Annesley Primary & Nursery School Progression Map

~

Computing

At Annesley Primary and Nursery School, we believe that 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 our curriculum is designed to ensure that every child becomes a confident, creative and responsible user of technology who understands how digital systems work and can put this knowledge to use through programming. We aim to provide an ambitious, coherently sequenced curriculum in which pupils develop the knowledge, skills and confidence to participate safely and successfully in an increasingly digital world.

Throughout their time in school, pupils move from following instructions and using programmable toys in the early years, through block-based coding in 2Code, 2Go and Logo, to writing programs with variables, functions and input, and using text-based Python by Year 6. Alongside this, they learn to use a widening range of software to create, combine and present content, to collect and analyse data in spreadsheets and databases, and to search for and judge information critically. Precise computing vocabulary is taught explicitly in every unit, and online safety is woven through all learning so that pupils know how to protect themselves and others and where to go for help.

By the end of Year 6, pupils leave Annesley Primary and Nursery School as confident, creative and responsible digital citizens who can design, write and debug programs to achieve specific goals, use logical reasoning to explain how algorithms work, select and combine software to create content and analyse data, evaluate digital content critically, and use technology safely and respectfully. They are well prepared to continue their study of computing at Key Stage 3 and beyond.

Our ARCHIE framework embeds six core values across our curriculum – Ambition, Resilience, Confidence, Honesty, Independence, and Equality. Below are how these values apply to Computing at Annesley Primary & Nursery School:

A	We prepare our children to be digital leaders in a technology-based society by promoting aspirational understanding and fluency across the three realms of digital literacy, computer science and information technology. Pupils follow a well-sequenced curriculum that moves from simple techniques towards more complex concepts, revisiting key knowledge and skills so that understanding deepens over time. Children are encouraged to aim high, to look beyond being users of technology and to see themselves as creators, problem-solvers and future leaders in a digital world. By exploring how technology is used in the real world, in careers and in everyday life, pupils develop the aspiration to use their skills purposefully and to a high standard. This ambitious approach ensures that every child is equipped with the knowledge and fluency to thrive in a rapidly changing technological landscape.
R	Through programming, coding and debugging, children learn that challenges often present themselves and that setbacks are a natural and valuable part of working with technology. Pupils are encouraged to constantly re-visit, retry and re-evaluate their work to find a solution, understanding that a program that does not work first time is an opportunity to learn rather than a reason to give up. They develop the perseverance to test their ideas, identify errors, and make thoughtful improvements, recognising that progress comes through sustained effort and systematic thinking. By valuing the process of debugging as much as the finished outcome, children build the determination and problem-solving mindset needed to tackle increasingly complex computing challenges, both in school and beyond.
C	We equip our children with the skills to be confident in navigating an increasingly online world and ensure that they are confident in safeguarding themselves whilst doing so, building practical competence and trust in their own abilities. Children learn how to protect their personal information, how to behave safely online and what to do when something makes them feel uncomfortable, including knowing who to turn to for help. This growing self-assurance enables pupils to approach new technologies without fear, to experiment with new tools, and to see themselves as capable and responsible digital citizens. Confidence in both skill and safety allows children to benefit from the opportunities technology offers while remaining protected from its risks.
H	We promote honesty and openness online, helping children understand the importance of being truthful and responsible in their digital interactions. Pupils are taught to be mindful of, and able to spot, content that could be misleading or untrue, developing the critical thinking skills to question the sources they encounter and to check the reliability of information. Children learn to be open about their online experiences and to speak honestly with trusted adults when they are unsure or concerned. They also reflect honestly on their own work, acknowledging what has been successful and where improvements are needed. By fostering a culture of honesty and critical evaluation, we prepare pupils to engage thoughtfully and responsibly with information throughout their lives.
I	We foster a love of technology that enables children to independently innovate and create their own digital content. Pupils are given opportunities to make their own choices about the tools, software and approaches they use, taking ownership of their learning and creative decisions. Whether designing a program, producing a presentation, creating digital artwork or organising data, children develop the autonomy to plan, build and refine their own projects. When challenges arise, pupils are encouraged to think for themselves, explore possible solutions and adapt their approach rather than relying solely on adult guidance. This growing independence empowers children to become self-directed learners and creative innovators who can continue to explore and develop their digital skills beyond the classroom.
E	We ensure that children understand the importance of showing respect and tolerance for others when creating digital content or communicating with others online. Pupils learn that every person deserves to be treated with kindness and dignity in digital spaces, just as they are in the real world, and that differences in opinion, background and belief should be respected. Through adaptive teaching, all children are supported to access equipment, develop their skills and participate fully in computing, regardless of their starting point or additional needs. We promote equal access to high-quality computing experiences for every pupil, removing barriers to participation so that all children can see themselves as capable users and creators of technology. By modelling and teaching respectful online behaviour, we help pupils become considerate digital citizens who contribute positively to online communities.

