



# The Computing Curriculum Year 5

Year 5 – Autumn 1		Unit 1 – Sharing Information	
National Curriculum objectives			
<ul style="list-style-type: none"><li>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</li><li>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</li><li>Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact</li></ul>			
Unit Overview			
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.			
Previous knowledge acquired - Technology			
Year 1	Year 2	Year 3	Year 4
<ul style="list-style-type: none"><li>Know technology as something that helps us</li><li>Know examples of technology in the classroom</li><li>Know how these technology examples help us</li><li>Know the names of the main parts of a computer</li><li>Know how to switch on and log into a computer</li><li>Know how to use a mouse to click and drag</li><li>Know how to use a mouse to open a program</li><li>Know how to click and drag to make objects on a screen</li><li>Know how to use a mouse to create a picture</li><li>Know what a keyboard is for</li><li>Know how to type my name on a computer</li><li>Know how to save my work to a file</li><li>Know how to open my work from</li></ul>	<ul style="list-style-type: none"><li>Know examples of computers</li><li>Know and describe some uses of computers</li><li>Know that a computer is a part of information technology</li><li>Know and explain the purpose of information technology in the home</li><li>Know how to open a file</li><li>Know how to move and resize images</li><li>Know how to find examples of information technology</li><li>Know and talk about uses of information technology</li><li>Compare types of information technology</li><li>Know how information technology is used in a shop</li><li>Know that information technology can be connected</li><li>Know and explain how information technology helps people</li><li>Know different uses of information technology</li></ul>	<ul style="list-style-type: none"><li>Know that digital devices accept inputs.</li><li>Know that digital devices produce outputs.</li><li>Know how to follow a process.</li><li>Know how to classify input and output devices.</li><li>Know how to model a simple process.</li><li>Know how to design a digital device.</li><li>Know how I use digital devices for different activities.</li><li>Know how to recognise similarities between using digital devices and non-digital tools.</li><li>Know the differences between using digital devices and non-digital tools.</li><li>Know how to recognise different connections.</li><li>Know how messages are passed through multiple connections.</li><li>Know why we need a network switch.</li><li>Know that a computer network is made up of a number of devices.</li><li>Know and can demonstrate how</li></ul>	<ul style="list-style-type: none"><li>Know the internet is a network of networks</li><li>Know how information is shared across the internet</li><li>Know why a network needs protecting</li><li>Know the different networked devices and how they connect</li><li>Know how the internet allows us to view the World Wide Web</li><li>Know that the World Wide Web is the part of the internet that contains websites and web pages</li><li>Know the types of media that can be shared on the World Wide Web (WWW)</li><li>Know where websites are stored when uploaded to the WWW</li><li>Know how to access websites on the WWW</li><li>Know how to create media which can be found on websites</li><li>Know that I can add content to the WWW</li><li>Know that new content can be created online</li><li>Know that websites and their content are created by people</li><li>Know who owns the content on websites</li></ul>

<ul style="list-style-type: none"> <li>• a file</li> <li>• Know how to use the arrow keys to move the cursor</li> <li>• Know how to delete letters</li> <li>• Know rules to keep us safe and healthy when we are using technology in and beyond the home and give examples of some of these rules</li> <li>• Know how we benefit from these rules</li> </ul>	<ul style="list-style-type: none"> <li>• Know how to use information technology responsibly</li> <li>• Know how those rules/guides can help me</li> <li>• Identify the choices that I make when using information technology</li> <li>• Know and explain simple guidance for using information technology in different environments and settings</li> </ul>	<p>information can be passed between devices.</p> <ul style="list-style-type: none"> <li>• Know and can explain the role of a switch, server, and wireless access point in a network.</li> <li>• Know and can identify how devices in a network are connected with one another.</li> <li>• Know and can identify networked devices around me.</li> <li>• Know and can identify the benefits of computer networks.</li> </ul>	<ul style="list-style-type: none"> <li>• Know that there are rules to protect content</li> <li>• Know that not everything on the World Wide Web is true.</li> <li>• Know why some information I find online may not be honest, accurate, or legal.</li> <li>• Know why I need to think carefully before I share or reshare content</li> </ul>
--	---	--	---

#### Progression of knowledge through the Computing Curriculum

#### Year 6

Know how to complete a web search to find specific information

- Know how to refine my search
- Know how to compare results from different search engines
- Know why we need tools to find things online
- Know the role of web crawlers in creating an index
- Know how to relate a search term to the search engine's index
- Know that search results are ordered
- Know that a search engine follows rules to rank relevant pages
- Know some of the criteria that a search engine checks to decide on the order of results
- Know some of the ways that search results can be influenced
- Know some of the limitations of search engines
- Know how search engines make money
- Know the different ways in which people communicate
- Know that there are a variety of ways of communicating over the internet
- Know to choose methods of communication to suit particular purposes
- Know how to compare different methods of communicating on the internet
- Know when I should and should not share
- Know that communication on the internet may not be private

#### Key knowledge acquired throughout this unit

- Know that systems are built using a number of parts
- Know that a computer system features inputs, processes, and outputs
- Know that computer systems communicate with other devices
- Know tasks that are managed by computer systems
- Know the human elements of a computer system
- Know the benefits of a given computer system

- Know that data is transferred using agreed methods
- Know that networked digital devices have unique addresses
- Know that data is transferred over networks in packets
- Know that connected digital devices can allow us to access shared files stored online
- Know how to send information over the internet in different ways
- Know that the internet allows different media to be shared
- Know and suggest strategies to ensure successful group work
- Compare working online with working offline
- Know different ways of working together online
- Know that working together on the internet can be public or private
- Know how the internet enables effective collaboration

### Teacher subject knowledge and guidance



Can you name any parts of this bicycle?

### A digital system

The dials and buttons on the front of a washing machine are the inputs.

The computer inside the washing machine follows a program based on the inputs.

The clothes are washed and spun. The display indicates how long the wash is taking. These are outputs.



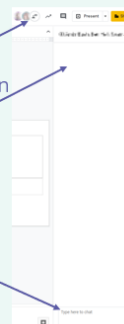
### Create a shared bank of information

Chat

Click on the **Chat** button

Read messages

Type messages in the chat box

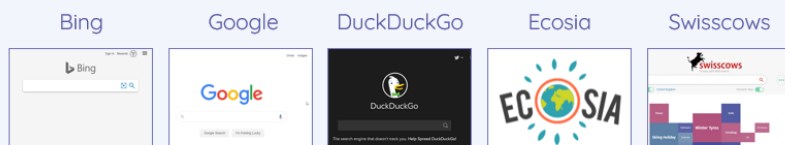
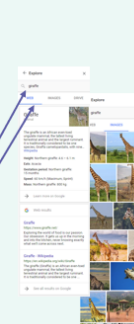


Explore

Click on the **Explore** button at the bottom of the screen

Enter a search term

Choose **Web** or **Images**



### Subject specific vocabulary and definitions (Tier 3 vocabulary)

Input	Data that is sent to a program				
Output	The result of data processed by a computer				
Process	A program that is running on a computer				
Internet	The global system of interconnected computer networks				
Network	A group of interconnected computing devices				
Computer system	A combination of hardware and software that can have data input to it, which it then processes and outputs.				
Medium Term Planning					
Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
To explain that computers can be connected together to form systems	To recognise the role of computer systems in our lives	To recognise how information is transferred over the internet	To explain how sharing information online lets people in different places work together	To contribute to a shared project online	To evaluate different ways of working together online

