



## Computing Curriculum Map 2025-2026

Subject Intent								
<p>We aim to foster computational thinking and digital creativity, equipping students with lifelong skills in Computing, ICT, and Digital Literacy. Our curriculum focuses on core principles like algorithms, programming, and problem-solving, while encouraging logical, creative, and critical thinking. Students learn how digital systems work, apply computational thinking through coding, and develop workplace-ready skills. They also explore technology's societal impact and build transferable skills like collaboration and research. We prioritise internet safety for young people, protecting their physical and emotional wellbeing as they navigate the online world.</p>								
Key Stage 2	Year	HT1	HT2	HT3	HT4	HT5	HT6	Beyond Elton
<p>By the end of KS2, students should:</p> <ul style="list-style-type: none"> <li>- Design, debug, and improve programs using sequences, loops, and conditions to solve problems.</li> <li>- Understand algorithms, fix errors, and explain how code works logically.</li> <li>- Explore how networks (like the internet) enable communication and collaboration.</li> <li>- Use search tools critically and evaluate online information.</li> <li>- Combine software to create projects, analyse data, and present findings.</li> <li>- Use technology safely, spotting risks and knowing how to report concerns.</li> </ul>	7	Digital Induction and core skills: Digital onboarding, Logging in securely, School email & Google Classroom, File management, Keyboard skills, Mouse control & shortcuts Hardware & software	Spy School (Excel)	Impact of Technology	Understanding Computers	Binary/Computational Thinking	Scratch/ Introduction to Python	<p><b>Key Stage 5</b> Our curriculum builds foundations (algorithms, binary, networks) for A-Level study (OOP, data structures, advanced theory, NEA project) at both Bury College and Holy Cross</p> <p><b>Careers</b> Potential career paths include: software developer, programmer, IT technician, IT support, web developer, data analyst, cybersecurity specialist as well as many tech apprenticeships</p>
	8	My Digital World/Advanced Email/Google Classroom	Binary/Denary / Hexadecimal/ Image Editing	Digital Safety (Safely using new technologies)/ Micro bits	Computer Networks	Website Design	Python Programming	
	9	E-Safety: Cyber Bullying/Grooming /Sending Nudes /Selfies	Cyber Security/ Advanced Python	Databases	Digital Graphics/Streaml istic Project	Spreadsheets	3D Printing and App Development	
	10	Architecture of the CPU/Programming /Computer Memory/Secondary Storage/Cables	Types and topologies of Networks/ Programming	Data Representation (binary, hexadecimal, data storage (text, images, sound),	Programming/ System Security threats and prevention /cybersecurity measures.	System Software (operating systems, utility software, user interfaces) / ELCE	Algorithms/ Algorithmic Thinking/Pse udocode/Inp uts, Processes & Outputs	

				file compression, logic gates.				
	11	Sorting Algorithms (bubble,merge, insertion)/Producing Robust Programs	Computational Thinking / Programming Languages and IDE	Revision	Revision	Revision		

## Year 7

### Overview of the Year

Year 7 ensures all students can confidently operate within a desktop computing environment before developing secure foundations in computational thinking, data representation and programming. Greater curriculum time is allocated to embedding keyboard fluency, digital organisation and Python fundamentals.

Autumn 1	Autumn 2	Spring 1
<b>Content:</b> Digital Induction & Core Skills	<b>Content:</b> Spy School (Excel) Unit of work designed to introduce Year 7 students to spreadsheets.	<b>Content:</b> Impact of Technology unit
<b>Concepts/Generalisations/Skills</b> <ul style="list-style-type: none"> <li>• Logging in securely</li> <li>• School email &amp; Google Classroom</li> <li>• File management</li> <li>• Keyboard skills, mouse control &amp; shortcuts</li> <li>• Hardware &amp; software</li> <li>• Using IDEA</li> </ul> In addition: <ul style="list-style-type: none"> <li>• Whole school: Bullying survey</li> </ul>	<b>Concepts/Generalisations/Skills</b> <ul style="list-style-type: none"> <li>• Basics: Entering data, identifying cell references, and understanding text vs. number alignment.</li> <li>• Calculations: Creating formulae (+, -, *, /), using brackets (BODMAS), and using SUM, AVERAGE, MIN, and MAX functions.</li> <li>• Efficiency: Using AutoFill to replicate work and absolute references (\$A\$1) to lock cells.</li> </ul>	<b>Concepts/Generalisations/Skills</b> <ul style="list-style-type: none"> <li>• Network Navigation: Logging on securely, finding files, and launching applications independently.</li> <li>• Digital Safety: Creating strong passwords, understanding school rules, and protecting online identity.</li> <li>• Email Communication: Recognising respectful messages and constructing effective emails for the right audience.</li> </ul>