		EYFS	Key Stage 1		Key Stage 2			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Algorithms and Computational Thinking	**Knowledge**	• Instructions tell us what to do, e.g. in songs and games. • Instructions must be followed in the right order, e.g. first, next, last. • A programmable toy follows instructions that someone has given it. **Key vocabulary:** instruction, order, first, next, last, forwards, backwards, turn, toy	• An algorithm is a set of ordered instructions used to solve a problem or achieve an objective. • Instructions must be clear, precise and in the right order, because a computer follows them exactly. • When pupils code, they turn an algorithm into something a computer can follow (Following Instructions, Coding). • In 2Code, an object block and an action block make a command; a set of blocks arranged together is an algorithm. **Key vocabulary:** algorithm, instruction, order, sequence, precise, clear, command, code block, object, action	• An algorithm is a set of instructions whose steps must be followed in order to get the intended outcome. • An algorithm can be written as a plan, then converted into code (Route Explorers, Coding). • In 2Go, a command moves a screen turtle using direction keys (eight directions) and number keys for distance on a grid. **Key vocabulary:** algorithm, command, direction, distance, plan, grid, turtle, route, sequence, event	• A flowchart is a diagram that shows an algorithm using labelled boxes and arrows, with a start, inputs and outputs, delays and processes (2Chart). • Tasks can be decomposed into smaller steps, e.g. planning the route of a turtle (Route Planners). • Angles of turn and repeated steps can be included in algorithms, which makes them more efficient. **Key vocabulary:** flowchart, decompose, step, input, output, process, angle, rotation, repeat, design	• A task can be decomposed into smaller parts so that it can be coded more easily, e.g. drawing a shape or letter in Logo. • Algorithms can include selection, shown in a flowchart as a decision: what happens if a condition is or is not met. • More efficient code does the same job with fewer commands, e.g. by using repeat. **Key vocabulary:** decompose, selection, condition, flowchart, Logo, repeat, efficient, pattern	• Decomposition breaks a task into manageable parts that can be coded separately and brought back together. • Abstraction removes unnecessary detail so a program can be made to work, e.g. leaving out pedestrians when modelling traffic lights. • A simulation is a model of a real or imaginary situation and can be planned as an algorithm. **Key vocabulary:** decomposition, abstraction, simulation, model, algorithm, efficient, function	• Detailed program design documentation (plans and flowcharts) produces accurate algorithms before coding starts. • Decomposition and abstraction are used to simplify and break down more complex tasks. • Algorithms can be written as flowcharts showing procedures, and then coded in block or text-based languages such as Python. **Key vocabulary:** design documentation, abstraction, decomposition, procedure, flowchart, algorithm, Python, text-based code
	Skills	• Follow and give simple instructions in the right order, e.g. in a game or a recipe. • Use programmable toys to follow a simple set of instructions.	• Give clear instructions that others can follow, and follow instructions given by others. • Put a set of instructions, e.g. for making a	• Create a computer program from an algorithm they have made, turning each step into code. • Follow a written algorithm on a	• Plan algorithms with increasing detail, decomposing the task, e.g. planning a route. • Write commands that include angles	• Independently design, code, test and debug a simple program. • Decompose a task, e.g. drawing a shape or a letter, into steps before coding.	• Consider a range of program structures when turning a task into an algorithm. • Plan an algorithm for a physical system, e.g. a traffic light	• Produce accurate algorithms from program design documentation, working at a deeper level of abstraction. • Follow flowcharts to design and debug programs.

		• Talk about what will happen when instructions are followed. • Explain what they asked the toy to do.	sandwich, into the correct order. • Draw symbols to represent instructions, and arrange printed object and action blocks into commands. • Plan a simple scene and what the objects in it will do.	plan and interpret it. • Build a simple algorithm in 2Go to move a turtle along a route, extending it when more commands are needed. • Identify events, objects and actions in a plan.	of turn and repetition in 2Go. • Follow and create simple flowcharts to represent an algorithm in 2Chart. • Use design documentation to plan a program before coding.	• Interpret flowcharts that show selection and explain what happens if a condition is or is not met. • Consider the efficiency of an algorithm, e.g. by using repeat.	sequence, including timings. • Use decomposition and abstraction to simplify a plan. • Convert the algorithm into code and compare how it behaves with real life.	• Use decomposition and abstraction confidently to simplify and break down tasks. • Translate their understanding of block code algorithms to text-based Python.
		EYFS	**Key Stage 1**		**Key Stage 2**			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Programming: Sequence, Selection and Repetition	**Knowledge**	• Toys and devices respond when we press buttons. • A programmable toy can be given a short sequence of commands.	• In 2Code, there are object blocks and action blocks; each single instruction such as 'Object Right' is a command. • An event is something that makes a block of code run, e.g. When Clicked. • When code is run, it is being executed (Run and Stop buttons); scenes can be made with backgrounds and objects, which have attributes that can be changed.	• A program follows a sequence of commands in order; events include When Key, When Swiped, When Clicked and collision detection. • Timers make parts of a program run after a set time (timer after); buttons are an object type that run code when clicked. • Objects of different types have attributes (properties), e.g. image and size, that can be changed.	• Timers can run after a delay (timer after) or at regular intervals (timer every). • The repeat command runs a block of commands a set number of times, avoiding repetition in code. • Code can be nested inside other code, which affects when it runs, e.g. a timer inside a When Clicked event (Coding, Route Planners).	• Logo (2Logo) is a programming language with its own instructions, including non-directional ones such as PU, PD and CS; repeat creates shapes efficiently. • Selection is a decision command: an if statement runs code only if a condition is true; if/else runs one set of commands or the other. • Repeat until runs code until a condition is met; coordinates (x and y) give an object's position. • A variable is a container in memory for a value that can change, e.g. a score; a number variable can be	• Computer-generated variables (tags) allow several objects to be controlled with fewer lines of code, which is more efficient. • A function is a block of code that is created once and called whenever it is needed. • Strings are text; string and number variables hold information; concatenation joins strings together. • A simulation, e.g. a traffic light sequence, can be programmed using timers, events and variables; timer every can repeat code forever.	• Input is information going into a computer, e.g. a key press, swipe, click or typing; prompt for input and get input are user inputs used in 2Code. • Number variables, if/else and a timer can build an on-screen countdown; coordinates, the launch command and tabs help organise more complex code. • Functions (procedures) are independent pieces of code, e.g. drawing regular shapes, and can be called many times. • Block code can be translated into text-based code; Python uses text commands, data types and repetition.

