Princess May Primary School

Computing Curriculum Knowledge and Skills Progression Map 2026–27

A Spiral Curriculum

Our Computing curriculum has been designed as a spiral curriculum with the following key principles in mind:

✓ **Cyclical:** Pupils return to the same skills again and again during their time in primary school.
✓ **Increasing depth:** Each time a skill is revisited it is covered with greater complexity
✓ **Prior knowledge:** Upon returning to a skill, prior knowledge is utilised so pupils can build upon previous foundations, rather than starting again.

Our curriculum aims

At Princess May, we aim for all our pupils to develop our school values of respect, empathy and determination. Our Computing scheme of work aims to inspire pupils to become curious and explorative thinkers with a diverse knowledge of the world – in other words, to think like computing experts. The intention is for pupils to develop the confidence to question and observe places, measure and record necessary data in a variety of ways and analyse and present their findings. We wish for them to contribute thoughtfully to the future, respectfully, empathically and with determination.

Knowledge developed within our curriculum

Digital literacy
Recognising the opportunities and risks of the online world and develop the skills to use technology safely, respectfully and responsibly.

Information technology
Using technology purposefully to create, organise, store, manipulate and present digital content across a range of contexts.

Computer science
Developing an understanding of how digital systems work and how to design, write and debug programs to control them.

EYFS Framework

ELG: Fine Motor Skills
ELG: Creating with Materials

Children at the expected level of development will:
• Safely use and explore a variety of materials, tools and techniques, experimenting with function.
• Share their creations, explaining the process they have used.

National Curriculum Programme of Study
A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science, and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world. At Princess May, the National Curriculum underpins our curriculum.

Progression of skills and knowledge

Computing systems and networks

	Identifying and operating hardware	Networks	Data storage	Digital security
EYFS	• **Hardware** • Learning how to operate a camera to take photographs of meaningful creations or moments. • Learning how to explore and tinker with hardware to develop familiarity and introduce relevant vocabulary. • Recognising and identifying familiar letters and numbers on a keyboard. • Developing basic mouse skills such as moving and clicking. • **Digital literacy** • Recognising that a range of technology is used for different purposes. • Learning to log in and log out. • **Knowledge** • To be able to understand what a computer keyboard is and recognising some letters and numbers. • To know that a mouse can be used to click, drag and create simple drawings. • To know that to use a computer you need to log in to it and then log out at the end of your session. • To know that different types of technology can be found at home and in school. • To know that you can take simple photographs with a camera or iPad. • To know that you must hold the camera still and ensure the subject is in the shot to take a photo. •			
Year 1	**Skills** • Using a mouse, trackpad and/or touchscreen to interact with digital devices. • Using a keyboard to interact with a device. • Describe where technology is used at home (e.g., playing games, watching videos). **Knowledge** • To know the names of common digital	**Skills** • Browsing and interacting with pre-selected websites, software or videos. **Knowledge** • Browsing and interacting with pre-selected websites, software or videos.	**Skills** • Create and name a file with guidance. • Save and retrieve their work with guidance. **Knowledge** • Naming and saving files so they can be found easily. • Recognising simple icons associated with saving their work.	**Skills** • Using a password to login and logout of a common platform (this could be a visual password or with support). • Recognising a login screen. **Knowledge** • To know that people should have different passwords. • To know that passwords keep things safe.

