I find online may not be honest, accurate, or legal. • I can explain why I need to think carefully before I share or reshare content

Progression

This unit progresses students' knowledge and understanding of networks in Year 3. In Year 5, they will continue to develop their knowledge and understanding of computing systems and online collaborative working.

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

Curriculum links

National curriculum links

Computing

- Understand computer networks including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration
- Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- 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
- Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

PSHE (Lesson 6)

- Evaluating content for honesty and accuracy

Art (Lesson 3)

- To improve their mastery of art and design techniques, including drawing, painting, and sculpture with a range of materials

Assessment

Formative assessment
Assessment opportunities are detailed in each lesson plan. The learning objectives 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.

Summative assessment
There are ten multiple choice questions in the quiz at the end of the unit. The questions are drawn from all six lessons.

- Please see the assessment question and answer documents for this unit.

Subject knowledge

Lesson 1:
Knowledge of computer networks is required for this lesson. It builds on concepts introduced in the Year 3 Computer systems and networks unit, in particular, the definition of a network which is covered in Lesson 4.

Lesson 2:
This lesson builds on Year 3, Computing systems and networks, in particular the parts of a network, covered in Lessons 4 and 5.

You will need an understanding of how data is routed around the internet. Some of the concepts covered in this lesson are explained in 'A Packet's Tale' (a YouTube video): https://www.youtube.com/watch?v=ewrBaIT_eBM

You will also need a clear understanding that the World Wide Web is part of the internet — this is explained in this video: https://www.bbc.co.uk/newsround/47523993

Lesson 3:
You will need an understanding of where websites are stored, this is also explained in 'A Packet's Tale' (a YouTube video): https://www.youtube.com/watch?v=ewrBaIT_eBM

Lesson 4:
An understanding of the elements common to many websites (text content, images, video, etc.). A knowledge of websites which can be used to generate content on the World Wide Web, in particular Chrome Music Lab.

Lesson 5:
A knowledge of copyright and the reasons for it. A useful short summary is here: https://www.gov.uk/copyright and a useful guide to creative commons: https://creativecommons.org/licenses/

Lesson 6
An awareness that there is a high volume of inaccurate, misleading, or false content on the internet. An understanding that search results are influenced by adverts and sponsored content. An awareness of how quickly information spreads around the World Wide Web.

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

Year 4 – Photo editing

Unit introduction

In this unit, learners will develop their understanding of how digital images can be changed and edited, and how they can then be resaved and reused. They will consider the impact that editing images can have, and evaluate the effectiveness of their choices.

Overview of lessons

Lesson	Brief overview	Learning objectives
1 Changing digital images	In this lesson, learners will be introduced to the online editor, and changes that can be made to images using a range of tools. They will look at changing the composition of images using the 'crop' tool, and evaluate the effect that this can have on an image.	To explain that digital images can be changed • I can identify changes that we can make to an image • I can explore how images can be changed in real life • I can explain the effect that editing can have on an image
2 Changing the composition of images	In this lesson, learners will identify changes that have been made to edited images. They will search for and save images from a copyright-free website. Learners will then use an image editor to make a new image composition linked to a cross-curricular theme.	To change the composition of an image • I can explain what has changed in an edited image • I can change the composition of an image by selecting parts of it • I can consider why someone

		might want to change the composition of an image
3 Changing images for different uses	In this lesson, learners will look at the effect that different colours and filters can have on an image. They will choose appropriate effects to fit a scenario, and explain how they made their choices. They will then edit the same original image using different effects to suit two different scenarios, and compare the two versions.	To describe how images can be changed for different uses • I can talk about changes made to images • I can choose effects to make my image fit a scenario • I can explain why my choices fit a scenario
4 Retouching images	This lesson is based on editing images by using retouching tools. Learners will consider why people may choose to retouch images, and the positive and negative effects that retouching can have on images. They will use retouching tools to improve images, and consider which tools are appropriate for retouching.	To make good choices when selecting different tools • I can identify how an image has been retouched • I can give examples of positive and negative effects that retouching can have on an image • I can choose appropriate tools to retouch an image
5 Fake images	This lesson is based on the concept of fake images. Learners will sort images into 'fake' and 'real', and give reasons for their decisions. They will create their own fake images and reflect on how easy it is to digitally alter images, and what this might mean for the images that they see around them.	To recognise that not all images are real • I can sort images into 'fake' or 'real' and explain my choices • I can combine parts of images to create new images • I can talk about fake images around me

6 Making and evaluating a publication	This lesson is the final lesson in the unit on photo editing. Learners will use the 'fake' image that they created in lesson 5 to make a publication designed to advertise their imaginary place. They will add elements such as text, shapes, and borders. They will design a survey for gaining feedback on their work, and compare their completed publications with the original images.	To evaluate how changes can improve an image • I can consider the effect of adding other elements to my work • I can compare the original image with my completed publication • I can evaluate the impact of my publication on others through feedback

Progression

Learners should have experience of making choices on a tablet/computer. They should be able to navigate within an application.

This unit progresses students' skills through editing digital images and considering the impact that editing can have on an image. Learners will also consider how editing can be used appropriately for different scenarios, and create and evaluate 'fake' images, combining all of their new skills.