		Key vocabulary: button, press, move, toy, forward, turn, go, stop	**Key vocabulary:** 2Code, command, event, When Clicked, execute, run, object, action, attribute, background	**Key vocabulary:** sequence, timer, collision detection, button, attribute, property, event, turtle, 2Go	**Key vocabulary:** timer after, timer every, repeat, nesting, loop, sequence, rotation, command	watched as the program runs. **Key vocabulary:** Logo, selection, if, else, condition, repeat until, coordinates, variable, loop, score	**Key vocabulary:** efficient, tag, function, call, string, concatenation, simulation, timer every, print to screen	**Key vocabulary:** input, output, user input, prompt, get input, procedure, function, cloning, hotspot, Python, data type, repetition
	Skills	• Press buttons and use controls on toys and devices to make things happen. • Use simple programmable toys and give them a short sequence of commands. • Talk about what happens when they change a command.	• Recognise object and action code blocks in 2Code and make a command using an object and an action together. • Arrange a When Clicked block in front of an object and give the object an action. • Execute code with the Run button and stop it with the Stop button. • Create a program from a plan, with a background, objects, actions and a When Clicked event.	• Use 2Go to build routes and algorithms with direction and number commands. • Create 2Code programs with more than one object type and change attributes such as image and size. • Include When Key, When Clicked and collision detection events, and add a button and a timer after command. • Run code in sequence and explain what each command does.	• Write commands that include angles of turn and use repeat in 2Go to create routes and shapes. • Use timers (after and every) and nest code to control when it runs. • Use the repeat command as an efficient alternative to a long sequence. • Build up to coding a program from design documentation, e.g. a flowchart.	• Write and sequence Logo commands to create shapes and letters, and refine the code using repeat and procedures. • Use selection (if and if/else) in a program, and repeat until, and change object coordinates. • Create a number variable, set its value and use it to keep track of a score. • Use these structures to create their own game.	• Refine code to make it more efficient, e.g. by using tags to control several objects. • Create and use functions and call them as part of an event. • Create and use string and number variables, and use concatenation and print to screen. • Program a simulation, e.g. traffic lights, adapting it to match real-life situations.	• Write programs that use variables, user input (prompt for input and get input), if/else and repeat. • Use cloning, hotspots, coordinates, tabs and the launch command to add complexity and interactivity. • Create and call several functions (procedures), e.g. to draw regular shapes. • Compare block and text code views in Python, write code for text output, use data types and code repetition in Python.
		EYFS	**Key Stage 1**		**Key Stage 2**			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Debugging and Logical Reasoning	**Knowledge**	• Sometimes things do not work and we can try again. • We can say what we think will happen next and then check.	• Correcting errors in an algorithm or program is called debugging. • The order of instructions affects what happens, so an incorrect order gives the wrong result.	• A bug is a piece of code that stops a program working as intended; unexpected errors are caused by the code that has been created. • Debugging means looking for problems, fixing	• Testing, debugging and fixing are an important part of making programs. • Understanding nesting, e.g. a timer inside a When Clicked event, helps to predict and debug how code will run.	• Multi-step programs can be read and their outcomes predicted by looking at how different parts of the code interact. • In Logo, errors can be found by looking at the output and	• Familiarity with coding structures allows programmers to read code at a high level and focus on the part that needs debugging. • Bugs are found using a mixture of logical reasoning and trial and testing.	• A logical approach starts debugging at the design and algorithm stage, and so relies less on trial and error. • Stepping through code line by line helps find where a bug is.

			• Logical reasoning is used to predict what an object will do when a program runs, by looking at its code blocks and events. **Key vocabulary:** debug, bug, error, fix, predict, order, sequence, test	them and repeatedly testing. • A plan or algorithm can be compared with how a program runs to see if it matches. **Key vocabulary:** bug, debug, test, predict, plan, error, logical, execute	• Flowcharts help explain how an algorithm works and support logical debugging. **Key vocabulary:** debug, test, nesting, flowchart, logical reasoning, predict, explain, error	then at the code that caused them. • Debugging is more logical when the cause of a problem is identified and tested. **Key vocabulary:** predict, debug, interact, logical, output, test, error, isolate	• Simulations are tested and compared with the real-life system they model. **Key vocabulary:** debug, logical reasoning, test, trial, simulation, compare, function, efficient	• A program can be tested against a flowchart to check it works as designed. • Code can be explained in terms of what each part does and what will happen next. **Key vocabulary:** logical, step through, trace, flowchart, test, debug, explain, predict, algorithm
	Skills	• Notice when something does not work and try again. • Say what they think will happen when they press a button or give an instruction. • Say what they changed to make it work.	• Find simple errors in a set of instructions, e.g. in the sandwich algorithm, and correct them by reordering. • Analyse where code in 2Code is not working and change blocks or actions to debug it. • Predict what will happen before running a program, and test the prediction. • Recognise when an algorithm or program has been debugged.	• Recognise what a program is supposed to do and run it to check. • Locate the code responsible for an error, change it and run the program again, repeating until it works. • Compare a design plan or written algorithm with how a program runs in play mode. • Identify the parts of a program that respond to specific events.	• Read and create flowcharts for planned or existing programs, and use them to debug. • Use logical reasoning to explain how a simple program works. • Compare programs with different nesting and predict how each will run. • Test, find and fix errors, explaining what went wrong.	• Read multi-step programs and accurately predict the outcomes. • Debug logically by recognising how different parts of the code interact. • Isolate errors in Logo output, identify why they occurred and correct the code. • Test selection, loops and variables to check that they work as expected.	• Read code at a high level and focus on the parts that need debugging. • Use logical reasoning together with trial and testing to find and fix bugs. • Test a simulation against a real-life situation and adjust the code. • Check that a simplified version of the code works the same as the original.	• Default to a logical approach when debugging, beginning at the program design or algorithm stage. • Use a step-by-step approach to step through code to find errors. • Test a program against a flowchart to check that it is working as designed. • Explain the function of each part of a program, including in Python code.
		EYFS	**Key Stage 1**		**Key Stage 2**			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Networks, the Internet and Technology in the World	**Knowledge**	• Technology is all around us, e.g. tablets, televisions and toys. • Technology is used at home and in other	• Technology is anything that helps people do a job, and it is used in school, at home,	• Many different devices connect to the internet, e.g. computers, tablets, toys and TVs, and the internet is used	• People communicate in digital and non-digital ways, each with advantages and disadvantages.	• Search engines are a service that uses the internet to find web pages and show results, which are ranked.	• The internet provides services such as the World Wide Web, email and collaborative tools which allow people	• The internet and the World Wide Web are different: the internet is the network of networks and the