<ul style="list-style-type: none"> <li>• BeeWell Questionnaire for Greater Manchester Combined Authority schools</li> <li>• Bebras Challenge</li> </ul>	<ul style="list-style-type: none"> <li>• Presentation: Formatting with merged cells, currency, column widths, and conditional formatting.</li> <li>• Data Handling: Sorting, filtering, and using comparison operators (&gt;, &lt;, =) to find specific data.</li> <li>• Visualisation: Creating bar and pie charts from data to answer questions.</li> <li>• Application: Building spreadsheets from scratch and modelling "what-if" scenarios.</li> </ul>	<ul style="list-style-type: none"> <li>• Online Conduct: Describing safe peer communication and verifying who you are talking to.</li> <li>• Cyberbullying Awareness: Defining cyberbullying and explaining its effects on others.</li> <li>• Presentation Skills: Planning and creating digital content tailored to a specific audience.</li> <li>• Generalisation: Evaluating digital data for trustworthiness and design.</li> <li>• Generalisation: Safe, respectful, and responsible online behaviour requires vigilance about identity, privacy, and knowing how to report concerns.</li> </ul>
<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>
<p><b>Revisit/Review</b></p> <p>This foundational unit introduces students to the school's digital environment. It builds upon Key Stage 2 expectations by ensuring all students, regardless of prior experience, can securely log in, manage files, use email/Google Classroom, and understand basic hardware/software. This establishes a baseline of digital competency required for all future Computing projects.</p>	<p><b>Revisit/Review</b></p> <p>Students apply their secure login and file management skills from Autumn 1 to access and save spreadsheet work. They build on basic number and data handling from KS2 Maths by learning to enter data, create formulae (using +, -, *, /), and use functions (SUM, AVERAGE). This introduces them to using software for data analysis, a skill revisited in Year 8 Spreadsheets and Year 9 Databases.</p>	<p><b>Revisit/Review</b></p> <p>Students revisit and deepen their understanding of online safety, first encountered in KS2. They build on the concept of a digital footprint by learning about secure passwords, school rules, and respectful online communication. The unit's focus on evaluating digital content for trustworthiness links directly to the Information Literacy skills taught in Year 8's "My Digital World" unit.</p>

Year 7		
Spring 2	Summer 1	Summer 2
<p><b>Content:</b> Understanding Computers</p>	<p><b>Content:</b> Binary / Computational Thinking</p>	<p><b>Content:</b> Scratch/ Introduction to Python</p>
<p><b>Concepts/Generalisations/Skills</b></p> <ul style="list-style-type: none"> <li>• Digital Security: Creating strong passwords to protect accounts and data.</li> <li>• File Management: Navigating and understanding how File Explorer organises storage.</li> <li>• Hardware Classification: Distinguishing between input/output devices and internal/external components with examples.</li> <li>• Storage Types: Describing the differences between magnetic, optical, and solid-state storage.</li> <li>• Processing Concepts: Explaining the CPU's role, the Fetch-Execute Cycle, and its relationship with RAM.</li> <li>• Performance Factors: Discussing how clock speed and the number of cores affect processing power.</li> <li>• System Operations: Understanding multi-tasking and the distinction between the Operating System and system software.</li> <li>• Data Management: Explaining the importance, reasons, and effects of backing up data.</li> </ul>	<p><b>Concepts/Generalisations/Skills</b></p> <p><b>Binary:</b></p> <ul style="list-style-type: none"> <li>• Binary Fundamentals: Explaining why computers use binary and the difference between base-2 (binary) and base-10 (denary) systems.</li> <li>• Data Units: Identifying and ordering units of data from bit and nibble through to kilobyte, megabyte, and gigabyte.</li> <li>• Number Conversion: Converting binary numbers into denary and denary numbers into binary.</li> <li>• Character Representation: Explaining how characters are stored digitally using the ASCII character set.</li> <li>• Core Principle: Understanding that all data in a computer is represented and manipulated digitally as binary digits.</li> </ul> <p><b>Computational Thinking:</b></p> <ul style="list-style-type: none"> <li>• Algorithm Definition: Understanding that an algorithm is a set of step-by-step instructions to solve a problem.</li> <li>• Design Methods: Using flowcharts and pseudocode to plan and represent algorithms visually and textually.</li> </ul>	<p><b>Concepts/Generalisations/Skills</b></p> <p><b>Scratch</b></p> <ul style="list-style-type: none"> <li>• Planning: Drawing algorithms and creating flow diagrams to plan out code before building it.</li> <li>• Control Structures: Implementing loops for repetition.</li> <li>• Conditional Logic: Using "If...Then...Else" statements to make decisions in code.</li> <li>• Data Handling: Creating and using variables to store and change data.</li> <li>• Interactivity: Using broadcast messages to trigger events between different parts of the program.</li> </ul> <p>Application &amp; Game Development Skills:</p> <ul style="list-style-type: none"> <li>• Geometry Integration: Recognizing basic shapes and their number of sides to apply to programming projects.</li> <li>• Project-Based Practice: Practising skills by making several different types of games.</li> <li>• The Design Cycle: Planning, creating, and finally evaluating a self-made project</li> </ul> <p><b>Introduction to Python</b></p> <ul style="list-style-type: none"> <li>• Input/Output: Writing programs that display messages and receive keyboard input from the user.</li> </ul>