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

Curriculum links

Computing national curriculum links

- Use search technologies effectively

- 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
- Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

[Education for a Connected World links]

Self-image and identity

- I can describe ways in which people might make themselves look different online.

Copyright and ownership

- When searching on the internet for content to use, I can explain why I need to consider who owns it and whether I have the right to reuse it.

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.

Summative assessment

Please see the assessment rubric document for this unit.

Subject knowledge

All lessons

- You will need to be familiar with the tools used throughout the unit in paint.net or your chosen image editor, and know how to save a new version of an image from within the editor. You can find a guide to all tools in paint.net at www.getpaint.net/doc/latest/index.html.
- You should consider how the learners will access the editor. For example, you may wish to create a shortcut to the program for them.

Lesson 1

- You will need to be familiar with the effect that cropping can have on an image. You can find more information at www.dpreview.com/forums/post/56318241.

Lesson 2

- You will need to know how to search for and save an image from pixabay.com.
- You will need to be familiar with how to combine parts of two images in your chosen image editor.

Lesson 3

- You will need to be familiar with how to make image adjustments and change effects in paint.net or your chosen image editor.

Lesson 4

- You will need to be familiar with the following tools in paint.net or your chosen image editor. For more information about tools in paint.net, visit the following websites:
    - The 'clone stamp': www.getpaint.net/doc/latest/CloneStamp.html
    - The 'recolor' tool: www.getpaint.net/doc/latest/RecolorTool.html
    - The 'magic wand' tool: www.getpaint.net/doc/latest/MagicWand.html

Lesson 5

- You will need to be familiar with the 'lasso select' tool in paint.net or your chosen image editor. For more information about this tool in paint.net, visit www.getpaint.net/doc/latest/LassoSelectionTool.html.

Lesson 6

- You will need to be familiar with the text and shape tools in paint.net or your chosen image editor. For more information about these tools in paint.net, visit www.getpaint.net/doc/latest/TextShapeTools.html.

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

Year 4 – Programming A – Repetition in shapes

Unit introduction

Learners will create programs by planning, modifying, and testing commands to create shapes and patterns. They will use Logo, a text-based programming language.
This unit is the first of the two programming units in Year 4, and looks at repetition and loops within programming

There are two Year 4 programming units:

- Programming A – Repetition in shapes
- Programming B – Repetition in games

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

You can use either a tablet, desktop or laptop computer for this unit. Logo software should be installed or accessible online, for example:

- You can use Turtle Academy online at turtleacademy.com/playground
- You can download FMSLogo from fmslogo.sourceforge.net

Note: The activities will be easier to complete on a laptop or desktop computer as there is more screen area available.

Overview of lessons

Lesson	Brief overview	Learning objectives
1 Programming a screen turtle	This lesson will introduce pupils to programming in Logo. Logo is a text-based programming language where pupils type commands that are then drawn on screen. Pupils will learn the basic Logo commands, and will use their knowledge of them to read and write code.	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
2 Programming letters	In this lesson, pupils will create algorithms (a precise set of ordered instructions, which can be turned into code) for their initials. They will then implement these algorithms by writing them in Logo commands to draw the letter. They will debug their code by finding and fixing any errors that they spot.	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
3 Patterns and repeats	In this lesson, pupils will first look at examples of patterns in everyday life. They will recognise where numbers, shapes, and symbols are repeated, and how many times repeats occur. They will create algorithms for drawing a square, using the same annotated diagram as in Lesson 2. They will use this algorithm to program a square the ‘long’ way, and recognise the repeated pattern within a square. Once they know the repeated pattern, they will use the repeat command within Logo to program squares the ‘short’ way.	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
4 Using loops to create shapes	In this lesson, pupils will work with count-controlled loops in a range of contexts. First, they will think about a real-life example, then they will move on to using count-controlled loops in regular 2D shapes. They will trace code to predict which	To modify a count-controlled loop to produce a given outcome • I can identify the effect of

	shapes will be drawn, and they will modify existing code by changing values within the code snippet.	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
5 Breaking things down	In this lesson, pupils will focus on decomposition. They will break down everyday tasks into smaller parts and think about how code snippets can be broken down to make them easier to plan and work with. They will learn to create, name, and call procedures in Logo, which are code snippets that can be reused in their programming.	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
6 Creating a program	In the final lesson, pupils will apply the skills that they have learnt in this unit to create a program containing a count-controlled loop. Over the course of the lesson, they will design wrapping paper using more than one shape, which they will create with a program that uses count-controlled loops. They will begin by creating the algorithm, either as an annotated sketch, or as a sketch and algorithm, and then implement it as code. They will debug their work throughout, and evaluate their programs against the original brief.	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

Progression

This unit progresses students' knowledge and understanding of programming. It progresses from the sequence of commands in a program to using count-controlled loops. Pupils will create algorithms and then implement those algorithms as code.

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

Curriculum links

National curriculum 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

Summative assessment

- Please see the assessment question and answer documents for this unit.

Subject knowledge

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 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’.

Different pedagogies are used in this programming unit. For example, 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.