		places, and some technology connects us to other people. **Key vocabulary:** technology, tablet, computer, camera, video call, home, school	outside and in the wider world. • Technology has a purpose, e.g. to communicate, to take photographs or to find out. • Digital books (e-books) and traditional books are different ways of sharing stories. **Key vocabulary:** technology, device, computer, tablet, purpose, e-book, use, wider world	to access websites and find information. • Voice assistants are not real people. • A website is a collection of web pages that is opened using a browser. **Key vocabulary:** internet, website, web page, browser, device, connected, search, voice assistant	• Email is an electronic version of a letter, sent and received almost instantly using an email address, through the internet. • The internet offers opportunities for communication and collaboration; emails can include attachments and be sent to more than one person. **Key vocabulary:** communication, email, inbox, attachment, address book, cc, bcc, internet, collaborate	• Artificial intelligence (AI) is technology that is already used in everyday life, e.g. in voice assistants, and can be used to create music and art. • AI has positive and negative impacts, and humans have to decide how it is used. **Key vocabulary:** search engine, web page, ranking, artificial intelligence, AI, impact, positive, negative	to communicate and share work. • Games, databases and documents can be shared online so that other people can use them. • Free apps and online services may collect and share information, such as contacts or location. **Key vocabulary:** internet, World Wide Web, collaborate, share, app, service, online, permission	web is a service that runs on it. • Computer networks connect devices; a LAN covers a small area such as a school and a WAN covers a wide area (networks content to be taught through the Purple Mash Networks unit once added). • Blogs are an internet service that people use to publish ideas and respond to others; posts can be moderated. **Key vocabulary:** network, LAN, WAN, internet, World Wide Web, router, wireless, blog, moderation, IP address
	Skills	• Use a range of technology, e.g. a tablet or camera, with adult support. • Talk about technology they use at home and at school. • Use technology to communicate with others, e.g. with a video call.	• Identify examples of technology in school and at home. • Say what a piece of technology is used for. • Compare traditional books with e-books and say what is good about each. • Use technology purposefully to create and open their own work.	• Identify a range of devices that connect to the internet. • Explain that the internet is used to access websites and find information. • Use links, tabs and buttons to navigate web pages. • Carry out simple keyword searches to find information.	• Explore the ways people can communicate with each other, digitally and non-digitally. • Open, read, reply to and compose emails using 2Email. • Send an email with an attachment, and send an email to more than one person using the address book. • Explain the advantages and disadvantages of email.	• Use a search engine and describe how it finds and ranks results. • Find examples of artificial intelligence in the real world. • Explain positive and negative impacts of AI, including its use in art and music. • Use AI tools appropriately for a given task and evaluate the result.	• Use internet services, such as collaborative databases, to share and work with others. • Share a finished product online so that others can use it, e.g. a game. • Describe how apps and online services collect information and use permissions. • Evaluate how a service helps people to collaborate.	• Explain the difference between the internet and the World Wide Web and name some services of each. • Explain the function of networks and the difference between a LAN and a WAN with examples. • Plan, write, publish and comment on blog posts, using moderation. • Discuss positive and negative uses of networks and the internet.