Year 5 – Autumn 2		Unit 2 – Video Editing	
National Curriculum objectives			
<ul style="list-style-type: none"><li>• Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content</li><li>• 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</li><li>• Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact</li></ul>			
Unit Overview			
Learners will learn how to create short videos by working in pairs or groups. As they progress through this unit, they will be exposed to topic-based language and develop the skills of capturing, editing, and manipulating video. Learners are guided with step-by-step support to take their idea from conception to completion. At the conclusion of the unit, learners have the opportunity to reflect on and assess their progress in creating a video.			
To teach this unit, you will need video recording equipment such as video cameras or tablets with video capabilities. The recommended editing software is Microsoft Video Editor, which is included for free with Windows 10.			
This unit progresses learners’ knowledge and understanding of creating media by guiding them systematically through the process involved in creating a video. The unit builds on the Year 4 unit ‘Photo editing’ where composition is introduced and the Year 3 unit ‘Stop-frame animation’ where learners explored some of the features of video production. By the end of this unit, learners will have developed the skills required to plan, record, edit, and share a video.			
Previous knowledge acquired – Digital Literacy			
Year 1	Year 2	Year 3	Year 4
<ul style="list-style-type: none"><li>• Know how to make marks on a screen and explain which tools were used</li><li>• Know how to draw lines on a screen and explain which tools were used.</li><li>• Know how to use paint tools to draw a picture.</li><li>• Know how to make marks with the square and line tools</li><li>• Know how to use shape and line tools effectively to recreate the work of an artist</li><li>• Choose appropriate shapes</li><li>• Know how to make appropriate colour choices</li><li>• Know how to create a picture in the style of an artist</li><li>• Know how to choose appropriate paint tools and colours to create the work of an artist</li></ul>	<ul style="list-style-type: none"><li>• Know what devices can be used to take photographs</li><li>• Know how to take a photograph</li><li>• Know and explain what I did to capture a digital photo</li><li>• Know the process of taking a good photograph</li><li>• Know how to take photos in both landscape and portrait format</li><li>• Know and explain why a photo looks better in portrait or landscape format</li><li>• Know what is wrong with a photograph</li><li>• Know how to take a good photograph</li><li>• Know that I can improve a photograph by retaking it</li><li>• Know the effect that light has on a photo</li></ul>	<ul style="list-style-type: none"><li>• Know how to draw a sequence of pictures</li><li>• Know how to create flip book—style animation.</li><li>• Know how an animation and flip book works.</li><li>• Know what an animation will look like.</li><li>• Know why little changes are needed for each frame.</li><li>• Know how to create and effective stop frame animation.</li><li>• Know how to break down a story into settings, characters and events.</li><li>• Know how to describe an animation that is achievable on screen.</li><li>• Know how to create a storyboard.</li><li>• Know how to use onion skinning to help me make small changes between frames.</li></ul>	<ul style="list-style-type: none"><li>• Know digital devices that can record sound and play it back</li><li>• Know the inputs and outputs required to play audio or record sound</li><li>• Know the range of sounds that can be recorded</li><li>• Know how to use a device to record audio and play back sound</li><li>• Know how to improve my recording</li><li>• Know what other people include when recording sound for a podcast</li><li>• Know how to plan and write the content for a podcast</li><li>• Know why it is useful to be able to save digital recordings</li><li>• Know how to save a digital recording as a file</li><li>• Know how to open a digital recording from a file</li><li>• Know ways in which audio recordings can be altered</li></ul>

<ul style="list-style-type: none"> <li>• Know which tools were helpful and why</li> <li>• Know how to make dots of colour on the page</li> <li>• Know how to change the colour and brush size</li> <li>• Know how to use dots of colour to create a picture in the style of an artist on my own.</li> </ul>	<ul style="list-style-type: none"> <li>• Know to experiment with different light sources</li> <li>• Know and explain why a picture may be unclear</li> <li>• Know that images can be changed</li> <li>• Know how to use a tool to achieve a desired effect</li> <li>• Know how to apply a range of photography skills to capture a photo</li> <li>• Know which photos have been changed</li> <li>• Know which photos are real and which have been changed</li> </ul>	<ul style="list-style-type: none"> <li>• Know how to review a sequence of frames to check my work.</li> <li>• Know how evaluate the quality of my animation.</li> </ul>	<ul style="list-style-type: none"> <li>• Know how to edit sections of an audio recording</li> <li>• Know sounds that other people combine</li> <li>• Know suitable sounds to include in a podcast</li> <li>• Know how to use editing tools to arrange sections of audio</li> <li>• Know that digital recordings need to be exported to share them</li> <li>• Know the features of a digital recording I like</li> <li>• Know and suggest improvements to a digital recording</li> </ul>
---	--	---	---

#### Progression of knowledge through the Computing Curriculum

#### Year 6

- Know how to explore a website
- Know the different types of media used on websites
- Know that websites are written in HTML
- Know the common features of a web page
- Know which media to include on my page
- Know how to draw a web page layout that suits my purpose
- Know why I should use copyright-free images
- Know how to find copyright-free images
- Know what is meant by the term 'fair use'
- Know how to add content to my own web page
- Know how to preview what my web page looks like
- Know to evaluate what my web page looks like on different devices and suggest/make edits.
- Know what a navigation path is
- Know why navigation paths are useful
- Know how to make multiple web pages and link them using hyperlinks
- Know the implication of linking to content owned by others
- Know how to create hyperlinks to link to other people's work
- Know to evaluate the user experience of a website

#### Key knowledge acquired throughout this unit

- Know that video is a visual media format
- Know features of videos
- Know how to compare features in different videos
- Know and find features on a digital video recording device
- Know how to experiment with different camera angles
- Know how to make use of a microphone
- Know and suggest filming techniques for a given purpose
- Know how to capture video using a range of filming techniques
- Know how to review how effective my video is
- Know how to outline the scenes of my video
- Know how to decide which filming techniques I will use
- Know how to create and save video content
- Know how to store, retrieve, and export my recording to a computer
- Know how to improve a video by reshooting and editing
- Know how to select the correct tools to make edits to my video
- Know how to make edits to my video and improve the final outcome
- Know that my choices when making a video will impact the quality of the final outcome
- Know how to evaluate my video and share my opinions

#### Teacher subject knowledge and guidance

This unit focuses on the skills associated with planning, recording, editing, and creating a video. You will need to be able to explain that video is the recording, reproducing, or broadcasting of moving visual images. You will also need to be familiar with a number of shot types and filming techniques, which are introduced in Lessons 2 and 3. A storyboard is used as a planning tool. It will be useful if you are familiar with the format of the storyboard.

Once learners begin filming, you will need to be familiar with the device they are using, including how to start and stop recording, how to zoom in and out, and how to download content from the device to a computer for editing.

It is important that you are familiar with the devices and apps or programs that you will use to import and edit video content. Windows 10 Video Editor is demonstrated in this unit, but there are many other free and paid-for apps available that can be used, either on a computer or on a device. Another option with limited function is to use the Microsoft Photos app.

You need to know where to locate the video files and where to save them for easy retrieval.

Once recording has been completed, learners will need to import their video files to the video editing software, so you will need to be familiar with this process, including where videos will be stored.

You will need to have a clear understanding of how to edit and complete the video creation process, deleting or reordering clips. Finally, you should be able to demonstrate how to export the video project into an \*.mp4 format for viewing.