<ul style="list-style-type: none"><li>• Generalisation: A computer functions through the interaction of hardware components, each with a specific role.</li><li>• Generalisation: Performance is determined by component specifications, and data loss can be mitigated through regular backup practices.</li></ul>	<ul style="list-style-type: none"><li>• Core Techniques: Applying sequence (order of steps) and selection (making decisions) within algorithms.</li><li>• Logic &amp; Conditions: Using Boolean statements (&lt;, &gt;, =) to create conditions that control program flow.</li><li>• Data Storage: Using variables to store and retrieve information for later use within an algorithm. Fundamental Principle: Algorithms are the logical foundation upon which computer programs are built.</li></ul>	<ul style="list-style-type: none"><li>• Data Storage: Using variables, including applying simple arithmetic expressions in assignment statements (e.g., total = price + tax).</li><li>• Data Structures:<ul style="list-style-type: none"><li>○ Creating and managing lists (accessing individual items, performing operations on items or whole lists).</li><li>○ Performing common operations on strings (text) and individual characters.</li></ul></li></ul> <p>Program Control &amp; Logic:</p> <ul style="list-style-type: none"><li>• Selection: Using if-elif-else statements to make decisions and control the flow of the program.</li><li>• Iteration (Loops):<ul style="list-style-type: none"><li>○ Using while statements for repetition based on a condition.</li><li>○ Using for loops to iterate over sequences (like list items or characters in a string).</li></ul></li></ul> <p>Problem-Solving &amp; Development Skills:</p> <ul style="list-style-type: none"><li>• Debugging: Locating and correcting common syntax errors.</li><li>• Algorithm Design: Using variables to keep track of counts (e.g., count = count + 1) and sums (e.g., total = total + number).</li><li>• Synthesis &amp; Application: Combining all the key programming features (input/output, variables, lists, strings, selection, and iteration) to develop complete solutions to meaningful problems.</li></ul>
---	--	--

<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>
<p><b>Revisit/Review</b></p> <p>This unit builds directly on the "Hardware &amp; software" introduction from Autumn 1. Students now deepen that knowledge by classifying input/output devices, distinguishing between storage types (magnetic, optical, solid-state), and explaining the CPU's role (Fetch-Execute Cycle). This theoretical foundation prepares them for more complex topics like computer networks in Year 8 and system architecture in Year 10.</p>	<p><b>Revisit/Review</b></p> <p>Students apply their understanding that computers process data (from Spring 2) to learn <i>how</i> they do it using binary. This builds on KS2 Maths (number bases) by introducing base-2. They are formally introduced to algorithms, building on the logical thinking used in Scratch (KS2) and the step-by-step problem-solving implicitly used in Excel (Autumn 2). This unit creates the foundation for all future programming and computational thinking work.</p>	<p><b>Revisit/Review</b></p> <p>This unit directly applies the computational thinking concepts (algorithms, sequence, selection) taught in Summer 1. It builds on any KS2 programming experience by introducing a more structured approach to planning (flow diagrams) and a new, text-based language (Python). The Scratch block-based work provides a visual bridge to the syntax of Python, preparing students for text-based programming in Year 8.</p>

## Year 8

### Overview of the Year

In Year 8, students develop digital literacy by learning to evaluate online content, use email and classroom tools responsibly, and understand online safety. They explore computer science fundamentals including number systems (binary/hex), image representation, computer networks, and physical computing with Micro:bits. Finally, they apply their programming knowledge to create websites using Dreamweaver and develop more complex Python programs using sequences of data like lists and strings.