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

Year 5 – Sharing information

Unit introduction

In this unit, learners will develop their understanding of computer systems and how information is transferred between systems and devices. Learners will consider small-scale systems as well as large-scale systems. They will explain the input, output, and process aspects of a variety of different real-world systems. Learners will also take part in a collaborative online project with other class members and develop their skills in working together online.

Overview of lessons

Lesson	Brief overview	Learning objectives
1 Systems	This lesson introduces learners to the concept of a system. Learners will develop their understanding of components working together to make a whole. They will outline how digital systems might work and the physical and electronic connections that exist.	To explain that computers can be connected together to form systems • I can explain that systems are built using a number of parts • I can describe that a computer system features inputs, processes, and outputs • I can explain that computer systems communicate with other devices
2 Computer systems and us	In this lesson, learners will consider how larger computer systems work. Learners will consider how devices and processes are connected. They will also reflect on how computer systems can help us.	To recognise the role of computer systems in our lives • I can identify tasks that are managed by computer systems • I can identify the human elements of a

		computer system • I can explain the benefits of a given computer system
3 Transferring information	This lesson introduces the idea that parts of a computer system are not always in the same place or country. Instead, those parts of a system must transfer information using the internet. This lesson builds on the introduction to the internet in the Year 4 'What is the internet?' unit, adding awareness of IP addresses and the rules (protocols) that computers have for communicating with one another.	To recognise how information is transferred over the internet • I can recognise that data is transferred using agreed methods • I can explain that networked digital devices have unique addresses • I can explain that data is transferred over networks in packets
4 Working together	In this lesson, learners will consider how people can work together when they are not in the same location. They will discuss ways of working and start a collaborative online project. The online activity assumes that learners can make simple slides including text and images. If your learners are unsure how to do this, you may wish to spend some time on the Year 3 'Desktop publishing' unit before this lesson.	To explain how sharing information online lets people in different places work together • I can recognise that connected digital devices can allow us to access shared files stored online • I can send information over the internet in different ways • I can explain that the internet allows different media to be shared
5 Better working together	In this lesson, learners will reflect on how they worked together in the previous lesson and how their working together might be improved. Learners will work together on an unplugged activity and use that experience to develop their own ideas of good collective working practices.	To contribute to a shared project online • I can suggest strategies to ensure successful group work • I can make thoughtful suggestions on my group's work

		• I can compare working online with working offline
6 Shared working	In the previous two lessons, learners worked together online on a shared project. This lesson introduces another approach to online working: reusing and modifying work done by someone else. (Using someone else's work needs to be done within the bounds of copyright and with the relevant permissions.) This lesson uses the Scratch programming tool, which allows learners to use other people's work.	To evaluate different ways of working together online • I can identify different ways of working together online • I can recognise that working together on the internet can be public or private • I can explain how the internet enables effective collaboration

Progression

This unit progresses learners' knowledge and understanding of computing systems and online collaborative working.

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

Curriculum links

National curriculum 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
- Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration

- 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
- Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact

Education for a Connected World links

- I can assess and justify when it is acceptable to use the work of others
- I can give examples of content that is permitted to be reused

Assessment

Summative assessment

Please see the assessment question and answer documents for this unit.

Subject knowledge

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

Resources are updated regularly — please check that you are using the latest version.

Year 5 – Programming A – Selection in physical computing

Unit introduction

In this unit, learners will use physical computing to explore the concept of selection in programming through the use of the Crumble programming environment. Learners will be introduced to a microcontroller (Crumble controller) and learn how to connect and program components (including output devices — LEDs and motors) through the application of their existing programming knowledge. Learners will be introduced to conditions as a means of controlling the flow of actions, and explore how these can be used in algorithms and programs through the use of an input device (push switch). Learners 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. To conclude the unit, learners will design and make a working model of a fairground carousel that will incorporate their understanding of how the microcontroller and its components are connected, and how selection can be used to control the operation of the model. Throughout this unit, pupils will apply the stages of programming design.

There are two Year 5 programming units:

- Programming A – Selection in physical computing
- Programming B – Selection in quizzes

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

Overview of lessons

Lesson	Brief overview	Learning objectives
1 Connecting Crumbles	In this lesson, learners will become familiar with the Crumble controller, some of its associated components, and the programming environment used to control it. They will explore how the items connect together to create a complete circuit, and how to construct programs that turn an LED on and off and set its colour. Learners will apply their understanding of repetition by identifying how their programs can be modified to make an LED flash continuously.	To control a simple circuit connected to a computer • I can build a simple circuit to connect a microcontroller to a computer • I can program a microcontroller to light an LED