## Technique 1: Static camera

### Key points:

- The camera is fixed in position
- A tripod or stand may be used (you can improvise)
- Examples: a newsreader or a weather forecaster

### Why is this a useful technique?

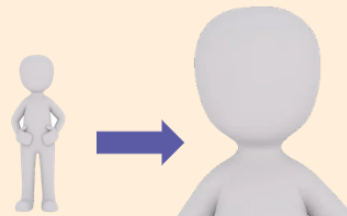


## Technique 2: Zoom

### Key points:

- This allows for close up video so you can see the subject in greater detail
- Using too much zoom may mean that the subject is blurry

### Why is this a useful technique?

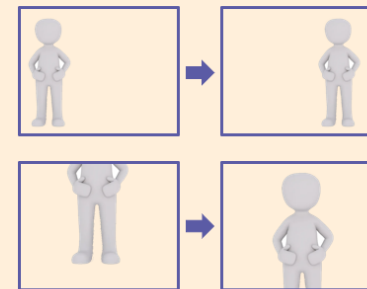


## Technique 3: Pan and tilt

### Key points:

- **Pan:** The camera position is **fixed**, but it is able to **rotate** from **side to side**
- **Tilt:** The camera position is **fixed**, but it is able to move **up** and **down**

### Why are these useful techniques?

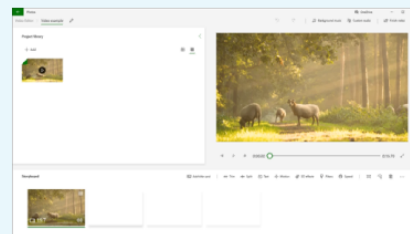


## Trimming videos

The trim tool lets you remove excess video from the beginning or end.

Right-clicking on a video thumbnail produces an additional menu.

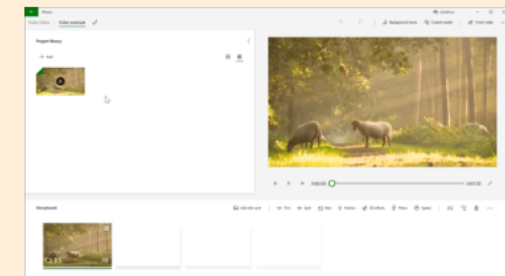
Here, you will be able to select **Split**. This will split a video into two pieces, which makes it easier to do things like removing a section you no longer want.



## Reordering videos

The video can be reordered by dragging the thumbnails to a new location in the timeline.

If you make a mistake, click the **Undo** button at the top of the screen.



This is a **talking head** section

- The camera is in a **fixed position** to keep it still
- You can see the subject's **head** and **body**
- You can hear the subject **speaking**
- You can hear **background music**



This is a **panning** section

- The camera is **hand-held**
- The camera **moves** to show different things
- You can hear the subject **speaking**
- You can hear **background music**



This is a **close up** section

- The camera is **hand-held**
- The camera is **close** to the subject
- The camera is **moved** to keep the subject in shot
- You can hear **speaking**
- You can hear **background music**



### Subject specific vocabulary and definitions (Tier 3 vocabulary)

Trim	Cut part of the video out
Export	Change the file from one type of software to another
Edit	To change or improve something
Reshoot	Take the video again
Pan	Fixed camera but can rotate from side to side
Tilt	Fixed camera but can move up and down

### Medium Term Planning

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
To explain what makes a video effective	To use a digital device to record video	To capture video using a range of techniques	To create a storyboard	To identify that video can be improved through reshooting and editing	To consider the impact of the choices made when making and sharing a video

Year 5 – Spring 1		Unit 3 – Flat File Databases	
National Curriculum objectives			
<ul style="list-style-type: none"><li>• Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content</li><li>• 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</li></ul>			
Unit Overview – Data Handling			
This unit looks at how a flat-file database can be used to organise data in records. Pupils use tools within a database to order and answer questions about data. They create graphs and charts from their data to help solve problems. They use a real-life database to answer a question and present their work to others.			
Previous knowledge acquired – Data Handling			
Year 1	Year 2	Year 3	Year 4
<ul style="list-style-type: none"><li>• Know how to describe objects using labels</li><li>• Know how to match objects to groups</li><li>• Know how to identify the label for a group of objects</li><li>• Know how to count objects</li><li>• Know how to group objects</li><li>• Know how to count a group of objects</li><li>• Know how to describe an object</li><li>• Know how to describe a property of an object</li><li>• Know how to find objects with similar properties</li><li>• Know how to group similar objects</li><li>• Know how to group objects in more than one way</li><li>• Know how to count how many objects share a property</li><li>• Know how to choose how to group objects</li><li>• Know how to describe groups of objects</li><li>• Know how to record how many objects are in a group</li><li>• Know how to decide how to group objects to answer a question</li><li>• Know how to compare groups of</li></ul>	<ul style="list-style-type: none"><li>• Know how to record data in a tally chart</li><li>• Know how to represent a tally count as a total</li><li>• Know how to compare totals in a tally chart</li><li>• Know how to enter data onto a computer</li><li>• Know how to use a computer to view data in a different format</li><li>• Know how to use pictograms to answer simple questions about objects</li><li>• Know how to organise data in a tally chart</li><li>• Know how to use a tally chart to create a pictogram</li><li>• Know what the pictogram shows</li><li>• Know how to tally objects using a common attribute</li><li>• Know how to create a pictogram to arrange objects by an attribute</li><li>• Know how to answer ‘more than’/‘less than’ and ‘most/least’ questions about an attribute</li><li>• Know how to choose a suitable attribute to compare people</li><li>• Know how to collect the data I need</li><li>• Know how to create a pictogram and draw conclusions from it</li></ul>	<ul style="list-style-type: none"><li>• Know how to investigate questions with yes/no answers.</li><li>• Know how to make up a yes/no question about a collection of objects.</li><li>• Know how to create two groups of objects separated by one attribute.</li><li>• Know select an attribute to separate objects into groups.</li><li>• Know how to create a group of objects within an existing group.</li><li>• Know how to arrange objects into a tree structure.</li><li>• I know how to select objects to arrange in a branching database.</li><li>• Know how to group objects using my own yes/no questions.</li><li>• Know how to prove my branching database works.</li><li>• Know how to create yes/no questions using given attributes.</li><li>• Know how to explain that questions need to be ordered carefully to split objects into similarly sized groups.</li><li>• Know hot to compare two branching database structures.</li><li>• Know how to select a theme and choose a variety of objects.</li><li>• Know how to create questions and apply them to a tree structure.</li></ul>	<ul style="list-style-type: none"><li>• Know how to choose a data set to answer a given question</li><li>• Know how to suggest questions that can be answered using a given data set</li><li>• Know that data gathered over time can be used to answer questions</li><li>• Know data that can be gathered over time</li><li>• Know that sensors are input devices</li><li>• Know how to use data from a sensor to answer a given question</li><li>• Know that data from sensors can be recorded</li><li>• Know a suitable place to collect data</li><li>• Know the intervals used to collect data</li><li>• Know about the data that I have captured</li><li>• Know how to import a data set</li><li>• Know how to use a computer to view data in different ways</li><li>• Know how to use a computer program to sort data</li><li>• Know to propose a question that can be answered using logged data</li><li>• Know how to collect data using a data logger</li><li>• Know how to interpret data that has been collected using a data logger</li><li>• Know how to draw conclusions from the data that I have collected</li><li>• Know the benefits of using a data logger</li></ul>