	devices (e.g., computer, tablet, camera). • To know the names of the parts of the device (e.g., screen, keyboard, mouse).			
Year 2	**Skills** • Using a mouse, trackpad and/or touchscreen for a specific purpose. • Using a keyboard for a specific purpose. • Using inputs to create outputs. • Recognising that different devices have different parts. • Identifying common uses of IT in the wider world (e.g., shops, self-service screens, traffic lights). **Knowledge** • To know that button presses and movements of the mouse, trackpad or on the touchscreen are inputs. • To know that the screen or speaker produces an output. • To know that IT is a part of our everyday life and the world we live in.	**Skills** • Following links on websites to get to other parts of a page. • Beginning to recognise a search tool. **Knowledge** • To know that some parts of websites are linked together. • To know that there are visual clues on websites to show the links.	**Skills** • Naming and saving files so they can be found easily. • Recognising simple icons associated with saving their work. **Knowledge** • To know that their work can be organised into folders so that they can find it easily. • To know that icons are used across digital devices.	**Skills** • Describing why passwords are important. • Beginning to use username and password independently to log in and out of a common platform. **Knowledge** • To know that strong passwords can't be easily guessed by others. • To know that passwords mean that they can share devices in school.
Year 3 ***Year 4***	**Skills** • Selecting and using appropriate devices for a specific purpose. • Recognising the shape and name of connectors and their ports. • Connecting or pairing a wireless peripheral, like headphones or a micro:bit. **Knowledge** • To know that a peripheral is something that you can connect to a device. • To know that ports are where peripherals can be attached. • To know that a computer is made from different components that do different jobs. • To know that some peripherals connect wirelessly.	**Skills** • Using search engines effectively to find information and and navigate links. • Recognising that a network is two or more wireless or wired connected devices. • Identifying examples of networks)e.g. school network, home Wi-Fi, internet)and their components (e.g. router, switch and server.). • Communicate digitally with others. **Knowledge** • To know that a network is a group of two or more connected computers. • To know that networks are made up of components such as servers, routers and switches. • To know that networks allow devices to share information and resources (e.g. printers and files). • To know that some networks are local and some are global. • To know what happens when they access a website. • To know that text, images and videos on the internet are all forms of data. • To know the world wide web consists of web pages and websites. • To know that the internet enables collaboration and communication. • To know the history and evolution of the internet	**Skills** • Saving and retrieving files from different places on a network. • Create and name a folder, then save files into it. • Using a cloud-based tool to save, retrieve and share a file (Google drive, OneDrive). **Knowledge** • To know that there are different places to save their work. • To know the difference between local, network and cloud storage. • To know that a server stores information and services.	**Skills** • Creating strong passwords. • Independently logging in and out. • Describing how computer accounts are personalised to a user. • Identifying places on the network that they can access. • Recognising that having multiple logins and passwords adds more security. **Knowledge** • To know that some areas on a network or device are restricted. • To know that permission means they can access part of a network. • To know that different logins have different purposes. • To know that the school network is a private system that can only be accessed at school.

Year 5 ***Year 6***	**Skills** • Using multiple devices or peripherals to achieve a goal (screen sharing, connecting to a printer). • Using logical steps to troubleshoot hardware by checking power, connections and settings. **Knowledge** • To know that hardware problems can be solved by checking connections and settings.	**Skills** • Evaluating search results to determine how useful they are. • Explain how search engines work and how they present information. • Beginning to describe a file's journey through a network (e.g., from a computer to a server). • Describing how a website works (e.g., how a browser requests a web page from a server). • Describing how data is broken down into packets, sent across a network and reassembled to form usable information. **Knowledge** • To know that search engine results are ranked and influenced by factors like keywords, advertisements and popularity. • To know that packets of data travel across the internet using a system of IP addresses. • To know that each packet contains pieces of data and the instructions on how to assemble them in the right order. • To know that files or requests sent across a network pass through different devices. • To know that websites are hosted on servers. • To know the history and evolution of computer systems.	**Skills** • Reducing file size for efficient storage. • Recognising how much data they are storing. • Describing the purpose of an storage device. • Choosing different file types to store information efficiently (jpeg, png have different file sizes). **Knowledge** • To know that downloading is copying data from one storage place to another, like from the cloud to a local device. • To know that using a cloud or a network to store data is more reliable than storing data locally.	**Skills** • Recognising that you can add extra layers of security to your login. • Describing how devices connected to the internet are at risk to hackers. • Recognising that some login methods are more secure than others. **Knowledge** • To know that networks need security to prevent cyber attacks. • To know that hackers gain access to networks or devices unlawfully. • To know that people can gain access to devices on networks if these are not properly secured.