Autumn 1	Autumn 2	Spring 1
<p><b>Content:</b> My Digital World unit/ Advanced email and Google classroom</p>	<p><b>Content:</b> Binary/ Denary/ Hexadecimal/ Image Editing</p>	<p><b>Content:</b> Digital Safety/ Microbits</p>
<p><b>Concepts/Generalisations/Skills</b> <b>My Digital World unit</b> Information Literacy &amp; Research Skills:</p> <ul style="list-style-type: none"><li>Evaluating Sources: Knowing how to assess the credibility and reliability of websites.</li><li>Understanding Search Technology: Learning how search engines function to find and rank information.</li><li>Advanced Search Strategies: Applying tools and techniques to "search smarter" by filtering results to increase relevance and reduce the volume of irrelevant information.</li></ul> <p>Legal &amp; Ethical Use of Digital Content:</p> <ul style="list-style-type: none"><li>Copyright Awareness: Understanding the legal and ethical principles of using online resources and others' ideas without infringing on copyright law.</li></ul> <p>Personal Safety &amp; Digital Citizenship:</p>	<p><b>Concepts/Generalisations/Skills</b> <b>Binary/ Denary/ Hexadecimal</b></p> <ul style="list-style-type: none"><li>Number Systems &amp; Conversion:<ul style="list-style-type: none"><li>Recalling binary (Base 2) and denary (Base 10) conversion techniques.</li><li>Applying understanding to hexadecimal (Base 16).</li></ul></li><li>Binary Operations:<ul style="list-style-type: none"><li>Adding two binary numbers and understanding the concept of an overflow error.</li></ul></li><li>Data Encoding:<ul style="list-style-type: none"><li>Understanding the use and purpose of ASCII and Unicode for text.</li><li>Explaining how images are stored digitally.</li><li>Understanding the relationship between resolution, colour depth, and file size.</li></ul></li></ul>	<p><b>Concepts/Generalisations/Skills</b> <b>Digital Safety</b> Key Concepts</p> <ul style="list-style-type: none"><li>Recognising a range of online dangers</li><li>Understanding the risks and consequences of unsafe online behaviour</li><li>Learning practical ways to stay safe online</li><li>Understanding the importance of E-Safety</li></ul> <p>Key Skills Developed:</p> <ul style="list-style-type: none"><li>Identifying and assessing online risks</li><li>Problem-solving ways to avoid dangers</li><li>Collaboration and discussion skills</li><li>Creative communication through an E-Safety poster</li><li>Developing digital literacy and responsible online behaviour</li></ul> <p>Learning Outcomes</p> <ul style="list-style-type: none"><li>Name multiple online dangers</li></ul>

<ul style="list-style-type: none"> <li>• Risk Awareness: Identifying potential dangers online and understanding strategies to avoid them.</li> <li>• Handling Abuse: Learning the procedures for documenting (evidencing) online abuse and knowing the steps to take to stop it, whether one is the victim or a witness.</li> <li>• Advanced email and Google classroom skills</li> <li>• Using IDEA</li> </ul> <p>In addition:</p> <ul style="list-style-type: none"> <li>• Whole school: Bullying survey</li> <li>• BeeWell Questionnaire for Greater Manchester Combined Authority schools</li> <li>• Bebras Challenge</li> </ul>	<p><b>Image Editing:</b></p> <ul style="list-style-type: none"> <li>• Learning techniques for customising and improving images using Photoshop.</li> <li>• Applying commonly used tools to enhance the appearance of images.</li> <li>• Understanding the term "airbrushing" and identifying where it has been used.</li> <li>• Exploring the reasons why images are airbrushed.</li> <li>• Discussing the ethics of manipulating images and the impact on society</li> </ul>	<ul style="list-style-type: none"> <li>• Explain how to avoid them</li> <li>• Produce an E-Safety poster outlining dangers and prevention strategies</li> </ul> <p><b>Microbits</b></p> <p>Key Concepts</p> <ul style="list-style-type: none"> <li>• Physical Computing: Devices use inputs (sensors), processes, and outputs.</li> <li>• Core Constructs: Programs rely on sequence (order), selection (if...then...else), and variables (storing data).</li> <li>• Sensing: Physical movement and direction are detected by sensors (accelerometer, compass) and used as input.</li> </ul> <p>Key Skills</p> <ul style="list-style-type: none"> <li>• Programming: Creating code in MakeCode using selection, variables, comparison operators, and random numbers.</li> <li>• Hardware: Flashing programs to a micro:bit, using buttons, accelerometer, and compass.</li> <li>• Design &amp; Debugging: Designing algorithms, tracing program flow, testing on emulators and devices, and fixing bugs.</li> </ul> <p>Key Generalizations</p> <ul style="list-style-type: none"> <li>• Order matters: The sequence of instructions determines the outcome.</li> <li>• Variables are dynamic: They store values that can be changed by inputs, but checking them does not alter their value.</li> <li>• Abstraction helps: Breaking a project into task, algorithm, and code levels makes complexity manageable.</li> <li>• Testing is essential: Programs behave differently on emulators versus physical</li> </ul>
--	---	---

		hardware due to real-world factors like sensor sensitivity.
<b>Assessment</b> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<b>Assessment</b> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<b>Assessment</b> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>
<b>Revisit/Review</b> Students revisit and significantly extend the core digital skills from Year 7 Autumn 1. They build on basic searching by learning how search engines work and applying advanced search strategies. They deepen their understanding of online safety (Year 7 Spring 1) by learning to evaluate website credibility, understand copyright, and handle online abuse. This unit professionalises their digital literacy for more complex project work.	<b>Revisit/Review</b> Students build directly on the binary and data representation concepts from Year 7 Summer 1. They extend their knowledge to hexadecimal and explore how images are stored digitally (pixels, colour depth, resolution), linking theory (binary) to practice (image files). The Photoshop skills provide a practical application of these concepts, while also revisiting file management (Year 7 Autumn 1) and introducing professional design software.	<b>Revisit/Review</b> Students revisit online safety (Year 7 Spring 1, Year 8 Autumn 1) with a sharper focus on current risks and practical prevention (E-Safety poster). The Microbit unit applies the programming constructs from Year 7 Summer 2 (sequence, selection, variables) to physical computing. They build on the concept of "sensing" from Year 7 (input devices) by programming the Microbit's sensors (accelerometer, compass) to control outputs.