<ul style="list-style-type: none"> <li>objects</li> <li>Know how to record and share what I have found</li> </ul>	<ul style="list-style-type: none"> <li>Know how to use a computer program to present information in different ways</li> <li>Know to share what I have found out using a computer</li> <li>Know to give simple examples of why information should not be shared</li> </ul>	<ul style="list-style-type: none"> <li>Know how to use my branching database to answer questions.</li> <li>I know how to explain what a pictogram tells me.</li> <li>Know how to explain what a branching database tells me.</li> <li>Know how to compare two ways of presenting information.</li> <li></li> </ul>	
---	---	--	--

#### Progression of knowledge through the Computing Curriculum

#### Year 6

- Know and explain the relevance of data headings
- Know how to answer questions from an existing data set
- Know to ask simple relevant questions which can be answered using data
- Know what an item of data is
- Know how to apply an appropriate number format to a cell
- Know how to build a data set in a spreadsheet application
- Know and explain the relevance of a cell's data type
- Know how to construct a formula in a spreadsheet
- Know that changing inputs changes outputs
- Know that data can be calculated using different operations
- Know how to create a formula which includes a range of cells
- Know how to apply a formula to multiple cells by duplicating it
- Know how to use a spreadsheet to answer questions
- Know and explain why data should be organised
- Know and apply a formula to calculate the data I need to answer questions
- Know how to produce a graph
- Know how to use a graph to show the answer to questions
- Know when to use a table or graph

#### Key knowledge acquired throughout this unit

- Know how to create multiple questions about the same field
- Know how information can be recorded
- Know how to order, sort, and group my data cards
- Know how to navigate a flat-file database to compare different views of information
- Know what a 'field' and a 'record' is in a database
- Know which field to sort data by to answer a given question
- Know how information can be grouped
- Know how to group information to answer questions
- Know how to combine grouping and sorting to answer more specific questions

- Know which field and value are required to answer a given question
- Know how 'AND' and 'OR' can be used to refine data selection
- Know to choose multiple criteria to answer a given question
- Know to an appropriate chart to visually compare data
- Know to refine a chart by selecting a particular filter
- Know the benefits of using a computer to create graphs
- Know to ask questions that will need more than one field to answer
- Know how to refine a search in a real-world context

## Teacher subject knowledge and guidance

Teachers will need to know that a flat-file database is a collection of data organised in a single table. The term 'database' means 'a collection of organised data that is stored on a computer'. Databases allow people to search and sort large quantities of data to find information. Data can be letters, words, numbers, dates, images, sounds etc. In addition, teachers will need to be familiar with the basic structure of a database, and the concept of 'grouping' and 'sorting' data records based on different fields. For example, grouping objects by colour, or sorting into alphabetical order.

A database is composed of 'records', which are sets of data on a particular object. Records are formed from one or more 'fields' of data. A field is one specific piece of data in a database record. For example, a record all about a country could have fields such as 'country name' and 'country population'. The value within the record is the 'answer' to each field, eg Mexico is the value in the 'country name' field and '126.2 million' is the value in the 'country population' field.

Teachers will also need to be aware that all objects have attributes. An attribute includes its 'name' and a 'value'. For example, a ball will have a 'colour', which might be 'red'. 'Colour' is the attribute 'name', 'red' is the attribute 'value'. In a flat-file database the attribute names become the fields when the data about the object is stored as a record. The values of the attributes become the values that are saved in the database fields.


Teachers will need to be familiar with using j2data sample databases. Support with navigating the databases can be found at <http://www.j2e.com/help/videos/datags4>. Knowledge of how to carry out a flight search using <https://www.expendia.co.uk/Flights>, and the ability to screenshot flight details from a web browser, would also be beneficial.

'A database is a collection of data that is stored in a computer and that can easily be used and added to'

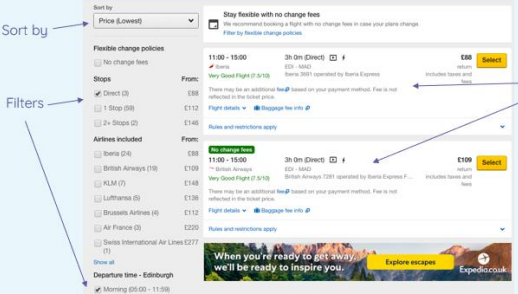
### Parts of a record

fields

- Country Name
- Area
- Population
- Language
- Flag
- Currency

fields	values	data type
Country Name	South Africa	This data is text
Area	1,221,037 km <sup>2</sup>	This data is numeric
Population	54956900	This data is numeric
Language	Afrikaans	This data is text
Flag		This data is an image
Currency	South African Rand	This data is text

### How can we search a flight database?



## Subject specific vocabulary and definitions (Tier 3 vocabulary)

Flat file database	A collection of data organised in a single table
Field	One specific piece of data in a database
Record	Sets of data on a particular object

Refine	Improve something by making small changes				
Data	A collection of organised data that is stored on a computer				
Medium Term Planning					
Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
To use a form to record information	To compare paper and computer-based databases	To outline how grouping and then sorting data allows us to answer questions	To explain that tools can be used to select specific data	To explain that computer programs can be used to compare data visually	To apply my knowledge of a database to ask and answer real-world questions

Year 5 – Spring 2		Unit 4 – Vector Drawing	
National Curriculum objectives			
<ul style="list-style-type: none"><li>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</li></ul>			
Unit Overview			
<p>In this unit, learners start to create vector drawings. They learn how to use different drawing tools to help them create images. Learners recognise that images in vector drawings are created using shapes and lines, and each individual element in the drawing is called an object. Learners layer their objects and begin grouping and duplicating them to support the creation of more complex pieces of work.</p> <p>Note: This unit is planned to use the Google Drawings application. All Google Drawings files are provided in the lesson plans (the links create a copy of the drawing that can be edited). However, if you wish to use a different vector drawing program, all the resources are included in the folders as Google Slides presentations. The contents of these files can be copied across to your preferred program.</p> <p>To demonstrate the tools and skills involved in vector drawings, it is recommended that you use a vector drawing program, such as Vector. While the resources could be used in the Google Slides files, it is important that learners recognise that true vector drawings are made using a vector drawing program.</p> <p>This unit progresses learners’ knowledge and understanding of digital painting and has some links to the Year 3 ‘Creating media – Desktop publishing’ unit, in which learners used digital images. In this Year 5 unit, learners create images that could be used in desktop publishing documents.</p>			
Previous knowledge acquired – Digital Literacy			
Year 1	Year 2	Year 3	Year 4
<ul style="list-style-type: none"><li>Know how to open a word processor</li><li>Know keys on a keyboard</li><li>Know how to enter text into a computer</li><li>Know how to use letter, number, and space keys</li><li>Know how to use backspace to remove text</li><li>Know how to type capital letters</li><li>Know where the toolbar is and use bold, italic, and underline</li><li>Know how to select a word by double-clicking</li><li>Know how to select all of the text by clicking and dragging</li><li>Know how to change the font</li><li>Know and say what tool I used to change the text</li></ul>	<ul style="list-style-type: none"><li>Know and identify simple differences in pieces of music</li><li>Know to listen with concentration to a range of music (links to the Music curriculum)</li><li>Know how music makes me feel, e.g., happy or sad</li><li>Know how to create a rhythm pattern</li><li>Know how to play an instrument following a rhythm pattern</li><li>Know how to explain that music is created and played by humans</li><li>Know how to connect images with sounds</li><li>Know how to use a computer to experiment with pitch and duration</li><li>Know to relate an idea to a piece of music</li></ul>	<ul style="list-style-type: none"><li>Know the difference between text and images.</li><li>Know that text and images can communicate messages clearly.</li><li>Know I can identify the advantages and disadvantages of using text and images.</li><li>Know how to change font style, size, and colours for a given purpose.</li><li>Know how to edit text.</li><li>Know how to explain that text can be changed to communicate more clearly.</li><li>Know and can explain what ‘page orientation’ means.</li><li>Know how to recognise placeholders and say why they are important.</li><li>Know how to create a template for a particular purpose.</li><li>Know how to add content to a desktop publishing publication.</li><li>Know how to choose the best locations for my content.</li></ul>	<ul style="list-style-type: none"><li>Know the changes that we can make to an image</li><li>Know how images can be changed in real life</li><li>Know the effect that editing can have on an image</li><li>Know what has changed in an edited image</li><li>Know how to change the composition of an image by selecting parts of it</li><li>Know why someone might want to change the composition of an image</li><li>Know and talk about changes made to images</li><li>Know how to use effects to make my image fit a scenario</li><li>Know why my choices fit a scenario</li><li>Know how an image has been retouched</li><li>Know examples of positive and negative effects that retouching can have on an image</li><li>Know appropriate tools to retouch an image</li><li>Know how to sort images into ‘fake’ or ‘real’ and explain my choices</li><li>Know how to combine parts of images to create</li></ul>