Computing systems and networks				
	Understanding programming	**Plan**	**Create**	**Evaluate**
EYFS	**Computational thinking** • Using logical reasoning to understand simple instructions and predict the outcome. **Programming** • Following instructions as part of practical activities and games. • Learning to give simple instructions. • Experimenting with programming a Bee-bot/Blue- bot and learning how to give simple commands. • Learning to debug instructions, with the help of an adult, when things go wrong. **Knowledge** • To know that being able to follow and give simple instructions is important in computing. • To understand that it is important for instructions to be in the right order. • To understand why a set of instructions may have gone wrong. • To know that you can program a Bee-Bot with some simple commands.			

	• To understand that debugging means how to fix some simple programming errors. • To understand that an algorithm is a set of clear and precise instructions.			
Year 1	**Skills** • Recognising that robots are programmed by humans. **Knowledge** • To know that humans need to give robots instructions to follow and that they will carry out these instructions exactly, even if they are wrong. • To know that humans need to give instructions in the correct language for the robot to understand.	**Skills** • Explaining what they are trying to achieve with their instructions (algorithms). • Writing clear, sequenced instructions (algorithms) for familiar tasks. • Using terms like 'start,' 'end' and 'next' to describe the steps in instructions (algorithms). **Knowledge** • To know that humans need to give robots instructions to follow and that they will carry out these instructions exactly, even if they are wrong. • To know that humans need to give instructions in the correct language for the robot to understand.	**Skills** • Changing their instructions or algorithms into code that the robot understands.	**Skills** • Beginning to identify errors in instructions (algorithms). • Making suggestions for how to fix errors in instructions (algorithms). **Knowledge** • To know that there may be an error if a set of instructions (an algorithm) does not give the expected result. • To know that errors could result from sequencing, unclear instructions or missing steps.
Year 2	**Skills** • Recognising examples of computers being programmed by humans to perform simple tasks (computer games, animations, computer-generated music). **Knowledge** • Identifying errors in code within a simple program. • Correcting errors or bugs in simple programs they have created.	**Skills** • Explaining what they are trying to achieve with their program. • Verbally planning what they aim to achieve with a program. **Knowledge** • To know that an algorithm is a set of instructions used to carry out a task. • To know that an algorithm becomes a program when it is coded. • To know that programs execute the exact instructions they are given, even if they are incorrect. • To know that a program is a series of instructions (algorithms) that are written for a computer to follow. • To know that a person can program a device by giving it an algorithm/algorithms to follow.	**Skills** • Using basic block code to implement an algorithm. • Looking at different forms of code (Bee-bot and Scratch) and predicting how a program will execute. • Tinkering with a basic existing visual code to see how it affects a program.	**Skills** • Identifying errors in code within a simple program. • Correcting errors or bugs in simple programs they have created. **Knowledge** • To know that there must be an error if a program does not execute as expected. • To know that an error in a computer program is known as a 'bug' and fixing errors is known as 'debugging'.
Year 3 ***Year 4***	**Skills** • Recognising visual and text-based programming languages. • Recognising further examples of computers being programmed by humans to perform simple tasks. **Knowledge** • To know that websites have all been programmed, often using HTML. • To know that programming languages can be visual (Scratch) or text-based (HTML). • To know that 'coding' is the process of turning an algorithm into a programming language. • To know that HTML tells a browser how to display text, images and multimedia on a webpage.	**Skills** • Working towards a given goal that a program needs to accomplish. • Breaking down what they want to achieve into smaller, manageable parts. • Using logic, pattern recognition and decomposition to solve simple problems. **Knowledge** • To know that 'decomposition' is the process of breaking down a task or problem into smaller parts. • To know that breaking down a problem into smaller parts makes it easier to solve the problem. • To know that 'abstraction' is identifying the important detail and ignoring irrelevant information.	**Skills** • Tinkering with an existing text-based code to see how it affects a program (website). • Remixing code to alter and add to an existing program. • Recognising repeating patterns in a program or code. • Creating loops to make code more efficient in block-based programs. • Beginning to use variables in block-based programming languages to make programs more interactive. • Including a conditional statement in block-based programming languages. **Knowledge** • To know that loops are used to save time when writing code by reducing	**Skills** • Recognising the relationship between what is happening in a program and the written (block) code. • Working backwards, beginning to identify the code they think a program uses. • Running small chunks of code at a time to find the error or 'bug.' **Knowledge** • To know that it is important to identify where the mistake is in the programming as part of the debugging process. • To know that errors in a program could result from sequencing errors, coding errors or missing code.