Year 8		
Spring 2	Summer 1	Summer 2
<p><b>Content:</b> Computer Networks</p>	<p><b>Content:</b> Website Design</p>	<p><b>Content:</b> Python Programming</p>
<p><b>Concepts/Generalisations/Skills</b></p> <ul style="list-style-type: none"> <li>To understand what is meant by a Local Area Network (LAN).</li> <li>Hardware required</li> <li>Methods of connection (WiFi &amp; Ethernet cable)</li> <li>To understand what is meant by a Wide Area Network (WAN).</li> <li>Methods to connect (GPRS, mobile, telephone cables &amp; satellite).</li> <li>To understand how peripheral devices can connect to a computer system using a Wireless Personal Area Network (WPAN).</li> <li>Methods to connect (Bluetooth).</li> <li>Understand the following methods of keeping you computer secure on a computer network:</li> <li>Firewall, anti-malware, passwords &amp; encryption.</li> </ul>	<p><b>Concepts/Generalisations/Skills</b></p> <p>Key Concepts</p> <ul style="list-style-type: none"> <li><b>Purpose &amp; Audience:</b> Websites are created for specific reasons (inform, sell) and must be tailored to a target audience.</li> <li><b>House Style &amp; Navigation:</b> Consistent colours, fonts, and predictable menus are essential for a professional, usable site.</li> <li><b>Development Cycle:</b> Building a website follows a cycle of Plan → Create → Test → Evaluate.</li> </ul> <p>Key Skills</p> <ul style="list-style-type: none"> <li><b>Analysis:</b> Critiquing websites to identify strengths and weaknesses.</li> <li><b>Design:</b> Planning page content, layout, and house style before building.</li> <li><b>Technical:</b> Creating pages in Dreamweaver, formatting text, inserting images, and using tables for layout.</li> <li><b>Linking:</b> Creating hyperlinks using text, buttons, and image hotspots.</li> <li><b>Testing:</b> Previewing pages, checking all links, and proofreading content.</li> </ul>	<p><b>Concepts/Generalisations/Skills</b></p> <p>Key Concepts</p> <ul style="list-style-type: none"> <li><b>Lists as Data Structures:</b> Ordered collections of items stored in memory, accessed via zero-based indices.</li> <li><b>Iteration:</b> Repeating actions using for loops (for sequences) and while loops (while a condition is true).</li> <li><b>Mutability:</b> Lists can be changed after creation using methods like .append(), .insert(), and .remove().</li> <li><b>Strings as Sequences:</b> Strings are sequences of characters supporting indexing, len(), in, and iteration.</li> <li><b>Program Control:</b> Using selection (if-elif-else) to make decisions based on conditions.</li> </ul> <p>Key Skills</p> <ul style="list-style-type: none"> <li><b>List Operations:</b> Creating lists; accessing items by index; using .append(), .insert(), .remove(), .index(), .sort(); checking length with len(); testing membership with in.</li> <li><b>Iteration Techniques:</b> Writing for loops to process list items and strings; using while loops with conditions; combining loops with if for filtering and counting.</li> </ul>