		• I can explain why I used an infinite loop
2 Combining output devices	In this lesson, learners will develop their knowledge of a Crumble controller further by connecting additional devices (another Sparkle and a motor) to the controller, and they will construct programs to control more than one of these. They will design sequences of actions for these output devices. They will then apply their understanding of repetition by using count-controlled loops when implementing their design as a program.	To write a program that includes count-controlled loops • I can connect more than one output device to a microcontroller • I can design sequences for given output devices • I can decide which output devices I control with a count-controlled loop
3 Controlling with conditions	In this lesson, learners will be introduced to conditions, and how they can be used in algorithms and programs to control their flow. They will identify conditions in statements, stating if they are true or false, and learn how they can be used to start and stop a set of actions. Learners will be introduced to a Crumble switch, and learn how it can provide the Crumble controller with an input that can be used as a condition. They will explore how to write programs that use an input as a condition, and use this knowledge to write a program that uses a condition to stop a repeating light pattern.	To explain that a loop can stop when a condition is met, eg number of times • I can explain that a condition is something that can be either true or false (eg whether a value is more than 10, or whether a button has been pressed) • I can experiment with a ‘do until’ loop • I can program a microcontroller to respond to an input
4 Starting with selection	In this lesson, learners will develop their understanding of how the flow of actions in algorithms and programs can be controlled by conditions. They will be	To conclude that a loop can be used to repeatedly check whether a condition has

	introduced to selection, and learn to represent conditions and actions using the 'if… then…' structure. They will apply their understanding by using selection in an algorithm created to meet the requirements of a task. They will discover that infinite repetition is required when programming input devices to repeatedly check if a condition has been met.	been met • I can explain that a condition being met can start an action • I can identify a condition and an action in my project • I can use selection (an 'if… then…' statement) to direct the flow of a program
5 Drawing designs	In this lesson, learners will apply their understanding of microcontrollers, output devices, and selection when designing a project to meet the requirements of a given task. To ensure their understanding, they will identify how selection might be used in real-world situations, then they will consider how they can apply this knowledge when designing their project. They will produce detailed drawings to show how their model will be made and how they will connect the microcontroller to its components.	To design a physical project that includes selection • I can identify a condition to start an action (real world) • I can describe what my project will do (the task) • I can create a detailed drawing of my project
6 Writing and testing algorithms	In this final lesson of the unit, learners will build on the designs that they developed in Lesson 5 by creating an algorithm to meet the requirements of the given task. They will identify how they are going to use selection before writing their algorithm. They will then move into the code level to test their algorithm by implementing it as a program, running it, identifying any bugs, and returning to the algorithm to debug it where necessary. Finally, to conclude the unit, they will evaluate their algorithms and other areas of their designs.	To create a controllable system that includes selection • I can write an algorithm to control lights and a motor • I can use selection to produce an intended outcome • I can test and debug my project

Progression

This unit assumes that learners will have prior experience of programming using block-based construction (eg Scratch) and understand the concepts of sequence and repetition. The National Centre for Computing Education key stage 1 units focus on floor robots and ScratchJr, however, experience of other languages or environments may also be useful.

See the learning graph for this unit for more information about progression.

Curriculum links

Computing

- 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

Science – Electricity (Year 4)

- construct a simple series electrical circuit, identifying and naming its basic parts, including cells, wires, bulbs, switches and buzzers

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.

Summative assessment

See the assessment rubric to support summative assessment for this unit.

Resources

The unit has been designed to make use of the components provided in the Crumble starter kit, which are as follows:

- 1 Crumble controller
- 12 crocodile leads
- 2 Sparkles (A Sparkle is an RGB LED — red, green, blue light-emitting diode. The D connector allows the Crumble to use an electronic signal to control the Sparkle. The signal sets the colour and brightness of the LED.)
- 1 push switch suitable for Crumble
- 1 light sensor suitable for Crumble
- 1 buzzer suitable for Crumble
- 1 micro USB cable
- 1 switched battery box suitable for Crumble

Unless stated otherwise in the individual lesson plan, learners will need access to these resources (preferably one kit per pair) in each lesson. In Lessons 2, 5, and 6, learners will also need to use motors (this is also indicated in the individual lesson plans).

Learners will also need access to devices capable of running the Crumble software. This is currently available for Microsoft Windows (XP SP3 or newer) and macOS (10.6 and above). Download the software from redfernelectronics.co.uk/crumble-software.

Subject knowledge

This unit focuses on physical computing that allows learners to control real-life events through the construction of programs. When learners undertake physical computing, they write programs that control real-world objects, like LEDs and motors, using a computer. The tangible effect of seeing the commands that they entered into a computer being carried out on a physical item, rather than on screen, can be highly motivational for learners. Physical computing also offers the opportunity to take a more project-based approach to learning, and allows learners to make choices about the purpose, design, and program of their product.

Throughout this unit, there are opportunities to demonstrate a concept within the Crumble programming software or play a video. Pedagogically, it is more beneficial to demonstrate the concepts to learners, as it allows for easier questioning and understanding. We recommend that you use the videos to see what to demonstrate, then show learners with a live demonstration, however, videos are provided on the slides if you wish to use them instead.

For this unit, you will need experience of constructing programs using the Crumble programming software (see the 'Resources' section at the end of this document). It uses the same drag-and-drop style as Scratch. You will need to write programs that turn LEDs (Sparkles) on and off, change LED colours, spin motors, use push switches as inputs, and combine a number of these peripherals. Additionally, you will be connecting the Crumble controller with battery packs, Sparkles, motors, and push switches. For further support on using Crumbles, see the Crumble 'Getting Started' guide at redfernelectronics.co.uk/crumble-getting-started.

Levels of abstraction

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

Spending time at the 'Task' and 'Design' levels before engaging in writing code aids learners in assessing the 'do-ability' 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.

