

Cycle 1/2	Digital Media and communication	Network & Systems	Digital literacy and online tools	Scratch
K1	Can explore pictures and symbols on a screen with adult support. May express likes/dislikes for simple designs. Understands that images can show meaning.	Can recognise that devices like phones and tablets connect to each other to share messages or pictures. Understands that a password protects their device.	Can explore and talk about pictures and simple text. May recognise some letters or symbols and begin to choose images with support.	Can follow simple on-screen instructions. May press buttons to cause an effect. Beginning to recognise cause and effect (e.g., pressing a block makes a sprite move).
K2	Can select images or icons using a touch screen or mouse. Begins to talk about pictures and colours. Recognises simple symbols or logos (e.g., McDonald's).	Can explain that some devices use wires and others use Wi-Fi. Understands that messages go from one place to another using the internet.	Can identify letters and words on a screen. Begins to use formatting tools like bold or colour with help. Can select a relevant image with adult guidance.	Can run a basic program in Scratch. Can place a few blocks in order (e.g., move and sound). Understands that instructions must be in the correct order.
K3	Can talk about what makes a poster or image nice to look at. With help, chooses a picture to match a topic. Begins to place text or pictures on screen with support.	Can name basic parts of a network (e.g. computer, router, switch). Knows that each device needs an address to connect.	Can format short pieces of text (e.g. change size or style). Can find and insert an image with minimal support. Starts to understand that some images can't be used freely.	"Can sequence blocks to create short, purposeful animations. Begins using 'say', 'wait', and 'repeat' blocks. Can recognise a variable or loop with support."
K4	Can search for a simple image online (e.g., dog, bus) with help. Can add image to a slide or poster. Begins to choose colours and text styles with some reasoning.	Can explain how devices send messages in packets. Understands that data travels different routes and may be slowed down.	Can structure simple information with headings. Understands what a blog is. Can select appropriate images and apply basic formatting tools independently.	Can use sequences and create simple animations with events. Can use basic loops (repeat) and introduce variables (score, name) into a program.
K5	Can choose colours and fonts that fit a theme (e.g., charity, sports). Can make a simple poster or slide using a consistent layout. Understands how to use feedback to make it better.	Can describe key network hardware and explain differences between wired and wireless connections. Can explain what an IP address is and how it's used.	Can use a word processor or blog tool to create a short post. Applies formatting for clarity and selects suitable images. Can explain why they chose a source or image.	Can use selection (if/else) in a program. Uses variables to store and display values. Begins to use logical operators in decision making.
K6	Can plan and design a set of slides or digital image using clear layout and theme. Uses software to recolour logos, add images and text, and make design choices independently.	Can describe how protocols like TCP/IP allow devices to communicate. Can explain how data is split into packets and routed through a network.	Can explain what plagiarism is and credit an image source. Selects and uses formatting styles to improve readability. Can compare reliable vs unreliable sources.	Can use count-controlled iteration to repeat actions. Can decompose a problem using subroutines. Begins debugging by tracing variables and control flow.

K7	Can combine tools (e.g., align, group, recolour, insert image) to create a purposeful design. Can edit and evaluate work using criteria (e.g., readability, consistency).	Can evaluate the performance of networks, considering bandwidth and types of usage. Understands basic concepts like DNS, and explains risks like bottlenecks or dropped data.	Can evaluate sources for bias or reliability. Structures content for clarity and impact. Uses digital tools effectively to create a blog post with citations and references.	Can use both count- and condition-controlled loops effectively. Combines selection, variables, and logic to create interactive outcomes. Starts working with lists.
K8	Can use vector tools like paths, nodes, and shape modifications to create detailed digital graphics. Can give feedback to others and make thoughtful improvements to a project.	Can describe how computing systems (CPU, memory, storage) work together to execute programs. Can explain how operating systems manage these processes.	Can use copyright-friendly media, apply consistent formatting, and structure persuasive digital content. Evaluates peer work and gives constructive digital feedback.	Can develop modular programs using subroutines. Applies logical reasoning to test and debug. Uses lists to store and access multiple values in projects.
K9	Can plan, create and refine digital artefacts (e.g., presentations, logos, vector images) for a given audience and purpose. Explains how design choices affect audience understanding.	Can explain logic gates (AND, OR, NOT), how hardware is built from logic, and how data/instructions are represented using binary. Understands the basics of machine learning and AI systems.	Can write credible, well-referenced posts using valid sources. Applies advanced formatting and design for audience impact. Reflects on online influence and responsibility.	Can design complex Scratch programs combining sequences, selection, iteration, subroutines, and lists. Can explain design choices and improve code for efficiency and clarity.