<ul style="list-style-type: none"> <li>• Know if my changes have improved my writing</li> <li>• Know how to use 'undo' to remove changes</li> <li>• Know how to write a message on a computer and on paper</li> <li>• Compare using a computer with using a pencil and paper</li> <li>• Know which method I like best</li> </ul>	<ul style="list-style-type: none"> <li>• Know that music is a sequence of notes</li> <li>• Know how to use a computer to create a musical pattern using three notes</li> <li>• Know how to describe an animal using sounds</li> <li>• Know to explain my choices</li> <li>• Know how to save my work</li> <li>• Know to refine my musical pattern on a computer</li> <li>• Know how to reopen my work</li> <li>• Know to explain how I made my work better</li> <li>• Know to listen to music and describe how it makes me feel</li> </ul>	<ul style="list-style-type: none"> <li>• Know how to paste text and images to create a magazine cover.</li> <li>• Know how to make changes to content after I've added it.</li> <li>• Know how different layouts can suit different purposes</li> <li>• Know how to identify different layouts.</li> <li>• Know how to match a layout to a purpose.</li> <li>• Know how to choose a suitable layout for a given purpose.</li> <li>• Know the benefits of desktop publishing</li> <li>• Know how to identify the uses of desktop publishing in the real world.</li> <li>• Know how to say why desktop publishing might be helpful.</li> <li>• Know how to compare work made on desktop publishing to work created by hand.</li> </ul>	<p>new images</p> <ul style="list-style-type: none"> <li>• Know the effect of adding other elements to my work</li> <li>• Know to compare the original image with my completed publication</li> <li>• Know to evaluate the impact of my publication on others through feedback</li> </ul>
--	--	--	---

#### Progression of knowledge through the Computing Curriculum

#### Year 6

- Know the similarities and differences between 2D and 3D shapes
- Know why we might represent 3D objects on a computer
- Know how to select, move, and delete a digital 3D shape
- Know how graphical objects can be modified
- Know how to resize a 3D object
- Know how to change the colour of a 3D object
- Know how to rotate a 3D object
- Know how to position 3D objects in relation to each other
- Know how to select and duplicate multiple 3D objects
- Know how to identify the 3D shapes needed to create a model of a real-world object
- Know how to create digital 3D objects of an appropriate size
- Know how to group a digital 3D shape and a placeholder to create a hole in an object
- Know to plan my 3D model
- Know which 3D objects I need to construct my model
- Know how to modify multiple 3D objects
- Know how my model can be improved
- Know to modify my model to improve it
- Know to evaluate my model against a given criterion



### Key knowledge acquired throughout this unit

- Know that vector drawings are made using shapes
- Know how to experiment with the shape and line tools
- Know how vector drawings are different from paper-based drawings
- Know the shapes used to make a vector drawing
- Know that each element added to a vector drawing is an object
- Know how to move, resize, and rotate objects I have duplicated
- Know how to use the zoom tool to help me add detail to my drawings
- Know how alignment grids and resize handles can be used to improve consistency
- Know how to modify objects to create a new image
- Know that each added object creates a new layer in the drawing
- Know how to change the order of layers in a vector drawing
- Know how to use layering to create an image
- Know how to copy part of a drawing by duplicating several objects
- Know how to recognise when I need to group and ungroup objects
- Know how to reuse a group of objects to further develop my vector drawing
- Know how to create a vector drawing for a specific purpose
- Know to reflect on the skills I have used and why I have used them
- Know how to compare vector drawings to freehand paint drawings

### Teacher subject knowledge and guidance

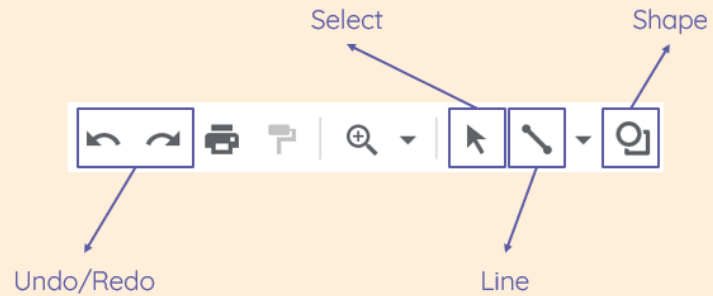
Teachers need a good understanding of the tools available in their chosen software. These tools include shape and line drawing tools, fill, undo and redo, select, and delete. In the unit, learners also need to move, resize, and rotate shapes, including the use of size and alignment guides. Many of these tools are available in presentation software such as Google Slides and Microsoft PowerPoint.

Teachers need to know that vector drawings are created using shapes and lines. In vector drawings, all these shapes and lines are called objects. Within vector drawings, each object is created using a new layer. These can be rearranged using the menu tool, which allows the objects to be sent backwards and forwards (one at a time) or sent to the back or front. Objects within these programs can be grouped, which enables the objects to be treated as though they are a single object. It is important to recognise that although they act as though they are a single object, grouped objects are still a number of individual objects.

Teachers need to have an understanding of how digital images can be made. This could be using shapes and lines in a vector drawing, or using pixels in a freehand paint program.

## A vector drawing is...

- Made on the computer
- Made using lines and shapes
- Made by putting the lines and shapes together to form a complete image



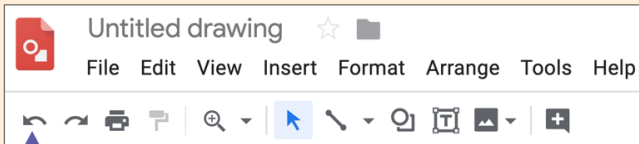
## Copy



## Paste



## Undo



Click to undo or use the keyboard shortcut:

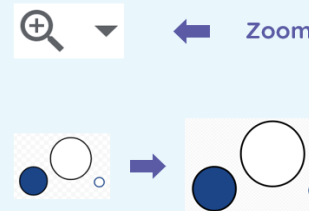


## Getting a better view

When objects are small, they can be hard to click on to move and resize.

To get a better view, you can use the zoom tool.

- The zoom tool does not change the size of the object
- Click the **Zoom** button and then click on the area where you want to see more clearly

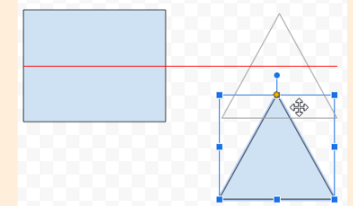


## Alignment guides

Did you notice the red lines?