		EYFS	Key Stage 1		Key Stage 2			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Searching and Evaluating Digital Content	**Knowledge**	• We can ask a grown-up to help us find out about something. • We can use a tablet or a voice assistant to find pictures or information. **Key vocabulary:** find, ask, picture, search, information, help	• A search engine or voice assistant is a simple way of finding information. • The internet contains all kinds of content: things we like, things we do not like, and things that may be real or make-believe. • If something upsets or worries us, we tell a trusted adult. **Key vocabulary:** search engine, voice assistant, internet, real, make-believe, trusted adult	• Simple keywords are used in search engines to find information. • Not everything found online is true, and there is a difference between 'real' and 'made up' information. • Web pages are navigated using links, tabs and buttons. **Key vocabulary:** keyword, search engine, link, tab, button, true, real, made up	• Key phrases are used in search engines, and features such as autocomplete suggest words. • The internet can be used to buy and sell things. • There is a difference between beliefs, opinions and facts. **Key vocabulary:** key phrase, autocomplete, fact, opinion, belief, search results, website	• Search engines are used to find relevant information, and results are ranked. • Clear keywords and techniques such as quotation marks refine a search. • Web pages must be evaluated for clues about whether the information is true and reliable; search results can be tailored to the searcher. **Key vocabulary:** search engine, keyword, quotation marks, ranking, reliable, trustworthy, result, algorithm	• Search technologies, such as voice search, have benefits and limitations. • Being sceptical means deciding whether content is trustworthy, e.g. by checking who made it and why. • Adverts, targeted content and sponsorship, hoaxes, fake news and stereotypes can influence people. **Key vocabulary:** sceptical, trustworthy, advert, sponsorship, targeted, hoax, fake news, stereotype	• Search engines use algorithms to rank results. • Opinions can be presented as facts, and persuasion, manipulation and targeted advertising can influence choices. • Misinformation is false information shared by mistake and disinformation is false information shared on purpose; repeated information may still be untrue. **Key vocabulary:** ranking, reliable, bias, persuasion, manipulation, misinformation, disinformation, source
	Skills	• Ask an adult to help them find information or pictures. • Use a tablet or computer to look at pictures and videos. • Talk about what they have found.	• Practise using simple digital tools, e.g. a search engine or voice assistant, to find information. • Ask a trusted adult for help if they see something that upsets or worries them. • Talk about whether something is real or make-believe. • Use the Purple Mash search to find a resource.	• Use simple keywords in a search engine. • Navigate web pages using links, tabs and buttons. • Discuss whether information is real or made up and not everything online is true. • Find specific information using a simple keyword search.	• Use key phrases in search engines and use autocomplete. • Discuss how the internet can be used to buy and sell. • Tell the difference between beliefs, opinions and facts. • Talk about how to get help if they see something upsetting.	• Use a search engine to obtain focused results using clear keywords. • Refine a search, e.g. with quotation marks. • Analyse a web page for clues about reliability. • Appreciate how results are selected and ranked.	• Compare different search technologies, e.g. voice search, and their limitations. • Decide whether content is trustworthy and explain why. • Recognise adverts, targeted content, sponsorship and hoaxes. • Think carefully before sharing content.	• Explain in more depth how search engines work and how results are ranked. • Evaluate information for reliability and separate opinions presented as facts. • Identify persuasion, manipulation and targeted advertising. • Explain the difference between misinformation and disinformation.

		EYFS	Key Stage 1		Key Stage 2			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Creating and Presenting Digital Content	**Knowledge**	• Devices can be used to take photographs, draw and record sounds. • We can make and share our own pictures, sounds and stories using technology. **Key vocabulary:** photograph, draw, record, sound, picture, screen, mouse, tablet	• Digital content such as pictures, music, text and sound can be created with software, and work should be saved in the Work area so it can be found again (Introduction to Purple Mash, 2Dos). • Creative tools, e.g. 2DIY, can be used to paint, make jigsaws and make drag and drop games (Creative Computing). • E-book software (2Create a Story) allows images, text, sound, backgrounds, animation and new pages to be added (Animated Stories). **Key vocabulary:** save, open, log in, 2Dos, 2Create a Story, e-book, image, sound, animation, page, background	• A drawing program has palettes of colours and a range of painting effects that can be combined (2Paint a Picture, Creating Pictures). • Brush size, colour intensity (dilute), outlines, fill, pattern and eCollage tools are used to create and change pictures. • Digital art can imitate the style of famous artists, and work can be saved in a portfolio. **Key vocabulary:** palette, effect, brush, dilute, outline, fill, pattern, eCollage, style, portfolio	• Different software is suited to different tasks, e.g. 2Calculate for spreadsheets, 2Email for messages and 2Question for branching databases. • Content such as text, pictures and files can be combined, e.g. attached to an email. • Creating content for an audience means thinking about who will read it and keeping the information safe. **Key vocabulary:** software, tool, task, attachment, compose, draft, audience, file, combine	• Hand-drawn animations (e.g. a flick book) and animation software (2Animate) both make drawings appear to move; onion skinning shows the previous frame to speed up the process; storyboards plan an animation. • Stop-frame animation uses a series of photographs taken with small movements. • Sound can be added to an animation, and different digital products can be combined, e.g. music inserted into an animation. **Key vocabulary:** animation, frame, onion skinning, storyboard, stop-frame, 2Animate, sound, combine, flick book	• A word processor can be used to create documents; text can be formatted, and images, tables, columns, text boxes and hyperlinks can be added. • Templates help create documents more quickly and consistently. • Images should be used with permission, referencing the owner of the image. • A game needs a design document, a scene and a playable route before it is made, and must be evaluated and improved (Game Creator). **Key vocabulary:** format, template, heading, table, columns, hyperlink, crop, reference, design document, sprite, playability	• A blog is an online way of displaying thoughts and ideas, with a theme, structure and posts that others can comment on; blogs are moderated. • Planning, e.g. using a concept map, comes before writing and publishing a post. • Digital artefacts, e.g. graphs, can be included within other pieces of work, such as a blog. • Skills in one tool can be transferred to other tools, including industry standard software. **Key vocabulary:** blog, post, theme, structure, publish, moderation, comment, concept map, audience, artefact, transferable
	Skills	• Take photographs and make pictures, sounds and short recordings using	• Log in and out of Purple Mash, open and use 2Dos, and save	• Use 2Paint a Picture to explore painting effects, brush size, dilute,	• Choose suitable software for a given task from the Purple Mash tools.	• Create a flick book and a simple moving animation in 2Animate.	• Create, format and edit a document using a word processor, adding	• Plan a blog post using a concept map, write, edit and publish it.