Repetition

You will need to know that repetition is used in programming to give the same instruction or set of instructions several times. Repetition uses loops as the means to give these instructions. This unit makes use of two types of loops: infinite and count-controlled. These have been defined below.

Infinite loop

An infinite loop is a loop that commands the instruction/set of instructions to repeat forever. When an infinite loop is used in a program, there is no way of ending the program, as the command(s) within the loop will be repeated endlessly. For this reason, infinite loops should only be used when writing a program that is intended to run forever. The exception to this is when using selection in physical computing, as you will see throughout this unit.

Count-controlled loop

A count-controlled loop is a form of repetition in which a set of commands are carried out a specific number of times. Count-controlled loops should only be used when it is known how many times a set of commands need to be repeated.

Condition-controlled loop

A condition-controlled loop is a form of repetition in which a set of commands stop being carried out when a condition is met. The condition could be anything from when 'score' in a game reaches a certain value to when a key on a keyboard has been pressed.

Conditions

Conditions are statements that need to be met for a set of actions to be carried out. They can be used in algorithms and programs to control the flow of actions. When a condition is met, it is referred to as 'true' and when it is not met, it is referred to as 'false'. You will need to be able to identify and use conditions in algorithms in the form of statements to both start and stop sets of action. Additionally, you will need to understand that conditions can be used in loops, and when they are, that the set of actions in the loop will be carried out repeatedly until the condition is true, for example, 'until button A is pressed'.

Selection

When designing programs, there are often points where a decision must be made. BBC Bitesize defines selection as:

> Selection – a decision within a computer program when the program decides to move on based on the results of an event (source: BBC Bitesize)

These decisions are known as selection, and are implemented in programming using if statements. Selection is used to control the flow of actions in algorithms and programs by checking if a condition (see above) has been met. If it has been met, the identified actions will be carried out. When selection is used in programs, loops (see above) have to be used to instruct the device to check the condition repeatedly. Without using loops, the condition would only be checked once. In the Crumble programming software, selection is implemented through the if... then... command block.

In addition to the above, you will also need to understand that programs are an implementation of an algorithm, and that when the program does not produce the required output, the algorithm should be debugged. This should then be implemented in the program.

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

Resources are updated regularly — please check that you are using the latest version.

Year 5 – Programming B – Selection in quizzes

Unit introduction

In this unit, pupils develop their knowledge of 'selection' by revisiting how 'conditions' can be used in programming, and then learning how the 'if... then... else...' structure can be used to select different outcomes depending on whether a condition is 'true' or 'false'. They represent this understanding in algorithms, and then by constructing programs using the Scratch programming environment. They learn how to write programs that ask questions and use selection to control the outcomes based on the answers given. They use this knowledge to design a quiz in response to a given task and implement it as a program. To conclude the unit, learners evaluate their program by identifying how it meets the requirements of the task, the ways they have improved it, and further ways it could be improved.

There are two Year 5 programming units:

- Programming A – Selection in physical computing
- Programming B – Selection in quizzes

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

Overview of lessons

Lesson	Brief overview	Learning objectives
Exploring conditions	In this lesson, learners revisit previous learning on 'selection' and identify how 'conditions' are used to control the flow of actions in a program. They are introduced to the blocks for using conditions in programs using the Scratch programming environment. They modify the conditions in an existing program and identify the impact this has.	To explain how selection is used in computer programs • I can recall how conditions are used in selection • I can identify conditions in a program

		• I can modify a condition in a program
Selecting outcomes	In this lesson, learners will develop their understanding of selection by using the 'if… then… else…' structure in algorithms and programs. They will revisit the need to use repetition in selection to ensure that conditions are repeatedly checked. They identify the two outcomes in given programs and how the condition informs which outcome will be selected. Learners use this knowledge to write their own programs that use selection with two outcomes.	To relate that a conditional statement connects a condition to an outcome • I can use selection in an infinite loop to check a condition • I can identify the condition and outcomes in an 'if… then… else…' statement • I can create a program with different outcomes using selection
Asking questions	In this lesson, learners consider how the 'if… then… else…' structure can be used to identify two responses to a binary question (one with a 'yes or no' answer). They identify that the answer to the question is the 'condition', and use algorithms with a branching structure to represent the actions that will be carried out if the condition is true or false. They learn how questions can be asked in Scratch, and how the answer, supplied by the user, is used in the condition to control the outcomes. They use an algorithm to design a program that uses selection to direct the flow of the program based on the answer provided. They implement their algorithm as a program and test whether both outcomes can be achieved.	To explain how selection directs the flow of a program • I can explain that program flow can branch according to a condition • I can design the flow of a program which contains 'if… then… else…' • I can show that a condition can direct program flow in one of two ways
Planning a quiz	In this lesson, learners will be provided with a task: to use selection to	To design a program which uses