			repetition. • To know that a variable is a container or holder for storing information that can change, e.g. numbers or text. • To know that conditional statements tell the computer what to do next based on a user's input.	
Year 5 ***Year 6***	**Skills** • Recognising a wider range of text-based programming languages. • Making links between different programming interfaces they are faced with. • Recognising examples of programming elements in real-life applications (e.g. live loops in music, variables in pedometers). **Knowledge** • To know that code may sometimes need to be downloaded onto a physical system/device. • To know that devices with sensors (such as pedometers, security systems, thermostats and light sensors) are often programmed to perform specific tasks in reaction to the input from the sensors.	**Skills** • Looking at programming blocks and considering how they could be used in a program. • Planning a program to be downloaded to a physical system/device to solve a particular problem. • Decomposing a program independently when given a specific outcome or task to achieve. **Knowledge** • To know that programmers often save time when creating code by taking code from one program and turning it into another.	**Skills** • Live coding (improvising with code). • Programming a physical system that has the capability for sensory input (e.g. a micro:bit). • Altering existing code with a new, specific outcome in mind. • Independently using loops to make code more efficient in text-based or block-based programs (e.g. Sonic Pi and Logo). • Using nested loops to make code more efficient. • Recognising real-life applications of conditional statements (e.g. when the temperature reaches 20 ℃, turn the heating off). • Using variables in block-based programming languages and understanding the impact of changing the variables in their code. **Knowledge** • To know that nested loops are loops within loops. • To know that it is important to follow the syntax rules in a programming language so the computer understands what we are trying to tell it but that we do not need to remember all these rules. • To know that computers use pixels to measure length in digital visuals. • To know that positions are often understood by a computer as x and y coordinates. • To know that turns or rotations are often described as degrees. • To know that many text-based coding languages use brackets or indentation to show which code belongs to a particular function or loop.	**Skills** • Explaining what a program does and how it works, referring to the inputs and outputs. • Becoming more efficient and effective at debugging their programs. • Systematically identifying mistakes, problems or 'bugs' in a program. **Knowledge** • To know that running a program to identify errors should be done before checking the code. • To know that errors in a program could be as a result of forgetting to 'end' a loop. • To know that typing and spacing are very important in text-based languages and can cause errors in code if used incorrectly.

Computing systems and networks				
	Understanding	**Plan**	**Create**	**Evaluate**

	programming			
EYFS	**Computational thinking** • Using logical reasoning to understand simple instructions and predict the outcome. **Programming** • Following instructions as part of practical activities and games. • Learning to give simple instructions. • Experimenting with programming a Bee-bot/Blue- bot and learning how to give simple commands. • Learning to debug instructions, with the help of an adult, when things go wrong. **Knowledge** • To know that being able to follow and give simple instructions is important in computing. • To understand that it is important for instructions to be in the right order. • To understand why a set of instructions may have gone wrong. • To know that you can program a Bee-Bot with some simple commands. • To understand that debugging means how to fix some simple programming errors. • To understand that an algorithm is a set of clear and precise instructions.			
Year 1	**Skills** • Recognising that robots are programmed by humans. **Knowledge** • To know that humans need to give robots instructions to follow and that they will carry out these instructions exactly, even if they are wrong. • To know that humans need to give instructions in the correct language for the robot to understand.	**Skills** • Explaining what they are trying to achieve with their instructions (algorithms). • Writing clear, sequenced instructions (algorithms) for familiar tasks. • Using terms like 'start,' 'end' and 'next' to describe the steps in instructions (algorithms). **Knowledge** • To know that humans need to give robots instructions to follow and that they will carry out these instructions exactly, even if they are wrong. • To know that humans need to give instructions in the correct language for the robot to understand.	**Skills** • Changing their instructions or algorithms into code that the robot understands.	**Skills** • Beginning to identify errors in instructions (algorithms). • Making suggestions for how to fix errors in instructions (algorithms). **Knowledge** • To know that there may be an error if a set of instructions (an algorithm) does not give the expected result. • To know that errors could result from sequencing, unclear instructions or missing steps.
Year 2	**Skills** • Recognising examples of computers being programmed by humans to perform simple tasks (computer games, animations, computer-generated music). **Knowledge** • Identifying errors in code within a simple program. • Correcting errors or bugs in simple programs they have created.	**Skills** • Explaining what they are trying to achieve with their program. • Verbally planning what they aim to achieve with a program. **Knowledge** • To know that an algorithm is a set of instructions used to carry out a task. • To know that an algorithm becomes a program when it is coded. • To know that programs execute the exact instructions they are given, even if they are incorrect. • To know that a program is a series of instructions (algorithms) that are written for a computer to follow. • To know that a person can program a device by giving it an algorithm/algorithms to follow.	**Skills** • Using basic block code to implement an algorithm. • Looking at different forms of code (Bee-bot and Scratch) and predicting how a program will execute. • Tinkering with a basic existing visual code to see how it affects a program.	**Skills** • Identifying errors in code within a simple program. • Correcting errors or bugs in simple programs they have created. **Knowledge** • To know that there must be an error if a program does not execute as expected. • To know that an error in a computer program is known as a 'bug' and fixing errors is known as 'debugging'.
Year 3	**Skills** • Recognising visual and text-based programming languages. • Recognising further examples of	**Skills** • Working towards a given goal that a program needs to accomplish. • Breaking down what they want to	**Skills** • Tinkering with an existing text-based code to see how it affects a program (website).	**Skills** • Recognising the relationship between what is happening in a program and the written (block) code.