	<ul style="list-style-type: none"> <li>• Evaluation: Giving peer feedback using tools like star graphs and reflecting on improvements.</li> </ul> <p>Key Generalizations</p> <ul style="list-style-type: none"> <li>• Consistency is key: Predictable navigation and a uniform house style improve usability.</li> <li>• Planning saves time: Detailed planning makes the building process smoother.</li> <li>• Testing is essential: All links must be checked and text proofread before a site is complete.</li> <li>• Creation is iterative: Building involves a cycle of feedback and improvement.</li> </ul>	<ul style="list-style-type: none"> <li>• String Manipulation: Accessing characters by index; iterating over strings; using len() and in.</li> <li>• Program Design: Using pair programming; tracing code execution; processing real-world datasets.</li> </ul> <p>Key Generalizations</p> <ul style="list-style-type: none"> <li>• Lists organise related data for efficient access and processing.</li> <li>• Loops eliminate repetitive code by automating iteration over sequences.</li> <li>• Zero-based indexing means the first item is at position 0, requiring careful index calculation.</li> <li>• Strings are character sequences that support list-like operations.</li> <li>• Selection enables decision-making by directing program flow based on conditions.</li> <li>• Real-world datasets (word lists, medical data) can be processed using lists and loops.</li> </ul>
<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>
<p><b>Revisit/Review</b></p> <p>This unit provides a deeper, more technical understanding of how computers communicate, building on the concept of the internet introduced in KS2. It connects to prior learning on hardware (Year 7 Spring 2) by exploring the specific hardware required for networks. It also links to online safety (Year 7 Spring 1, Year 8 Spring 1) by introducing network security methods like firewalls and encryption.</p>	<p><b>Revisit/Review</b></p> <p>Students apply their understanding of audience and purpose (from Year 7 Spring 1 and Year 8 Autumn 1) to design a website. They build on file management skills (Year 7 Autumn 1) by organising website assets. The process of planning, creating, testing, and evaluating revisits the development cycle introduced in Scratch (Year 7 Summer 2) and applies it to a new, professional context (Dreamweaver).</p>	<p><b>Revisit/Review</b></p> <p>This unit consolidates and extends all programming knowledge from Year 7 (Scratch, Python intro) and Year 8 (Micro:bits). Students move from basic constructs to using data structures (lists) and processing strings. This builds directly on the "sequences of data" concept, preparing them for more complex programming challenges in Year 9 (Advanced Python) and GCSE (arrays, records).</p>

## Year 9

### Overview of the Year

In Year 9, students deepen their understanding of their responsibilities and safety in the digital world by exploring complex issues such as online grooming, sexting, and the legal frameworks surrounding cybercrime. They advance their programming proficiency in Python, tackling more complex problems involving string manipulation and algorithmic thinking. Students also develop practical workplace skills by learning to design professional databases, create digital graphics to a client brief, build sophisticated spreadsheets, and explore emerging technologies like 3D printing, culminating in a substantial mobile app development project that consolidates their Key Stage 3 learning.

Autumn 1	Autumn 2	Spring 1
<b>Content:</b> E-Safety: Cyber Bullying/Grooming /Sending Nudes /Selfies	<b>Content:</b> Cyber Security/Advanced Python	<b>Content:</b> Databases
<b>Concepts/Generalisations/Skills</b> <b>E-Safety</b> Key Concepts <ul style="list-style-type: none"><li>• Digital Consent: Understanding that consent applies online; pressure to send images is not consent.</li><li>• Online Grooming: Recognising that grooming involves building trust to exploit sexually, often through fake identities and manipulation.</li><li>• Digital Footprint: Once an image is shared, control is lost; "disappearing" messages can be saved permanently.</li><li>• Legal Frameworks: Multiple laws govern online behaviour. Creating, possessing, or sharing explicit images of minors (including self-taken nudes) is illegal under the Protection of Children Act 1978 and Criminal Justice Act 1988. The Online Safety</li></ul>	<b>Concepts/Generalisations/Skills</b> <b>Cyber Security</b> Key Concepts <ul style="list-style-type: none"><li>• Cyber Security Threats: Understanding the nature of hacking, malware (viruses, Trojans, logic bombs), phishing, and identity theft.</li><li>• Legal Frameworks: Understanding the key provisions of the Computer Misuse Act (unauthorised access, impairment of computer operations), Data Protection Act/GDPR (protecting personal data), and Copyright Law (protecting intellectual property).</li><li>• Data Vulnerability: Recognising that personal data placed on social networking sites can be harvested by companies and criminals.</li></ul>	<b>Concepts/Generalisations/Skills</b> <b>Databases</b> Key Concepts <ul style="list-style-type: none"><li>• Database Types: Understanding the difference between relational databases (linked multiple tables) and flat-file databases (single table).</li><li>• Table Structure: Data is organised into tables with fields and records; each table requires a primary key for unique identification.</li><li>• Data Types &amp; Relationships: Fields must have appropriate data types; forms are linked to tables and provide a user-friendly interface for data interaction.</li><li>• Query Logic: Data is extracted using criteria, logical operators (AND/OR), and wildcards; queries can combine multiple tables or perform calculations.</li></ul>