		devices, with support. • Use a mouse, touchscreen or tablet with growing control. • Share what they have made and say how they made it. • Add their own ideas to a digital creation.	work in the Work area. • Use 2DIY tools to paint, make a jigsaw and make a drag and drop game. • Create an e-book in 2Create a Story, using pens, text, backgrounds, animation and sound, and add new pages. • Open previously saved work and make changes to it.	outline, fill and pattern tools. • Create pictures in the style of a famous artist, using a range of tools. • Create a portfolio of digital art, saving and opening their work. • Say what they like about their own work and others' work.	• Compose, reply to and send emails with attachments. • Use a draft to review a message before sending it. • Present information with a purpose, thinking about the reader.	• Use onion skinning and a storyboard to plan and make an animation. • Create a simple stop-frame animation and add sound. • Present their animation to an audience, and combine digital products.	images, tables and hyperlinks. • Use a template and the spell check, and reference the owner of images. • Design and make a game, adding sprites, quest items, music and levels, and write instructions for players. • Share a game and evaluate peers' games against criteria.	• Review and comment on others' blog posts appropriately. • Include their own digital artefacts, e.g. a graph, within a larger piece of work. • Select and use software with sophistication and transfer skills to other tools.
		EYFS	**Key Stage 1**		**Key Stage 2**			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Data and Databases	**Knowledge**	• Things can be sorted into groups, e.g. by colour or size. • We can count things and show how many on a simple chart.	• Things can be sorted and grouped by features they share (a maths link). • Information can be collected and shown in a simple way, such as a tally or a block graph (a maths link). • Data is information, such as a number or word, that we collect.	• Pictograms answer simple questions but are limited in what they can show. • A database is a computerised system that makes it easy to store, search and select information; it contains records, which are made up of fields. • Yes/no questions separate information into two groups; a binary tree sorts information in this way (Questioning).	• A database is a collection of data organised so that it can be searched and information found easily, e.g. the school's attendance database. • A branching database uses yes/no (binary) questions to sort objects and has a branching structure; each question has two possible answers. • Branching databases must be planned, tested and debugged so that they work as intended (Branching Databases).	• Data in a program can be stored in variables, e.g. a score (see Programming). • Revisit: a database holds records made up of fields, and branching databases sort using yes/no questions. • Search engines and AI use very large amounts of data.	• Databases are used to search for information; records and fields can be entered using words, numbers and drop-down menus. • Data can be sorted, grouped and searched, and displayed in tables or charts. • A collaborative database lets many people add their own data and use the results to answer questions.	• Large (big) data sets can be filtered, sorted and grouped to find specific information, and tables can be linked (Data Detectives). • Different graphs suit different types of data; graphs can show multiple datasets, and can be used to solve a problem (Graphing). • Graphs and data can be exported and imported between programs.

		Key vocabulary: sort, group, same, different, count, how many, chart	**Key vocabulary:** data, sort, group, count, feature, same, different, graph	**Key vocabulary:** pictogram, tally chart, data, database, record, field, question, binary tree, yes/no	**Key vocabulary:** database, branching database, binary question, record, branch, test, debug, 2Question	**Key vocabulary:** data, variable, record, field, search, sort, store	**Key vocabulary:** database, record, field, query, search, sort, group, collaborative, table, chart, drop-down menu	**Key vocabulary:** filter, sort, group, link, big data, graph, dataset, bar chart, line graph, pie chart, axis, export, import
	Skills	• Sort and match objects, and talk about how they have grouped them. • Count and show how many using objects or a simple chart. • Answer simple questions about what they have counted.	• Sort and group things by their features. • Collect data by counting, record it and say what it shows. • Answer simple questions using what they have found out.	• Create a class pictogram with 2Count and decide what questions it can and cannot answer. • Create suitable yes/no questions and use them to find records or objects. • Open a 2Investigate database, identify records and fields and search using the find tool, including more than one search term. • Design a simple binary tree to sort information.	• Explain what a database is and give examples of how it is used. • Sort physical objects using appropriate yes/no questions. • Plan and create a branching database using 2Question, with questions, answers and final answer cards. • Test every branch of the database, identify errors and fix them.	• Use data about their own programs, e.g. a score variable, to see how a program is working. • Use knowledge of databases and yes/no questions to help sort and search for information. • Talk about how search engines and AI use data. • Explain how information was found.	• Open and search an existing database in 2Investigate and enter data using drop-down menus. • Sort, group and arrange information, and display it in tables and charts. • Create a database with a suitable topic, appropriate fields and at least eight records. • Write questions for peers and answer questions using a database made by others.	• Filter, sort and group data in a database, and link tables, to answer questions. • Choose the most suitable graph type for data and explain why. • Include multiple datasets in a graph, label it accurately and use it to solve a problem. • Export and import files between programs.
		EYFS	**Key Stage 1**		**Key Stage 2**			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Spreadsheets and Modelling	**Knowledge**	• We can make marks and count to record how many. • Numbers and pictures can be shown in rows and columns.	• Information can be organised into rows and columns (a maths link). • Technology can help us to count and total things.	• A spreadsheet is a grid of cells in rows (across) and columns (down) in which words and numbers are entered (2Calculate, Spreadsheets). • Images can be inserted into cells and given a value,	• Every cell has its own address, using a letter (column) and a number (row), e.g. B3. • A formula calculates automatically using cell addresses and the four operators and starts with an equals sign; the	• Revisit: a spreadsheet organises numbers in cells, uses formulae and shows data in graphs. • Number data can be organised in a table for use in other subjects,	• A formula can convert units, e.g. cm to m, and be copied by dragging to neighbouring cells. • A spreadsheet can model real-life problems, e.g. area and perimeter, the results of rolling	• Industry standard spreadsheet software uses cell references and formulae to calculate automatically when data changes. • Formulae can total, find percentages and averages, maximum and