Year 4	computers being programmed by humans to perform simple tasks. **Knowledge** • To know that websites have all been programmed, often using HTML. • To know that programming languages can be visual (Scratch) or text-based (HTML). • To know that 'coding' is the process of turning an algorithm into a programming language. • To know that HTML tells a browser how to display text, images and multimedia on a webpage.	achieve into smaller, manageable parts. • Using logic, pattern recognition and decomposition to solve simple problems. **Knowledge** • To know that 'decomposition' is the process of breaking down a task or problem into smaller parts. • To know that breaking down a problem into smaller parts makes it easier to solve the problem. • To know that 'abstraction' is identifying the important detail and ignoring irrelevant information.	• Remixing code to alter and add to an existing program. • Recognising repeating patterns in a program or code. • Creating loops to make code more efficient in block-based programs. • Beginning to use variables in block-based programming languages to make programs more interactive. • Including a conditional statement in block-based programming languages. **Knowledge** • To know that loops are used to save time when writing code by reducing repetition. • To know that a variable is a container or holder for storing information that can change, e.g. numbers or text. • To know that conditional statements tell the computer what to do next based on a user's input.	• Working backwards, beginning to identify the code they think a program uses. • Running small chunks of code at a time to find the error or 'bug.' **Knowledge** • To know that it is important to identify where the mistake is in the programming as part of the debugging process. • To know that errors in a program could result from sequencing errors, coding errors or missing code.
Year 5 ***Year 6***	**Skills** • Recognising a wider range of text-based programming languages. • Making links between different programming interfaces they are faced with. • Recognising examples of programming elements in real-life applications (e.g. live loops in music, variables in pedometers). **Knowledge** • To know that code may sometimes need to be downloaded onto a physical system/device. • To know that devices with sensors (such as pedometers, security systems, thermostats and light sensors) are often programmed to perform specific tasks in reaction to the input from the sensors.	**Skills** • Looking at programming blocks and considering how they could be used in a program. • Planning a program to be downloaded to a physical system/device to solve a particular problem. • Decomposing a program independently when given a specific outcome or task to achieve. **Knowledge** • To know that programmers often save time when creating code by taking code from one program and turning it into another.	**Skills** • Live coding (improvising with code). • Programming a physical system that has the capability for sensory input (e.g. a micro:bit). • Altering existing code with a new, specific outcome in mind. • Independently using loops to make code more efficient in text-based or block-based programs (e.g. Sonic Pi and Logo). • Using nested loops to make code more efficient. • Recognising real-life applications of conditional statements (e.g. when the temperature reaches 20 ℃, turn the heating off). • Using variables in block-based programming languages and understanding the impact of changing the variables in their code. **Knowledge** • To know that nested loops are loops within loops. • To know that it is important to follow the syntax rules in a programming language so the computer understands what we are trying to tell it but that we do not need to remember all these rules. • To know that computers use pixels to measure length in digital visuals. • To know that positions are often understood by a computer as x and y coordinates.	**Skills** • Explaining what a program does and how it works, referring to the inputs and outputs. • Becoming more efficient and effective at debugging their programs. • Systematically identifying mistakes, problems or 'bugs' in a program. **Knowledge** • To know that running a program to identify errors should be done before checking the code. • To know that errors in a program could be as a result of forgetting to 'end' a loop. • To know that typing and spacing are very important in text-based languages and can cause errors in code if used incorrectly.