These red lines are used for alignment:

- You can align them with the centre of another object
- You can align them with the edge of another object

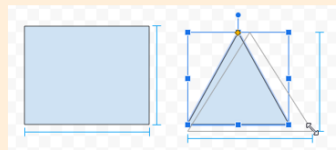


## Size guides

These blue lines are used as a size guide.

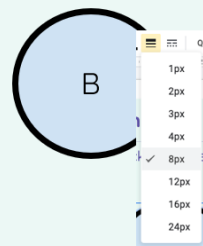
As you resize objects, you can use these lines to match them with other objects on the canvas:

- You can match the width and/or height with another object

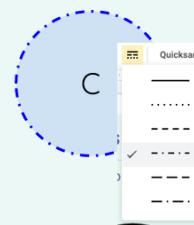


## Border effects

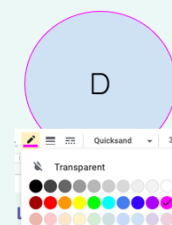
Border weight



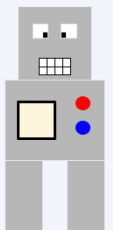
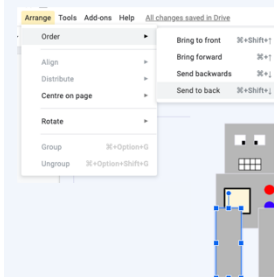
Border dash



Border colour



## How would you fix this drawing?





Remember to group the drawing again after you've made a change. Otherwise the next time you move it, you might lose a part of it!

If you select **Rotate**, there is the option to flip a drawing. Flipping an image is like looking in a mirror. It can make the object look the other way!

### Subject specific vocabulary and definitions (Tier 3 vocabulary)

Layering	A layer that allows you to edit lines that have already been drawn
Duplicating	Make an exact copy
Alignment grids	Guides that help you line objects up
Resize	Make a shape bigger or smaller
Rotate	Turn a shape or object clockwise or anticlockwise

### Medium Term Planning

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
To identify that drawing tools can be used to produce different outcomes	To create a vector drawing by combining shapes	To use tools to achieve a desired effect	To recognise that vector drawings consist of layers	To group objects to make them easier to work with	To apply what I have learned about vector drawings

Year 5 – Summer 1		Unit 5 – Selection in Physical Computing	
National Curriculum objectives			
<ul style="list-style-type: none"><li>• Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts</li><li>• Use sequence, selection, and repetition in programs; work with variables and various forms of input and output</li><li>• Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs</li><li>• 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</li></ul>			
Cross Curricular Links			
Science – Electricity (Year 4) <ul style="list-style-type: none"><li>• construct a simple series electrical circuit, identifying and naming its basic parts, including cells, wires, bulbs, switches and buzzers</li></ul>			
Unit Overview			
<p>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.</p> <p>There are two Year 5 programming units:</p> <ul style="list-style-type: none"><li>• Programming A – Selection in physical computing</li><li>• Programming B – Selection in quizzes</li></ul> <p>This is unit A, which should be delivered before unit B.</p> <p>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.</p>			
Previous knowledge acquired – Programming			
Year 1	Year 2	Year 3	Year 4
<ul style="list-style-type: none"><li>• I can predict the outcome of a command on a device</li><li>• Know how to match a command to an outcome</li><li>• Know how to run a command on a device</li><li>• Know how to follow an instruction</li><li>• Know how to recall words that can be acted out</li><li>• Know how to give directions</li><li>• Know how to compare forwards</li></ul>	<ul style="list-style-type: none"><li>• I can predict the outcome of a command on a device</li><li>• Know how to match a command to an outcome</li><li>• Know how to run a command on a device</li><li>• Know how to follow an instruction</li><li>• Know how to recall words that can be acted out</li><li>• Know how to give directions</li></ul>	<ul style="list-style-type: none"><li>• Know all the objects in a Scratch project (sprites, backdrops)</li><li>• Know that objects in Scratch have attributes (linked to)</li><li>• Know that commands in Scratch are represented as blocks</li><li>• Know that each sprite is controlled by the commands I choose</li><li>• Know a word which describes an on-screen action for my design</li></ul>	<ul style="list-style-type: none"><li>• Know how to program a computer by typing commands</li><li>• Know the effect of changing a value of a command</li><li>• Know how to create a code snippet for a given purpose</li><li>• Know how to use a template to draw what I want my program to do</li><li>• Know how to write an algorithm to produce a given outcome</li></ul>

<ul style="list-style-type: none"> <li>• and backwards movements</li> <li>• Know to start a sequence from the same place</li> <li>• I can predict the outcome of a sequence involving forwards and backwards commands</li> <li>• Know to compare left and right turns</li> <li>• Know how to experiment with turn and move commands to move a robot</li> <li>• I can predict the outcome of a sequence involving up to four commands</li> <li>• Know how to what my program should do</li> <li>• Know how to choose the order of commands in a sequence</li> <li>• Know how to debug my program</li> <li>• Know to identify several possible solutions</li> <li>• Know how to plan two programs</li> <li>• Know how to use two different programs to get to the same place</li> </ul>	<ul style="list-style-type: none"> <li>• Know how to compare forwards and backwards movements</li> <li>• Know to start a sequence from the same place</li> <li>• I can predict the outcome of a sequence involving forwards and backwards commands</li> <li>• Know to compare left and right turns</li> <li>• Know how to experiment with turn and move commands to move a robot</li> <li>• I can predict the outcome of a sequence involving up to four commands</li> <li>• Know how to what my program should do</li> <li>• Know how to choose the order of commands in a sequence</li> <li>• Know how to debug my program</li> <li>• Know to identify several possible solutions</li> <li>• Know how to plan two programs</li> <li>• Know how to use two different programs to get to the same place</li> </ul>	<ul style="list-style-type: none"> <li>• Know how to create a program following a design</li> <li>• Know how to start a program in different ways</li> <li>• Know how to create a sequence of connected commands</li> <li>• Know how to explain that the objects in my project will respond exactly to the code</li> <li>• Know how to explain what a sequence is</li> <li>• Know how to combine sound commands</li> <li>• Know how to order notes into a sequence</li> <li>• Know how to build a sequence of commands</li> <li>• Know how to decide the actions for each sprite in a program</li> <li>• Know how to make design choices for my artwork</li> <li>• Know the names of the objects I will need for a project</li> <li>• Know how to relate a task description to a design</li> <li>• Know how to implement my algorithm as code</li> </ul>	<ul style="list-style-type: none"> <li>• Know how to test my algorithm in a text-based language</li> <li>• Know examples of repetition in everyday tasks</li> <li>• Know how to identify patterns in a sequence</li> <li>• Know how to use a count-controlled loop to produce a given outcome</li> <li>• Know the effect of changing the number of times a task is repeated</li> <li>• Know to predict the outcome of a program containing a count-controlled loop</li> <li>• Know how to choose which values to change in a loop</li> <li>• Know where there are 'chunks' of actions in the real world</li> <li>• Know how to use a procedure in a program</li> <li>• Know that a computer can repeatedly call a procedure</li> <li>• Know how to design a program that includes count-controlled loops</li> <li>• Know how to make use of my design to write a program</li> <li>• Know how to develop my program by debugging it</li> </ul>
---	--	--	---