<p>Act 2023, the Sexual Offences Act 2003 and the Computer Misuse Act 1990</p> <ul style="list-style-type: none"> <li>Emotional and Legal Impact: Online abuse can lead to shame and anxiety; legal consequences can include criminal prosecution, court proceedings, and sex offender notification requirements for adult offenders.</li> </ul> <p>Key Skills</p> <ul style="list-style-type: none"> <li>Risk Prevention: Recognising grooming signs; managing privacy settings; being cautious about sharing personal information.</li> <li>Response Strategies: Knowing how to report abuse to trusted adults and platforms (e.g., using reporting tools on social media); preserving evidence.</li> <li>Critical Thinking: Evaluating online authenticity; considering legal consequences before sharing images (including that creating or forwarding indecent images of under-18s is a serious criminal offence).</li> <li>Communication and Seeking Help: Developing assertiveness to give or deny consent; knowing where to find help (e.g., child protection services, school safeguarding leads, CEOP).</li> </ul> <p>Key Generalizations</p> <ul style="list-style-type: none"> <li>Online actions have serious legal consequences: Cyberbullying, grooming, and sharing nudes can lead to criminal prosecution, court proceedings, and, for</li> </ul>	<ul style="list-style-type: none"> <li>Physical &amp; Environmental Safety: Understanding physical health hazards of computer use (RSI, backache, eyestrain) and the importance of safe e-waste disposal.</li> </ul> <p>Key Skills</p> <ul style="list-style-type: none"> <li>Threat Identification: Recognising signs of fraudulent emails (phishing) and responding appropriately.</li> <li>Data Protection: Using privacy settings effectively to protect online identity and prevent identity theft.</li> <li>Legal Compliance: Adhering to copyright law when using written text, downloading music, images, and movies.</li> <li>Impact Analysis: Describing the effects on individuals and companies of illegal downloading and copyright infringement.</li> <li>Health &amp; Safety Practice: Using computers sensibly to avoid physical hazards and describing how to safely dispose of old computer equipment.</li> </ul> <p>Key Generalizations</p> <ul style="list-style-type: none"> <li>Not everything online is trustworthy: Emails and messages can be fraudulent; personal data shared online can be harvested and misused by criminals.</li> <li>You have rights and responsibilities online: Laws exist to protect computer users (Data Protection, GDPR) and to define acceptable use (Computer Misuse Act, Copyright Law); users must understand and adhere to them.</li> <li>Safety requires proactive behaviour: Protecting yourself online requires active steps—using privacy settings, verifying</li> </ul>	<p>Key Skills</p> <ul style="list-style-type: none"> <li>Database Setup: Creating a database from scratch in Microsoft Access.</li> <li>Table Creation: Creating tables, naming fields, selecting data types, and entering data.</li> <li>Form Design: Creating forms with navigation buttons for improved user experience.</li> <li>Query Building: Using single/multiple criteria, logical operators, wildcards, multiple tables, and calculated queries.</li> <li>Reporting: Creating and formatting professional-looking reports.</li> <li>Explanation: Explaining table-form links and identifying primary keys.</li> </ul> <p>Key Generalizations</p> <ul style="list-style-type: none"> <li>Good structure ensures efficiency: Proper table design with primary keys and appropriate data types makes data storage and retrieval more effective.</li> <li>Forms and queries enhance usability: Forms simplify data interaction; queries enable powerful data filtering, combination, and analysis.</li> <li>Relational design reduces redundancy: Linking tables avoids duplicate data compared to flat-file structures.</li> </ul>
--	---	--

adults, sex offender notification requirements.

- Privacy is not guaranteed: "Disappearing" messages can be captured and shared permanently. Platforms now have legal duties to tackle intimate image abuse under the Online Safety Act.
- Safety requires proactive steps: Staying safe means managing settings, thinking critically, knowing the law, and understanding how to report concerns.
- Abuse can come from anywhere: Harm can come from peers as well as strangers; coercive control and online domestic abuse are now recognised as serious harms.
- Empowerment through knowledge: Young people are safer when they understand risks, legal consequences, and know they can speak to trusted adults without judgment.

sources, and understanding legal safeguards.

- Digital wellbeing is holistic: Safe computer use involves not just protecting data, but also managing physical health (ergonomics) and environmental responsibility (e-waste).

### **Advanced Python**

#### Key Concepts

- Iteration on Strings: for loops can iterate over strings character by character.
- Accumulator Pattern: Using variables to maintain running counts and sums while iterating through sequences.
- Type Conversion: Characters may need conversion (e.g., int()) before arithmetic operations.
- Algorithmic Problem-Solving: Real-world problems can be solved by applying core programming constructs.
- Self-Assessment: Reflecting on one's own confidence and understanding of programming concepts.

#### Key Skills

- String Processing: Iterating over strings; counting character types (e.g., vowels); calculating digit sums.
- Accumulator Techniques: Maintaining running counts and sums using variables.
- Program Design: Building complete mini-projects (password safety, ISBN, Caesar cipher, planetary quiz) that apply multiple constructs.
- Testing & Reflection: Assessing personal confidence levels; taking summative assessments; reflecting on learning.