			• To know that turns or rotations are often described as degrees. • To know that many text-based coding languages use brackets or indentation to show which code belongs to a particular function or loop.	

Creating media

	Plan	Creating digital content	Design principles	Evaluate
EYFS	**Using software** • Using a simple online paint tool to create digital art.			
Year 1	**Skills** • Making decisions about the digital content they make. • Ordering ideas so that content makes sense. **Knowledge** • To know that ordering ideas is important to make content clear. • To know that digital tools can be used to make content that is presented clearly.	**Skills** • Using a basic drawing or writing tools to create a simple piece of digital content (e.g., drawing a picture, typing words). • Selecting and using simple editing tools (e.g., formatting, resizing). • Typing simple words and short phrases. **Knowledge** • To know that ordering ideas is important to make content clear. • To know that digital tools can be used to make content that is presented clearly.	**Skills** • Choosing and arranging simple elements (e.g., text, images, colours) to make digital content clear and appealing. • Using basic tools to change the appearance of digital content (e.g., selecting different brush styles, filling shapes with colour, formatting text). **Knowledge** • To know that colours, shapes and space can be used to make content clearer and more interesting.	**Skills** • Explaining their decisions about the digital content they created and the tools they used to make it. • Making simple changes to improve their work (e.g., changing colours, deleting). **Knowledge** • To know that digital content can be changed and improved.
Year 2	**Skills** • Identifying the audience and purpose of digital content. • Planning how to use the space on a digital page. **Knowledge** • Knowledge • To know that there must be an error if a program does not execute as expected. • To know that an error in a computer program is known as a 'bug' and fixing errors is known as 'debugging'.	**Skills** • Importing images onto a digital page. • Positioning images and text on a digital page.. • Typing familiar words and short sentences. • Cutting, copying and pasting. **Knowledge** • To know that text and images can be combined to make digital content. • To know that cut and paste can be used to change the position of text or images.	**Skills** • Arranging text and images to create an organised, clear layout. **Knowledge** • To know that size, position and space can be used to organise digital content. • To know that different fonts and colours can make text easier to read. • To know that images and text work together to communicate.	**Skills** • Suggesting a change to their digital content or someone else's to improve how clear it is. • Making small improvements based on feedback (e.g., reordering text, adjusting image placement). **Knowledge** • To know that size, position and space can be used to organise digital content. • To know that different fonts and colours can make text easier to read. • To know that images and text work together to communicate.
Year 3	**Skills** • To know that size, position and space can be used to organise digital content. • To know that different fonts and colours can make text easier to read. • To know that images and text work together to communicate. **Knowledge** • To know that websites are created using carefully selected and created	**Skills** • Capturing digital images (e.g. photos and video). • Adding layers to create digital content both photos and videos. • Using the features of a word processor (e.g. tabs and hyperlinks) to structure digital content. • Editing and refining images (e.g., cropping, adjusting brightness/contrast.	**Skills** • Choosing colours and adjusting spacing and positioning to create a well-organised layout. • Adjusting the order of clips on a timeline in video content. **Knowledge** • To know that a timeline with layers is a way of organising video content.	**Skills** • Reviewing digital content made by others to check that it achieves a given goal. • Editing and refining digital content to improve clarity and effectiveness. **Knowledge** • To know that a timeline with layers is a way of organising video content.