#### Progression of knowledge through the Computing Curriculum

#### Year 6

- Know examples of information that is variable
- Know that the way that a variable changes can be defined
- Know that variables can hold numbers or letters
- Know to identify a program variable as a placeholder in memory for a single value
- Know that a variable has a name and a value
- Know that the value of a variable can be changed
- Know where in a program to change a variable
- Know to make use of an event in a program to set a variable
- Know that the value of a variable can be used by a program
- Know how to choose the artwork for my project

- Know to explain my design choices
- Know how to create algorithms for my project
- Know how to create the artwork for my project
- Know to choose a name that identifies the role of a variable
- Know how to test the code that I have written
- Know ways that my game could be improved
- Know how to extend my game further using more variables
- Know how to share my game with others

#### Key knowledge acquired throughout this unit

- Know how to build a simple circuit to connect a microcontroller to a computer
- Know how to program a microcontroller to light an LED
- Know and explain why I used an infinite loop
- Know how to connect more than one output device to a microcontroller
- Know how to design sequences for given output devices
- Know which output devices I control with a count-controlled loop
- Know 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)
- Know to experiment with a 'do until' loop
- Know how to program a microcontroller to respond to an input
- Know that a condition being met can start an action
- Know how to identify a condition and an action in my project
- Know how to use selection (an 'if... then...' statement) to direct the flow of a program
- Know how to identify a condition to start an action (real world)
- Know and describe what my project will do (the task)
- Know how to create a detailed drawing of my project
- Know how to write an algorithm to control lights and a motor
- Know to use selection to produce an intended outcome
- Know how to test and debug my project

#### Teacher subject knowledge and guidance

## 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](http://redfernelectronics.co.uk/crumble-software).

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](http://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.



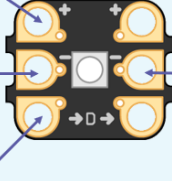
## Connecting Sparkle

Connects to a positive power (+) pad on the Crumble controller

Connects to a negative power (-) pad on the Crumble controller

The pads on this side are used to connect other Sparkles

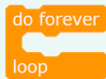
Connects to the D pad on the Crumble controller



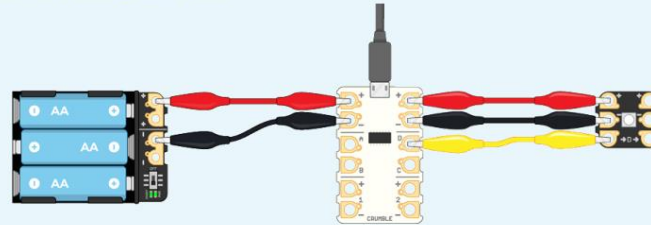
## Forever flashing

You might want to repeat some or all of the commands in your program. You can do this using a repeat block.

This block repeats the commands inside it forever.



## Connecting circuits



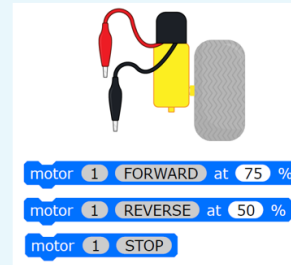
Connect your Crumble using the guide above. The Sparkle will flash white six times when you've connected it correctly.

## Controlling outputs

You are going to connect a geared motor to your Crumble.

A motor can:

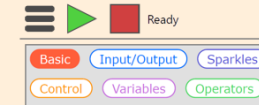
- Spin forwards
- Spin backwards
- Go at different speeds
- Stop



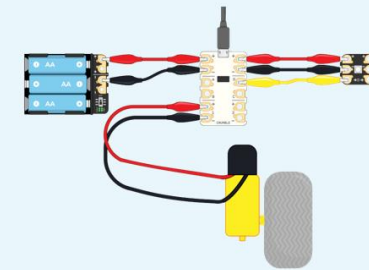
## Programming Crumbles

Create this program in the Crumble software.

After you've checked your Crumble is connected to your computer, press the green play button.



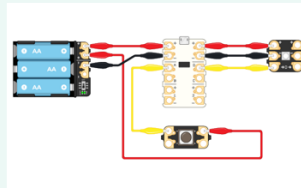
## Connecting a motor and a Sparkle



## Using inputs

A push switch can be used as an input for a Crumble controller, and can therefore be used in condition statements.

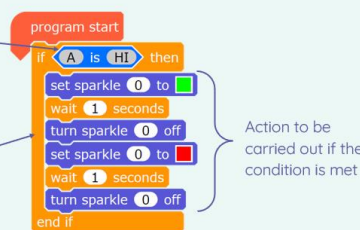
Use the diagram to connect the switch to your Crumble controller.



## Selection in programs

Condition — if the push button is pressed

Selection command block

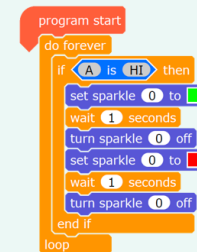


Action to be carried out if the condition is met

## Selection in programs

When using selection in programs, you must instruct the device to check constantly if the condition has been met.

To do this, you use repetition in the form of an infinite loop. If you do not use repetition, the device will only check once if the condition has been met.



## Subject specific vocabulary and definitions (Tier 3 vocabulary)

Algorithm	A precise set of ordered steps that can be followed by a human or a computer to achieve a task
Selection	When the program decides to move on based on the results of an event
Output device	A piece of hardware that is controlled by outputs from a computer
Condition	Something that can be either true or false
Infinite loop	A command that repeatedly runs a section of code indefinitely
Count controlled loop	A command that repeatedly runs a section of code a certain number of times

Condition controlled loop	A command that repeatedly runs a section of code until a condition is met				
Debug	The process of finding and correcting errors in a program				
Medium Term Planning					
Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
To control a simple circuit connected to a computer	To write a program that includes count-controlled loops	To explain that a loop can stop when a condition is met, eg number of times	To conclude that a loop can be used to repeatedly check whether a condition has been met	To design a physical project that includes selection	To create a controllable system that includes selection

Year 5 – Summer 2		Unit 6 – Selection in quizzes	
National Curriculum objectives			
<ul style="list-style-type: none"><li>• Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts</li><li>• Use sequence, selection, and repetition in programs; work with variables and various forms of input and output</li><li>• Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs</li><li>• 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</li></ul>			
Unit Overview			
<p>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.</p> <p>There are two Year 5 programming units:</p> <ul style="list-style-type: none"><li>• Programming A – Selection in physical computing</li><li>• Programming B – Selection in quizzes</li></ul> <p>This is unit B, which should be delivered after unit A.</p>			
Previous knowledge acquired - Programming			
Year 1	Year 2	Year 3	Year 4
<ul style="list-style-type: none"><li>• I can predict the outcome of a command on a device</li><li>• Know how to match a command to an outcome</li><li>• Know how to run a command on a device</li><li>• Know how to follow an instruction</li><li>• Know how to recall words that can be acted out</li><li>• Know how to give directions</li><li>• Know how to compare forwards and backwards movements</li><li>• Know to start a sequence from the same place</li><li>• I can predict the outcome of a sequence involving forwards and backwards commands</li><li>• Know to compare left and right turns</li></ul>	<ul style="list-style-type: none"><li>• Know where the start of a sequence is</li><li>• Know that a program needs to be started</li><li>• Know how to run my program</li><li>• Know to predict the outcome of a sequence of commands</li><li>• Know how to match two sequences with the same outcome</li><li>• Know how to change the outcome of a sequence of commands</li><li>• Know how to work out the actions of a sprite in an algorithm</li><li>• Know which blocks to use to meet the design</li><li>• Know how to build the sequences of blocks I need</li><li>• Know how to choose backgrounds for the design</li><li>• Know how to choose characters for the</li></ul>	<ul style="list-style-type: none"><li>• Know the relationship between an event and an action</li><li>• Know which keys to use for actions and explain my choices</li><li>• Know a way to improve a program</li><li>• Know a suitable size for a character in a maze</li><li>• Know how to program movement</li><li>• Know how to use a programming extension</li><li>• Know to consider the real world when making design choices</li><li>• Know which blocks to use to set up my program</li><li>• Know additional features (from a given set of blocks)</li><li>• Know suitable keys to turn on additional features</li><li>• Know how to build more sequences of commands to make my design work</li><li>• Know how to test a program against a given</li></ul>	<ul style="list-style-type: none"><li>• Know an everyday task as a set of instructions including repetition</li><li>• Know to predict the outcome of a snippet of code</li><li>• Know how to modify a snippet of code to create a given outcome</li><li>• Know how to modify loops to produce a given outcome</li><li>• Know when to use a count-controlled and an infinite loop</li><li>• Know that some programming languages enable more than one process to be run at once</li><li>• Know which action will be repeated for each object</li><li>• Know what the outcome of the repeated action should be</li></ul>