Cycle 1/2	Spreadsheets	Mobile App Development	Representations	
K1	Can explore simple tables with support. May recognise that different boxes (cells) hold different bits of information.	Can explore simple apps by pressing buttons and watching responses. Recognises that tapping a screen causes something to happen.	Can recognise that symbols or signs can represent ideas (e.g., pictures for toilets, emojis). May use physical gestures or visuals to represent meaning.	
K2	Can identify rows and columns. Begins to enter data into individual cells with guidance. Recognises that calculations can be done using numbers in cells.	Can open a simple app and describe what it does. Understands that tapping different parts triggers different actions.	Can match symbols to their meanings. Understands that sounds, images, or marks can be used to send messages or represent basic information.	
K3	Can enter text and numbers into a spreadsheet. Can perform basic formatting (e.g., bold, fill colour). Uses simple formulas like =A1+B1 with support.	Can explain what an app does and how users interact with it. Begins using a visual coding environment to change basic app elements (e.g., colours, labels).	Can explain that different media (e.g., pictures, light, sound) can be used to communicate. Begins to understand that information can be stored or shared as symbols.	
K4	Can use cell references to create simple formulas. Can format a spreadsheet for readability. Uses autofill to continue a pattern or apply formulas.	Can identify and create events (e.g., on click). Can use blocks to make something happen based on user input. Begins to understand variables.	Can explain that letters and numbers can be encoded as symbols. Understands that symbols must follow rules to be understood by others.	
K5	Can apply functions like SUM and MIN to analyse data. Can format cells and apply charts with support. Begins to sort and filter data.	Can use event-driven code blocks to trigger actions. Uses variables to store values like scores or names. Begins to plan an app using basic decomposition.	Can explain how text is represented using characters and symbols. Begins to understand binary as a language used by computers.	
K6	Can independently use functions (SUM, MAX, MIN, COUNTA). Can create and label appropriate charts. Uses formatting to clearly present data.	Can create a multi-screen app using variables, selection, and user input. Can test and debug event-driven programs. Explains why an app is designed a certain way.	Can explain what a binary digit is and how bits are used to represent characters or numbers. Can convert small decimal numbers into binary.	
K7	Can use conditional formatting and sort/filter large data sets. Applies more advanced functions such as COUNTIF and AVERAGE. Analyses data effectively.	Can break down a user need into app features. Uses variables, conditions, and user feedback to drive development. Responds	Can explain how bytes and units like kilobytes and megabytes are used to measure data. Understands why binary is used in digital systems.	

		to feedback to improve design.		
K8	<p>Can use IF statements to evaluate conditions. Uses complex functions in a spreadsheet project.</p> <p>Creates charts appropriate to different audiences and purposes.</p>	<p>Can manage an app development project from planning to testing. Uses modular design, evaluates success criteria, and makes improvements based on testing.</p>	<p>Can convert between binary and decimal. Understands how text and numbers are encoded. Recognises prefixes (kilo, mega, giga) and estimates data sizes.</p>	
K9	<p>Can structure, analyse, and present large datasets using a full range of functions and formatting. Justifies chart choices and data presentation strategies for specific goals or stakeholders.</p>	<p>Can design a fully interactive app with multiple screens, user input, conditionals, variables, and event-driven logic. Evaluates effectiveness against user needs and provides justification for all features.</p>	<p>Can represent and manipulate numbers using binary. Understands how different data types are encoded and how binary is stored or transmitted using physical media (e.g., electrical signals, light, magnets).</p>	

Year 9	Python	Animation	Data Science	Representations
K1	<p>Can follow simple instructions to type text.</p> <p>May press keys to produce output and begin to notice that computers respond differently to specific letters or symbols.</p>	<p>Can explore 3D objects in a modelling environment with support. May move or delete basic shapes and observe changes.</p>	<p>Can recognise basic charts (e.g., bar or pictograms) and describe simple patterns. Can say what they see in a picture graph with support.</p>	<p>Can recognise that pictures and sounds can be stored on a computer. Enjoys exploring simple image or sound tools with support.</p>
K2	<p>Can type short commands with support. Recognises that exact typing (e.g. correct spelling and punctuation) is important in programming.</p>	<p>Can add and move 3D objects in a simple scene. Begins to change object properties such as size or colour with guidance.</p>	<p>Can sort data into groups and use simple tally charts or pictograms. Begins to recognise when a pattern is repeating or unusual.</p>	<p>Can describe how pictures are made of small parts (e.g., squares or sounds). Matches colours or patterns to digital images or sounds.</p>
K3	<p>Can write simple programs to display text. Understands that programs follow instructions in a specific order (sequence).</p>	<p>Can create a basic 3D model using primitive shapes. Can move, rotate, and scale objects. Adds a material to change colour.</p>	<p>Can create and read bar charts or pie charts with support. Can describe what a graph shows in simple terms (e.g., more, less, biggest, smallest).</p>	<p>Can explain that pictures are made from pixels and sounds are captured as recordings. Begins to link sound/picture size to quality.</p>
K4	<p>Can use input and output in simple programs. Uses variables to store text or numbers and applies arithmetic to manipulate them.</p>	<p>Can create short animations using keyframes. Understands how the timeline affects animation. Uses parenting to group objects.</p>	<p>Can describe what a data set shows and identify clear patterns. Begins to ask questions about data and test ideas using charts or tables.</p>	<p>Can describe how pictures and sounds can be broken into bits. Understands basic ideas like colour and resolution, and how quality depends on how much data is used.</p>
K5	<p>Can use selection (if/else) to control program flow. Begins to use random values and recognises how decisions can change program outcomes.</p>	<p>Can edit 3D shapes using extrude, loop cut, and face editing. Applies different colours to parts of a model. Creates a simple scene.</p>	<p>Can use digital tools to visualise and explore data sets. Can identify patterns, trends, and outliers. Understands the purpose of data visualisation.</p>	<p>Can define key terms like pixel, resolution, colour depth, and sample. Understands how image or sound quality changes when fewer bits are used.</p>
K6	<p>Can use multiple if/elif/else statements. Understands and applies count-controlled iteration (e.g. while loops) for repetition.</p>	<p>Can use organic modelling techniques (e.g., proportional editing, knife tool, subdivision). Starts to use symmetry creatively.</p>	<p>Can use the investigative cycle (e.g., problem, plan, data, analyse, conclusion) to explore a given data set. Can clean, sort, and filter data effectively.</p>	<p>Can calculate the size of a digital image or sound file using resolution/sampling formulas. Can manipulate digital media using tools like GIMP or Audacity.</p>
K7	<p>Can combine selection and iteration in programs. Uses variables to count, calculate, and track</p>	<p>Can light a scene, set up a camera, and render a basic animation. Understands render</p>	<p>Can plan a data investigation to answer a real-world question. Can select data, create</p>	<p>Can explain trade-offs in size vs quality for digital media. Can adjust settings to</p>