Year 4	media. • To know that digital images (e.g. photos and videos) and websites are created with different goals in mind.)	• Editing and refining video content (e.g., trimming, adjusting brightness/contrast). • Using content blocks to insert text and images into a webpage. **Knowledge** • To know that websites are created using carefully selected and created media. • To know that digital images (e.g. photos and videos) and websites are created with different goals in mind.)		
Year 5 ***Year 6***	**Skills** • Selecting appropriate types of digital content (e.g., text, audio, image, video) to meet the goal of a multimedia project.) • Setting a goal for a multimedia project and identifying success criteria. • Using a spreadsheet to plan and organise digital content. • Plan a sequence of images to create the effect of movement. **Knowledge** • To know that digital content should achieve a given goal. • To know that recording a project plan in an efficient way helps with organisation.	**Skills** • Combining and manipulating different types of digital content to create a multimedia project. • Choosing and combining sound or music with other digital content. • Making edits or changes to improve the quality of an audio recording. **Knowledge** • To know that audio clips can be cut into smaller parts. • To know that combining many still images can create the effect of movement. • To know that AI tools can generate and edit digital content, such as text, images, or audio, and that AI-generated content requires human refinement.	**Skills** • Creating simple and focused digital content. • Use templates to ensure content is consistent. • Selecting music to match the mood of digital content. • Adjusting the timing and speed of clips on a timeline. **Knowledge** • To know that keeping digital content simple and consistent makes it easier for people to use and understand. • To know that different audiences may require different design choices (e.g., using high contrast for readability).	**Skills** • Evaluating a digital project against specific criteria. • Applying feedback from a reviewer to improve a multimedia project so that it meets the goal. **Knowledge** • To know that having specific criteria helps to evaluate and improve digital content.

Data handling

	Understanding data	Plan	Create	Evaluate
EYFS	**Using data** • Representing data through sorting and categorising objects in unplugged scenarios. • Representing data through physical pictograms. • Exploring branch databases through physical games. **Knowledge** • To know that sorting objects into various categories can help you locate information. • To know that using yes/no questions to find an answer is a branching database. • To know that a pictogram is a way of showing information.			
Year 1 ***Year 2***	**Skills** • Recognising that data is collected for different purposes. **Knowledge** • To know that data can be organised into groups. • To know different ways data can be shown on a screen.	**Skills** • Collect simple data using a survey. • Group data using simple digital tools (table, tally chart). • Organise data using labels or categories. **Knowledge** • To know that data can be organised into	**Skills** • Representing data using different features of digital tools (pictogram, bar chart). **Knowledge** • To know that colours, pictures, labels can be changed using digital tools. • To know that digital tools organise data	**Skills** • Checking that data in a table or tally matches their collected data. • Checking that data in a graph or pictogram matches the data in the table or tally. **Knowledge** • To know that organising data makes it

	• To know that digital data can be changed and updated.	groups. • To know different ways data can be shown on a screen. • To know that digital data can be changed and updated.	for you, once it is entered. • To know that digital tools allow changes to be made as they create.	easier to understand. • To know that the same data can be presented in different ways.
Year 3 ***Year 4***	**Skills** • Exploring information in a branching database. • Following Yes/No questions to see how a branching database works. **Knowledge** • To know that a branching database sorts information using yes/no questions. • To know that questions in a branching database help narrow down results. • To know that data can be grouped using shared characteristics. • To know that a database is a collection of organised information that people use.	**Skills** • Planning categories for a digital branching database. • Selecting appropriate data to include. • Identifying suitable questions to classify objects. **Knowledge** • To know that a branching database sorts information using yes/no questions. • To know that questions in a branching database help narrow down results. • To know that data can be grouped using shared characteristics. • To know that a database is a collection of organised information that people use.	**Skills** • Creating a simple branching database using a digital tool. • Entering and organising data using categories and questions. • Testing a branching database. **Knowledge** • To know that digital tools organise and present result automatically. • To know that edits can be made to the database as they create them.	**Skills** • Comparing data on digital tools with data produced by hand. • Using a branching database to find information. • Checking whether questions work effectively. **Knowledge** • To know that digital tools organise and present result automatically. • To know that edits can be made to the database as they create them.
Year 5 ***Year 6***	**Skills** • Exploring data in spreadsheets. • Identifying some real world applications of spreadsheets. **Knowledge** • To know that data in spreadsheets can be used to create charts and graphs. • To know that spreadsheets organise data into rows, columns and cells. • To know that spreadsheets can analyse large amounts of data quickly.	**Skills** • Enter, edit and organise data in a spreadsheet. **Knowledge** • To know that columns and rows should have titles and headings. • To know that each cell has a unique cell reference.	**Skills** • Formatting data in a spreadsheet. • Using simple formulas that include the four operations from maths to automate calculations. • Creating charts from spreadsheet data (bar, line, pie). **Knowledge** • To know that columns and rows should have titles and headings. • To know that each cell has a unique cell reference.	**Skills** • Evaluate the effectiveness of a spreadsheet. • Answering questions about the data in a spreadsheets. **Knowledge** • To know that errors in data or formulas can affect results.