				and the totalling tool adds up the cells it covers. • Simple formulae use + and - to calculate an amount; data in a table can be used to create a block graph. **Key vocabulary:** spreadsheet, cell, row, column, 2Calculate, total, formula, graph, value	formula wizard helps to write one. • A graph is generated from the data in a sheet and updates when the data changes. • A spreadsheet can model a real-life situation, e.g. a budget. **Key vocabulary:** cell address, formula, formula wizard, operator, graph, axis, budget, total, 2Calculate	e.g. Maths and Science. **Key vocabulary:** cell, formula, table, graph, data, total, calculate	dice, or an event budget. • A hypothesis can be tested using a spreadsheet tool such as the count tool, e.g. that the letter 'e' is the most common. **Key vocabulary:** formula, model, convert, area, perimeter, variable, hypothesis, count tool, prediction	minimum values, and the series fill and flash fill save time. • A spreadsheet is a computational model of a real-life situation which can be used to plan, e.g. an event, and to solve problems. **Key vocabulary:** formula, sum, average, percentage, maximum, minimum, series fill, flash fill, model, sort, currency
	Skills	• Count and record how many using marks, objects or numbers. • Talk about numbers they have recorded. • Make a simple chart or table on a device with support.	• Sort and count objects and show the results in rows and columns. • Use technology to help count and total. • Say what a simple table or chart shows.	• Enter numbers and words into cells and navigate using arrows or a click. • Insert images into cells and use the totalling tool. • Use + and - to create simple formulae, e.g. whether coins equal a given amount. • Create a block graph from a table, adding a title and labels.	• Use cell addresses to find and select cells. • Write formulae with the formula wizard and directly in the formula bar. • Create a chart from data in a table, label the axes, and edit the data to see the chart change. • Create their own budget template and see the effect of changing amounts on the totals.	• Recall and use spreadsheet vocabulary when handling data. • Record and tidy number data in a table. • Use a spreadsheet to answer a simple question from another subject. • Explain why a spreadsheet is useful.	• Write formulae using cell references to convert units and drag them to adjoining cells. • Model a real-life situation, e.g. area and perimeter or a school event, using formulae to answer questions. • Use the count tool and graphing to investigate a hypothesis, e.g. frequency of dice rolls. • Predict results and identify patterns.	• Perform calculations with formulae using cell references, including sum, average, percentage, maximum and minimum. • Use series fill, flash fill, sort and cell formatting, e.g. currency. • Build a computational model, e.g. to plan an event, and use it to solve real-life problems. • Present data in graphs and charts, and print or share selected data.
		EYFS	**Key Stage 1**		**Key Stage 2**			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Online Safety: Behaviour,	**Knowledge**	• Be kind to others, including when using technology.	• Online bullying: be kind and respectful online; behaviour	• Online bullying: bullying can happen online and	• Online bullying: why it is important to behave	• Online bullying: bullying can	• Online bullying: online bullying can	• Online bullying: practical strategies

Relationships and Wellbeing		• Technology can be fun, but we need breaks and should tell an adult if something worries us. **Key vocabulary:** kind, safe, tell, adult, break, share, worry, rules	that upsets others is not kind. • Online relationships: ask permission before doing things online and communicate only with people you know, with adult support. • Health and wellbeing: there are rules that help us stay safe when using technology. • Self-image: just as in real life, there may be people online who could upset us; we tell a trusted adult if something worries us. **Key vocabulary:** kind, respectful, permission, trusted adult, safe, rules, worried, online, upset	can affect people's feelings; no one being bullied is to blame and help is always available. • Online relationships: people may use technology to communicate with strangers, which is risky; ask a trusted adult, and it is OK to say 'no' or 'I'm not sure'. • Health and wellbeing: technology is used in many places, with simple guidance to stay safe. • Self-image: people may look or behave differently online from in person. **Key vocabulary:** bullying, stranger, permission, say no, trusted adult, report, feelings, safe, risky	appropriately to others online; how bullying behaviour may appear and where to seek support. • Online relationships: knowing someone online is not the same as knowing them in real life, and trusting someone is not the same as liking them; it is OK to change your mind. • Health and wellbeing: too much time online can affect health, mood, sleep and relationships; age restrictions exist for a reason. • Self-image: identity can be shown in different ways online, e.g. avatars and profiles. **Key vocabulary:** appropriate, bullying, support, trust, identity, avatar, age restriction, screen time, report	happen through images, videos, texts or chats; what is posted can affect others' feelings and a person's own reputation. • Online relationships: strategies for enjoying gaming or livestreaming safely, and recognising respectful and unhealthy behaviour. • Health and wellbeing: technology can be distracting, and strategies, such as limiting screen time, help. • Self-image: online identity may differ from offline identity, people can pretend to be someone else, and kindness shapes how others see us. **Key vocabulary:** bullying, respectful, unhealthy, livestream, screen time, identity, positive, report, pretend	be different from bullying in real life, 'banter' can be hurtful, and it can be reported to helplines such as Childline. • Online relationships: emojis, memes and GIFs are forms of communication; some people may cause harm, and this is never the child's fault. • Health and wellbeing: technology can affect health positively and negatively, e.g. in-app purchases and loot boxes. • Self-image: online identities can be copied, changed or altered. **Key vocabulary:** banter, block, report, Childline, emoji, meme, community, loot box, in-app purchase, wellbeing	include capturing evidence (screenshots, URLs, profiles) and sharing it with trusted adults; how to report bullying in different contexts. • Online relationships: sharing about others needs respect for boundaries; even private sharing can have unintended consequences, and images can be risky. • Health and wellbeing: age ratings (PEGI, BBFC), persuasive design and strategies such as taking breaks and balancing screen time. • Self-image: representations of gender, race, religion or disability can be harmful or unfair, and it is important to challenge stereotypes. **Key vocabulary:** evidence, screenshot, boundary, PEGI, BBFC, persuasive design, stereotype, consent, report, unintended
	Skills	• Talk about how to be kind when using technology.	• Give examples of kind and respectful behaviour online.	• Explain what bullying is and say how it can affect someone, and	• Discuss why it is important to behave appropriately online	• Recognise when others are upset, hurt or angry	• Explore how online bullying differs from bullying in real life and use	• Use strategies to deal with online bullying, including