	control the outcomes in an interactive quiz. They will outline the requirements of the task and use an algorithm to show how they will use selection in the quiz to control the outcomes based on the answer given. Learners will complete their designs by using storyboards to identify the questions that will be asked, and the outcomes for both correct and incorrect answers. To demonstrate their understanding of how they are using selection to control the flow of the program, learners will identify which outcomes will be selected based on given responses.	selection • I can outline a given task • I can use a design format to outline my project • I can identify the outcome of user input in an algorithm
Testing a quiz	In this lesson, learners will use the Scratch programming environment to implement the first section of their algorithm as a program. They will run the first section of their program to test whether they have correctly used selection to control the outcomes, and debug their program if required. They will then continue implementing their algorithm as a program. Once completed, they will consider the value of sharing their program with others so that they can receive feedback. Learners conclude the lesson by using another learner's quiz and providing feedback on it.	To create a program which uses selection • I can implement my algorithm to create the first section of my program • I can test my program • I can share my program with others
Evaluating a quiz	In this lesson, learners will return to their completed programs and identify ways in which the program can be improved. They will focus on issues where answers similar to those in the condition are given as inputs, and identify ways to avoid such problems. Learners will also consider how the outcomes may change the program for subsequent users, and identify how they can make use of setup to provide all users with the same experience. They will implement their identified improvements by returning to the Scratch programming environment and adding to their programs. They conclude the unit by identifying how	To evaluate my program • I can identify ways the program could be improved • I can identify the setup code I need in my program • I can extend my program further

	they met the requirements of the given task, and identifying the aspects of the program that worked well, those they improved, and areas that could improve further.	

Progression

This unit assumes that learners will have prior experience of programming using block-based construction (eg Scratch), understand the concepts of 'sequence' and 'repetition', and have some experience of using 'selection'. Ideally, learners will have completed 'Programming A – Selection in physical computing' before undertaking this unit, as this will provide them with the required knowledge of 'selection'.

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

Curriculum links

Computing

- 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

Assessment

Formative assessment

Assessment opportunities are detailed in each lesson plan. The learning objectives 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 objectives using thumbs up, thumbs sideways, or thumbs down.

We recommend the use of teacher accounts in Scratch to help with assessment throughout this unit. For guidance on setting up teacher accounts, please visit the Scratch website (scratch.mit.edu/educators/faq).

Summative assessment

- Please see the assessment question and answer documents for this unit.

Subject knowledge

This unit focuses on developing learners' understanding of selection in an on-screen context. It highlights what 'conditions' are and how they are used as part of 'selection'. This unit also develops learners' understanding of design in programming, using the approach outlined below.

Levels of abstraction

When programming, there are four levels which 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 - this is what is needed
- Design - this is what it should do
- Code - this is how it is done
- Running the code - this is what it does

Spending time at the 'Task' and 'Design' levels before engaging in code-writing aids learners in assessing the 'do-ability' 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.

Conditions

'Conditions' are statements that need to be met for a set of actions to be carried out. They can be used in algorithms and programs to control the flow of actions. When a condition is met it is referred to as 'true' and when it is not met it is referred to as 'false'. You need to be able to identify and use conditions in algorithms in the form of statements to both start and stop sets of action. Additionally, you need to understand that conditions can be used in loops, and when they are, that the set of actions in the loop will be carried out repeatedly until the condition is true. For example, 'until button 'A' is pressed'.

Selection

When designing programs, there are often points where a decision must be made. These decisions are known as 'selection', and are commonly implemented in programming using 'if' statements. Selection is used to control the flow of actions in algorithms and programs by checking whether a condition (see above) has been met. If it has been met, the identified actions will be carried out. When selection is used in programs, infinite loops (see above) are often used to instruct the device to check the condition repeatedly. Without using loops, the condition would only be checked once following the sequence of the code.

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

Year 6 – Communication

Unit introduction

In this unit, the class will learn about the World Wide Web as a communication tool. First, they will learn how we find information on the World Wide Web, through learning how search engines work (including how they select and rank results) and what influences searching, and through comparing different search engines. They will then investigate different methods of communication, before focusing on internet-based communication. Finally, they will evaluate which methods of internet communication to use for particular purposes.

Overview of lessons

Lesson	Brief overview	Learning objectives
1 Searching the web	In this lesson, learners will be introduced to a range of search engines. They will be given the opportunity to explain how we search, then they will write and test instructions. Next, they will learn that searches do not always return the results that we are looking for, and will refine their searches accordingly. Finally, they will be introduced to the two most common methods of searching: using a search engine and the address bar.	To identify how to use a search engine • I can complete a web search to find specific information • I can refine my search • I can compare results from different search engines
2 Selecting search results	In this lesson, learners will gain an understanding of why search engines are necessary to help us find things on the World Wide Web. They will conduct their own searches and	To describe how search engines select results • I can explain why we need tools to find things online

	break down, in detail, the steps needed to find things on the web. They will then emulate web crawlers to create an index of their own classroom. Finally, they will consider why some searches return more results than others.	• I can recognise the role of web crawlers in creating an index • I can relate a search term to the search engine's index
3 How search results are ranked	This lesson includes an unplugged activity in which the class will learn about some of the main factors that influence how a search engine ranks a web page. Learners will create paper-based 'web pages' in groups, on a topic that they are currently studying. They will then discover how their web pages would rank when searching for keywords relating to their content.	To explain how search results are ranked • I can explain that search results are ordered • I can explain that a search engine follows rules to rank relevant pages • I can suggest some of the criteria that a search engine checks to decide on the order of results
4 How are searches influenced?	In this lesson, learners will explore how the person performing a web search can influence the results that are returned, and how content creators can optimise their sites for searching. Learners will also explore some of the limitations of searching, then discuss what cannot be searched.	To recognise why the order of results is important, and to whom • I can describe some of the ways that search results can be influenced • I can recognise some of the limitations of search engines • I can explain how search engines make money
5 How we communicate	In this lesson, learners will deepen their understanding of the term 'communication'. They will explore different methods of communication, then they will consider internet-based communication in more detail. Finally, they will evaluate	To recognise how we communicate using technology • I can explain the different ways in which people communicate