Online safety

	Being online	Online relationships	Risks and harms	Responsible and safe use	Reporting online
EYFS	**N/A**				
Year 1	**Skills** • Recognising when they are online. • Identifying programs (e.g. apps and websites) they use	**Skills** • Recognising that being kind, polite and safe is just as important online as in-person.	**Skills** • Describing how someone might change their account if they have their password. **Knowledge**	**Skills** • Checking with an adult before they use new apps or websites. • Beginning to remember	**Skills** • Recognising when something online scares or worries them. • Describing how to respond safely to an online message.

Year 2	online. • Recognising indicators of being logged into an account (e.g. avatar, username, initials). **Knowledge** • Recognising when something online scares or worries them. • Describing how to respond safely to an online message.	**Knowledge** • To know they should not speak to or message unfamiliar people online. • To know that they can still hurt someone's feelings online.	• To know they should not speak to or message unfamiliar people online. • To know that they can still hurt someone's feelings online.	their username and password for an online account. • Treating devices carefully. **Knowledge** • To know that using someone else's login is not safe or respectful.	**Knowledge** • To know that using someone else's login is not safe or respectful.
Year 3 ***Year 4***	**Skills** • Identifying what the internet can be used for. **Knowledge** • To know that online games allow players to interact with each other in various ways.	**Skills** • Identifying how people connect online. • Recognising the difference between acceptable and unacceptable behaviour. **Knowledge** • To know that connecting online can involve actions such as liking, commenting, emoting and chatting. • To know that if behaviour is unacceptable in-person, it is unacceptable online.	**Skills** • Identifying when online contact from other players is unwanted or inappropriate. • Recognising signs of online bullying (e.g. repeated unkind comments or being excluded). **Knowledge** • To know that usernames should not contain personal information. • To know that not everyone online is who they claim to be. • To know that online bullying is repeated unkind behaviour.	**Skills** • Describing appropriate and safe responses to unwanted or unkind behaviour. **Knowledge** • To know that being anonymous online does not make unkind behaviour acceptable.	**Skills** • Recognising when something online is unsafe, inappropriate or suspicious. **Knowledge** • To know reporting problems online to a trusted adult is an important part of staying safe.
Year 5 ***Year 6***	**Skills** • Identifying the features of social media platforms. **Knowledge** • To know that online platforms are digital spaces, such as apps or websites, where people interact. • To know that online platforms show digital content based on the things they have liked, watched or subscribed to.	**Skills** • Recognising how their online identity is shaped by what they share, like and comment on. **Knowledge** • To know that interactions through online platforms such as social media should still be polite and respectful. • To know that people may present themselves differently online.	**Skills** • Identifying when online apps or games are trying to encourage people to spend money. **Knowledge** • To know that the things they share and post online can remain online. • To know that once something is shared online, it cannot always be deleted or fully removed. • To know that sharing images or posting comments online is often public.	**Skills** • Recognising when games are trying to persuade them to spend money. **Knowledge** • To know that social media platforms have age restrictions. • To know that if they are under the age limit for a particular online platform or app, they should not create an account.	**Skills** • Using tools like block, mute, flag and report on different online platforms. • Identifying which reporting tools to use in a specific situation. **Knowledge** • To know that the block and mute tools can stop people from contacting them online. • To know that reporting or flagging content online will anonymously alert someone to a problem.