		• Say who they would tell if something online upset them. • Take part in simple talk about staying safe. • Follow simple rules for using devices.	• Ask a trusted adult for permission before doing things online. • Explain the rules that help them stay safe using technology at home and in other places. • Tell a trusted adult if something online worries them.	where to ask for help. • Say 'no' or 'I'm not sure' and say why this is important. • Discuss simple guidance for keeping safe using devices in different places. • Describe situations online that feel worrying and the steps to ask for help.	and how to seek support. • Discuss the difference between knowing someone online and in real life. • Identify positive and negative online activities that take up time, and follow age restrictions. • Explain how people can show themselves in different ways online.	online and say what to do. • Recognise respectful and unhealthy behaviour in online spaces. • Discuss strategies to manage screen time. • Reflect on how their online identity might differ from their offline identity.	strategies for getting help. • Collaborate positively online and support others. • Discuss how technology can affect health and wellbeing, and ask permission before spending money online. • Think about the choices they make when creating an online identity.	capturing evidence and reporting. • Think critically about the impact of sharing online, positive and negative, and how to get help if worried. • Use strategies to protect their health, e.g. taking breaks and balancing screen time. • Challenge stereotypes and reject inappropriate content, and ask for help until they get it.
		EYFS	**Key Stage 1**		**Key Stage 2**			
		Nursery/ Reception	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Online Safety: Privacy, Reputation and Ownership	**Knowledge**	• We keep our name and where we live private. • What we make belongs to us; we ask before using other people's things.	• Privacy and security: passwords help to protect information, accounts and devices; where we live or go to school should only be shared online after asking a trusted adult. • Online reputation: information shared online can stay there and be copied by others. • Copyright and ownership: creations belong to the person who made them, and saving a copy of other people's work does not make it ours.	• Privacy and security: passwords should be created and protected; 'private' information should not be shared; many devices in the home connect to the internet. • Online reputation: once information is online it can last a long time and be seen by others, so ask for help if something is shared without consent. • Copyright and ownership: content found online often belongs to other people, and	• Privacy and security: strategies for keeping passwords secure; only share information with people you know and trust; connected devices can collect personal information. • Online reputation: how to search for information about people online, and what is safe to share about yourself. • Copyright and ownership: copying someone else's work from the internet without permission is unfair.	• Privacy and security: strategies for keeping personal information private; internet use is never fully private; online services may ask for consent, and the 'digital age of consent' exists. • Online reputation: information about people may have been created, copied or shared by others and not by the person themselves. • Copyright and ownership: who owns online content such as music, images and	• Privacy and security: strong passwords; free apps and services can collect and share contacts, photos or location; app permissions. • Online reputation: searching for information about individuals and what it shows; others can form judgements, which may not be fair or correct. • Copyright and ownership: when it is acceptable to use someone else's work, and how to find content that can be reused.	• Privacy and security: managing and protecting passwords, keeping software up to date, privacy settings, scams and phishing, and terms and conditions. • Online reputation: how to build and protect a positive reputation, privacy settings and different levels of anonymity. • Copyright and ownership: searching for content that can be reused, and giving credit by referencing and acknowledging sources.

		Key vocabulary: private, name, mine, yours, ask, share, password	**Key vocabulary:** password, private, personal information, trusted adult, copy, owner, creation, save	ownership must be respected. **Key vocabulary:** password, private, secure, personal information, consent, online, owner, respect	**Key vocabulary:** password, secure, trust, personal details, reputation, permission, copyright, connected devices	videos, and whether we have the right to use it. **Key vocabulary:** privacy, consent, digital age of consent, monitored, reputation, copyright, permission, ownership	**Key vocabulary:** strong password, app permission, reputation, judgement, copyright, reuse, licence, location	**Key vocabulary:** phishing, scam, privacy settings, terms and conditions, anonymity, digital identity, reference, acknowledge, software update
	Skills	• Say what information they should keep private. • Ask before using things that belong to someone else. • Tell an adult if something online worries them. • Follow rules for taking turns and sharing devices.	• Explain why their creations belong to them and save work with a clear name. • Explain why passwords help to protect information. • Recognise personal information, e.g. where they live or go to school, and ask a trusted adult before sharing it. • Talk about what information should not be shared without checking first.	• Explain why it is important to respect ownership of content found online. • Create and protect passwords and explain what 'private' means. • Explain that information online can last a long time and be seen by others. • Ask for help if something is shared without consent or is wrong.	• Explain why copying someone else's work without permission is unfair. • Practise creating and keeping passwords secure. • Search for information about people online and think about what is safe to share. • Explain who they would tell if they feel unsure or pressured.	• Explain who owns content they find online and give examples of what must not be reused. • Use different strategies to keep personal information private. • Explain that information about people online may have been created or shared by others. • Describe what the digital age of consent means and when to ask an adult.	• Assess when it is acceptable to use someone else's work and find content that can be reused. • Create a strong password and understand app permissions. • Search for and summarise information about individuals online and explain why judgements may not be fair. • Explain how free apps can collect and share private information.	• Use search tools to find reusable content and give credit by referencing and acknowledging sources. • Manage and protect passwords, use privacy settings and keep software up to date. • Recognise scams and phishing and know how to avoid them. • Describe strategies for building a positive online reputation and safeguarding their digital identity.