	which methods of communication suit particular purposes.	• I can identify that there are a variety of ways of communicating over the internet • I can choose methods of communication to suit particular purposes
6 Communicating responsibly	In this lesson, learners will use information provided and their own prior knowledge to categorise different forms of internet communication. They will then choose which method they would use for the scenarios discussed in the previous lesson. During these activities, they will explore issues around privacy and information security.	To evaluate different methods of online communication • I can compare different methods of communicating on the internet • I can decide when I should and should not share • I can explain that communication on the internet may not be private

Progression

This unit progresses students' knowledge and understanding of computing systems and online collaborative working.

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

Curriculum links

National curriculum links

- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration

- Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- 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
- Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact

Education for a Connected World links

- I can describe and assess the benefits and the potential risks of sharing information online.
- I can use various additional tools to refine my searches (e.g. search filters: size, type, usage rights etc.).
- I can explain how to use search effectively and use examples from my own practice to illustrate this.
- I can explain how search engine rankings are returned and can explain how they can be influenced (e.g. commerce, sponsored results).

Assessment

Summative assessment

- Please see the assessment question and answer documents for this unit.

Subject knowledge

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

Resources are updated regularly — please check that you are using the latest version.

Year 6 – Web page creation

Unit introduction

This unit introduces learners to the creation of websites for a chosen purpose. Learners identify what makes a good web page and use this information to design and evaluate their own website using Google Sites. Throughout the process learners pay specific attention to copyright and fair use of media, the aesthetics of the site, and navigation paths.

Overview of lessons

Lesson	Brief overview	Learning objectives
What makes a good website?	In this lesson learners will explore and review existing websites and evaluate their content. They will have some understanding that websites are created using HTML code.	To review an existing website and consider its structure • I can explore a website • I can discuss the different types of media used on websites • I know that websites are written in HTML
How would you lay out your web page?	Learners will look at the different layout features available in Google Sites and plan their own web page on paper. **Homework:** Learners will look at two of their favourite websites and sketch them on the worksheet provided, detailing the similarities and differences. **Note:** For the homework activity, teachers could provide printed 'home page' images for anyone who doesn't have	To plan the features of a web page • I can recognise the common features of a web page • I can suggest media to include on my page • I can draw a web page layout that suits my purpose

	internet access at home.	
Copyright or copyWRONG?	During this lesson learners will become familiar with the terms 'fair use' and 'copyright'. They will gain an understanding of why they should only use copyright-free images and will find appropriate images to use in their work from suggested sources. **Homework:** Learners answer a series of questions based on copyright and fair use.	To consider the ownership and use of images (copyright) • I can say why I should use copyright-free images • I can find copyright-free images • I can describe what is meant by the term 'fair use'
How does it look?	Today learners will revise how to create their own web page in Google Sites. Using their plan from previous lessons, learners will create their own web page/home page. They will preview their web page as it will appear on different devices and suggest or make edits to improve the user experience on each device.	To recognise the need to preview pages • I can add content to my own web page • I can preview what my web page looks like • I can evaluate what my web page looks like on different devices and suggest/make edits.
Follow the breadcrumbs	During this lesson learners will begin to appreciate the need to plan the structure of a website carefully. They will plan their website, paying attention to the navigation paths (the way that pages are linked together). They will then create multiple web pages for their site and use hyperlinks to link them together as detailed in their planning.	To outline the need for a navigation path • I can explain what a navigation path is • I can describe why navigation paths are useful • I can make multiple web pages and link them using hyperlinks
Think before you link!	Learners will consider the implications of linking to content owned by other people and create hyperlinks on their own websites that link to other people's work. They will then evaluate the user experience when using their own website	To recognise the implications of linking to content owned by other people • I can explain the implication of linking to content owned by others

	and that of another learner.	• I can create hyperlinks to link to other people's work • I can evaluate the user experience of a website

Progression

This unit progresses students' knowledge and understanding of the following: digital writing, digital painting, desktop publishing, digital photography, photo editing, and vector drawing.

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

Curriculum links

National curriculum links

- Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- 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.
- use technology safely, respectfully, and responsibly; recognise acceptable/unacceptable behaviour.

English links

- Writing composition: Identifying the audience for and purpose of the writing, selecting the appropriate form, and using other similar writing as models for their own.