<ul style="list-style-type: none"> <li>• Know how to experiment with turn and move commands to move a robot</li> <li>• I can predict the outcome of a sequence involving up to four commands</li> <li>• Know how to what my program should do</li> <li>• Know how to choose the order of commands in a sequence</li> <li>• Know how to debug my program</li> <li>• Know to identify several possible solutions</li> <li>• Know how to plan two programs</li> <li>• Know how to use two different programs to get to the same place</li> </ul>	<ul style="list-style-type: none"> <li>• design</li> <li>• Know how to create a program based on the new design</li> <li>• Know how to choose the images for my own design</li> <li>• Know how to create an algorithm</li> <li>• Know how to build sequences of blocks to match my design</li> <li>• Know how to compare my project to my design</li> <li>• Know how to improve my project by adding features</li> <li>• Know how to debug</li> </ul>	<ul style="list-style-type: none"> <li>• design</li> <li>• Know how to match a piece of code to an outcome</li> <li>• Know how to modify a program using a design</li> <li>• Know to make design choices and justify them</li> <li>• Know how to implement my design</li> <li>• Know to evaluate my project</li> </ul>	<ul style="list-style-type: none"> <li>• Know to evaluate the effectiveness of the repeated sequences used in my program</li> <li>• Know which parts of a loop can be changed</li> <li>• Know the effect of my changes</li> <li>• Know to re-use existing code snippets on new sprites</li> <li>• Know to evaluate the use of repetition in a project</li> <li>• Know to select key parts of a given project to use in my own design</li> <li>• Know how to develop my own design explaining what my project will do</li> <li>• Know to refine the algorithm in my design</li> <li>• Know to build a program that follows my design</li> <li>• Know to evaluate the steps I followed when building my project</li> </ul>
---	---	--	--

#### Progression of knowledge through the Computing Curriculum

#### Year 6

- Know how to apply my knowledge of programming to a new environment
- Know how to test my program on an emulator
- Know how to transfer my program to a controllable device
- Know how to identify examples of conditions in the real world
- Know to use a variable in an if, then, else statement to select the flow of a program
- Know how to determine the flow of a program using selection
- Know to use a condition to change a variable
- Know how to experiment with different physical inputs
- Know that if you read a variable, the value remains
- Know the importance of the order of conditions in else, if statements
- Know how to use an operand (e.g. <=>) in an if, then statement
- Know how to modify a program to achieve a different outcome
- Know what variables to include in a project
- Know how to design the algorithm for my project
- Know how to design the program flow for my project
- Know how to create a program based on my design
- Know how to test my program against my design
- Know to use a range of approaches to find and fix bugs

#### Key knowledge acquired throughout this unit

- Know how conditions are used in selection
- Know how to identify conditions in a program
- Know how to modify a condition in a program
- Know how to use selection in an infinite loop to check a condition
- Know to identify the condition and outcomes in an 'if... then... else...' statement
- Know how to create a program with different outcomes using selection
- Know that program flow can branch according to a condition
- Know how to design the flow of a program which contains 'if... then... else...'
- Know that a condition can direct program flow in one of two ways
- Know how to outline a given task
- Know how to use a design format to outline my project
- Know how to identify the outcome of user input in an algorithm
- Know how to implement my algorithm to create the first section of my program
- Know how to test my program
- Know how to share my program with others
- Know to identify ways the program could be improved
- Know how to identify the setup code I need in my program
- Know how to extend my program further

#### Teacher subject knowledge and guidance

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'.

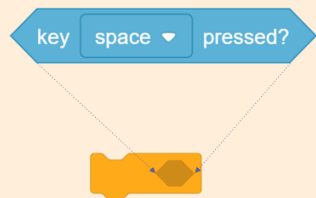
##### Selection

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.

### Conditions in programs

This **condition block** needs to be used with other blocks to control the flow of actions.



**Condition blocks** are all the same shape (a hexagon). The blocks they are used with have a hexagonal space in them.

### Conditions in programs

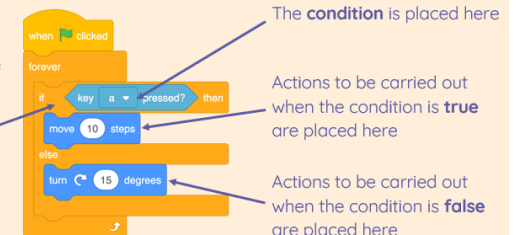
**Condition:** if space bar is pressed

**Action:** to be carried out when the condition is true



### Different outcomes — code solution

To write programs that use selection with two outcomes, we need to use an **if... then... else...** orange **control block**



The **condition** is placed here

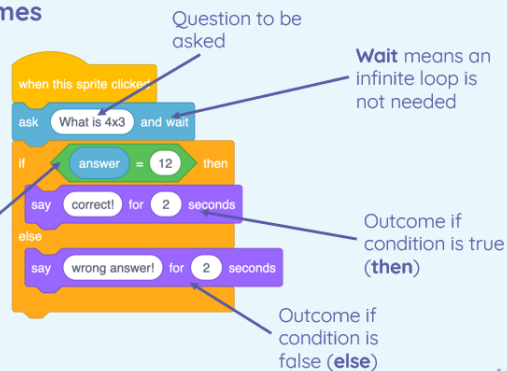
Actions to be carried out when the condition is **true** are placed here

Actions to be carried out when the condition is **false** are placed here

### Identifying outcomes

- Where is the:
- Condition
  - Outcome if the condition is true
  - Outcome if the condition is false

Condition — the answer is the same as '12'



Question to be asked

**Wait** means an infinite loop is not needed

Outcome if condition is true (**then**)

Outcome if condition is false (**else**)

### Subject specific vocabulary and definitions (Tier 3 vocabulary)

Condition	Something that can be either true or false
Selection	When the program decides to move on based on the results of an event
Program	A set of ordered commands that can be run by a computer to complete a task
Repetition	When something happens over and over again
Outcome	End result of the program
Code	The commands that a computer can run

### Medium Term Planning

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
--------	--------	--------	--------	--------	--------

To explain how selection is used in computer programs	To relate that a conditional statement connects a condition to an outcome	To explain how selection directs the flow of a program	To design a program which uses selection	To create a program which uses selection	To evaluate my program
---	---	--	--	--	------------------------