	outcomes. Begins using Boolean expressions as flags.	settings and their effects on output.	capture forms, clean results, and analyse using graphs.	compress or improve file quality and explain effects.
K8	Can design programs using decomposed logic. Writes programs that combine user input, loops, selection, and arithmetic. Uses iteration to refine results.	Can plan, model, animate, and render a short scene using multiple objects and techniques. Makes creative and technical decisions.	Can complete the full investigative cycle using large or messy data sets. Can present insights clearly with appropriate visualisations and conclusions.	Can compare bitmap vs vector image representation, and PCM vs MIDI sound. Explains benefits and drawbacks of each. Can edit media using layered or symbolic formats.
K9	Can write modular programs using functions and Boolean logic. Applies abstraction and iteration to solve complex problems efficiently. Understands and explains how execution flow works.	Can produce a polished 3D animation combining advanced modelling, lighting, camera work, and rendering. Evaluates and improves final output based on peer/self-assessment.	Can evaluate the reliability of a data investigation. Explains the significance of findings and proposes real-world actions based on the data.	Can explain how image and sound data is structured, compressed, and encoded. Makes decisions about format and method depending on audience or purpose. Justifies creative and technical choices.

Year 9	Cyber Security	Physical Computing		
K1	Can name common types of personal data (e.g. name, birthday). Can say that sharing information online can be risky.	Can recognise that technology can interact with the real world (e.g. lights, sounds). Enjoys pressing buttons or seeing lights flash on devices.		
K2	Can explain why we need to protect personal data. Recognises simple online risks (e.g. clicking on pop-ups or sharing passwords).	Can identify simple physical inputs (like buttons) and outputs (like LEDs). Can use a pre-made program to trigger a response (e.g. press button ,Üí light on).		
K3	Can describe what a password is for and give examples of safe passwords. Recognises that people may try to trick others into giving away data.	Can follow simple instructions to create a working program using input and output. Understands the link between action in the code and response in the real world.		
K4	Can identify different types of personal information and explain how it might be misused. Can explain what social engineering is and name examples like phishing.	Can write and test a basic program that uses built-in micro:bit features (e.g. LEDs, buttons). Begins to recognise patterns in code like input ,Üí action.		
K5	Can explain how malware works and name common types (e.g. virus, worm, ransomware). Understands the risks of weak passwords and knows how to strengthen them.	Can use inputs and outputs in a purposeful program (e.g. light sensor triggers LED). Can describe how a micro:bit program works step by step.		
K6	Can describe how a DDoS attack works and how it affects systems. Can compare different malware types and their impact on users and networks.	Can connect external components (e.g. speaker, LED, switch) and write code to control them. Can debug errors with support. Starts to use repetition in programs.		
K7	Can explain technical methods used to protect against cyberattacks, such as firewalls, anti-malware, and authentication	Can design a simple interactive system using multiple components and programming constructs (e.g. if, loop, variable).		

	systems. Understands the role of laws like the Data Protection Act and the Computer Misuse Act.	Tests and improves code independently.		
K8	Can evaluate how different security measures protect data. Can explain the ethical and legal consequences of unauthorised access or data misuse. Can assess how human error contributes to cybersecurity risks.	Can plan, build, and document a complete physical computing project. Uses decomposition and feedback to refine work. Uses radio or GPIO pins for communication.		
K9	Can critically analyse real-world cybersecurity threats and propose balanced defence strategies. Can evaluate the societal implications of data breaches and cybercrime, and suggest how individuals and organisations should respond.	Can build a reliable, well-tested system that solves a real-world problem. Applies structured design thinking, explains trade-offs, and justifies programming and hardware decisions.		