[Education for a Connected World links]

Online relationships

- I can use the internet with adult support to communicate with people I know. (EY-7)

Managing information online

- I can navigate online content, websites, or social media feeds using more sophisticated tools to get to the information I want (e.g. menus, sitemaps, breadcrumb-trails, site search functions). (11-14)

Copyright and ownership

- I can explain why copying someone else's work from the internet without permission can cause problems.
- I can give examples of what those problems might be.
- When searching on the internet for content to use, I can explain why I need to consider who owns it and whether I have the right to reuse it.
- I can give some simple examples.
- I can assess and justify when it is acceptable to use the work of others.
- I can give examples of content that is permitted to be reused.
- I can demonstrate the use of search tools to find and access online content which can be reused by others.
- I can demonstrate how to make references to and acknowledge sources I have used from the internet.
- I can explain the principles of fair use and apply this to case studies. (11-14)

Assessment

Summative assessment

- Please see the assessment rubric document for this unit.

Subject knowledge

This unit focuses on the design and creation of web pages and websites using Google Sites.

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

Resources are updated regularly — please check that you are using the latest version.

Year 6 – Programming A – Variables in games

Unit introduction

This unit explores the concept of variables in programming through games in Scratch. First, pupils will learn what variables are, and relate them to real-world examples of values that can be set and changed. Pupils will then use variables to create a simulation of a scoreboard. In Lessons 2, 3, and 5, which follow the Use-Modify-Create model, pupils will experiment with variables in an existing project, then modify them, then they will create their own project. In Lesson 4, pupils will focus on design. Finally, in Lesson 6, pupils will apply their knowledge of variables and design to improve their game in Scratch.

There are two Year 6 programming units:

- Programming A – Variables in games
- Programming B – Sensing

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

Overview of lessons

Lesson	Brief overview	Learning objectives
1 Introducing variables	In this lesson, pupils will be introduced to variables. Pupils will see examples of real-world variables (score and time in a football match), then they will explore them in a Scratch project. Pupils will then design and make their own project including variables. Finally, pupils will identify that variables are named and can be letters (strings) as well as numbers.	To define a 'variable' as something that is changeable • I can identify examples of information that is variable • I can explain that the way that a variable changes can be defined

		• I can identify that variables can hold numbers or letters
2 Variables in programming	In this lesson, pupils will understand that variables are used in programs, and that they can hold a single value at a time. Pupils will complete an unplugged task that will demonstrate the process of changing variables. Next, they will explore why it is important to name variables, then they will apply their learning in a Scratch project in which they will make, name, and update variables.	To explain why a variable is used in a program • I can identify a program variable as a placeholder in memory for a single value • I can explain that a variable has a name and a value • I can recognise that the value of a variable can be changed
3 Improving a game	In this lesson, pupils will apply the concept of variables to enhance an existing game in Scratch. They will predict the outcome of changing the same `change score` block in different parts of a program, then they will test their predictions in Scratch. They will also experiment with using different values in variables, and with using a variable elsewhere in a program. Finally, they will add comments to their project, explaining how they have met the objectives of the lesson.	To choose how to improve a game by using variables • I can decide where in a program to change a variable • I can make use of an event in a program to set a variable • I can recognise that the value of a variable can be used by a program
4 Designing a game	This lesson focuses on the design elements of programming. For the majority of the tasks, pupils will be working at the algorithmic level of abstraction. Pupils will first design the sprites and backgrounds for their project, then they will design their algorithms to create their program flow.	To design a project that builds on a given example • I can choose the artwork for my project • I can explain my design

		choices • I can create algorithms for my project
5 Design to code	In this lesson, pupils will implement the algorithms that they created in Lesson 4 as code. In doing this, they will identify variables in an unfamiliar project and learn the importance of naming variables. They will also have the opportunity to add another variable to enhance their project.	To use my design to create a project • I can create the artwork for my project • I can choose a name that identifies the role of a variable • I can test the code that I have written
6 Improving and sharing	This lesson gives pupils the opportunity to build on the project that they created in Lesson 5. As the lesson develops, the scaffolding is gradually removed, so that the last main activity is without constraints. Finally, pupils will evaluate each other's projects, identifying features that they like, and features that could be improved further.	To evaluate my project • I can identify ways that my game could be improved • I can extend my game further using more variables • I can share my game with others

Progression

This unit assumes that pupils will have some prior experience of programming in Scratch. Specifically, they should be familiar with the programming constructs of sequence, repetition, and selection. These constructs are covered in the Year 3, 4, and 5 National Centre for Computing Education programming units respectively. Each year group includes at least one unit that focuses on Scratch.

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

Curriculum links

National curriculum 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

Assessment opportunities are detailed in each lesson plan. The learning objectives 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.

We recommend the use of teacher accounts in Scratch to help with assessment throughout this unit. For guidance on setting up teacher accounts, please visit the Scratch website (scratch.mit.edu/educators/faq).

Summative assessment

Please see the assessment question and answer documents for this unit.

Subject knowledge

This unit focuses on developing pupils' understanding of variables in a new programming language. It highlights where variables can be used and how they can be set and changed through the running of a program. This unit also develops pupils' understanding of design in programming, using the approach outlined below.

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 can aid pupils in assessing the 'do-ability' of their programs. It also 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.

During this unit, pupils are required to save their work in Scratch. We recommend the use of teacher and pupil accounts to manage this process. You can find detailed guidance on setting up and managing accounts in Scratch on the Scratch website (scratch.mit.edu/educators/faq).

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

Resources are updated regularly — please check that you are using the latest version.