	<p><b>Key Generalizations</b></p> <ul style="list-style-type: none"> <li>• Loops enable efficient text processing: Iteration structures allow examination of every character without repetitive code.</li> <li>• Accumulator variables track running totals: Counters and sums are essential for analysing sequences.</li> <li>• Real-world problems can be solved programmatically: Complex tasks can be broken down and solved using core programming constructs.</li> <li>• Application reinforces learning: Building mini-projects consolidates understanding by integrating multiple concepts.</li> </ul>	
<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>
<p><b>Revisit/Review</b></p> <p>This unit provides the most advanced and critical revisit of online safety, building on Year 7 Spring 1 and Year 8 Autumn 1/Spring 1. Students move from general safety rules to understanding specific, high-risk issues like grooming and the legal consequences of sharing images. It connects digital literacy with personal wellbeing and introduces specific UK laws (e.g., Protection of Children Act, Online Safety Act), preparing them for legal frameworks in GCSE topics like cyber security.</p>	<p><b>Revisit/Review</b></p> <p>This term connects two strands. <b>Cyber Security</b> builds on network security (Year 8 Spring 2) and legal awareness (Year 9 Autumn 1) to explore specific threats (malware, phishing) and the Computer Misuse Act. <b>Advanced Python</b> builds directly on Year 8 Summer 2, using more complex string manipulation and iterative patterns (accumulator) to solve problems, reinforcing programming as a tool for real-world applications (password checking, encryption).</p>	<p><b>Revisit/Review</b></p> <p>Students apply their data handling skills from Excel (Year 7 Autumn 2 and Year 9 Summer 1) to a more powerful and structured tool: relational databases. They build on the concept of organising data by learning about tables, primary keys, and relationships. Querying databases using criteria, operators, and wildcards directly applies the logical thinking and search strategies developed in Year 8 Autumn 1.</p>

Year 9		
Spring 2	Summer 1	Summer 2
<p><b>Content:</b> Digital Graphics/ Streamlistic Project</p>	<p><b>Content:</b> Spreadsheets</p>	<p><b>Content:</b> 3D Printing and App Development</p>
<p><b>Concepts/Generalisations/Skills</b></p> <p>Key Concepts</p> <ul style="list-style-type: none"> <li>• Raster vs. Vector: Raster graphics use pixels (for photos); vector graphics use mathematical paths (for logos) and scale without quality loss.</li> <li>• Visual Identity: Consistent logos, colours, and typography create recognisable branding.</li> <li>• Image Structure: Pixels store colour values as binary; resolution (DPI) determines quality for web vs. print.</li> <li>• Layers: Transparent sheets that allow independent editing of elements.</li> <li>• File Formats: Different formats serve different purposes (PNG for web, TIFF for print).</li> </ul> <p>Key Skills</p> <ul style="list-style-type: none"> <li>• Software Proficiency (Photoshop): Setting up canvases with specific dimensions/resolution; using Place tool, layers, selection tools (Marquee, Lasso), retouching tools (Healing Brush, Blur),</li> </ul>	<p><b>Concepts/Generalisations/Skills</b></p> <p>Key Concepts</p> <ul style="list-style-type: none"> <li>• Cell-Based Structure: Spreadsheets organise data into a grid of cells for storage and manipulation.</li> <li>• Formulas vs. Functions: Formulas are user-defined calculations; functions (SUM, AVERAGE) are pre-built operations.</li> <li>• Data Visualisation: Charts (bar, pie, line) visually represent data to reveal patterns and comparisons.</li> <li>• Formatting for Clarity: Techniques like merging cells and number formats improve readability and professionalism.</li> <li>• Audience and Purpose: Spreadsheets should be designed with the end-user in mind.</li> </ul> <p>Key Skills</p> <ul style="list-style-type: none"> <li>• Data Management: Adding, deleting, and updating data; using the fill handle for efficiency.</li> <li>• Formatting: Merging cells; applying text alignment and number formats.</li> </ul>	<p><b>Concepts/Generalisations/Skills</b></p> <p><b>3D Printing</b></p> <p>Key Concepts</p> <ul style="list-style-type: none"> <li>• 3D Design Workflow: 3D modelling (Tinkercad) → slicing (Cura) → printing.</li> <li>• Slicing: Cura converts models into G-code (xyz coordinates) for printer instructions.</li> <li>• Additive Manufacturing: Objects are built layer by layer from extruded material.</li> <li>• Material Versatility: Printing uses plastic, metal, wood, ceramic, chocolate, and concrete.</li> <li>• Print Preparation: Models need optimisation for size, orientation, stability, and supports.</li> </ul> <p>Key Skills</p> <ul style="list-style-type: none"> <li>• 3D Modelling (Tinkercad): Mastering Place, View, Move, Rotate, Size, Group, Align; creating personalised objects; importing/editing existing models.</li> <li>• Print Preparation (Cura): Importing models; adjusting size and orientation; ensuring base stability; adding supports; generating G-code for SD card.</li> </ul>

<p>painting tools (Brush, Paint Bucket), and Type tool.</p> <ul style="list-style-type: none"> <li>• Image Manipulation: Scaling objects; applying layer effects and masks; cropping; removing/replacing colours.</li> <li>• Project Management: Following client briefs (dimensions, resolution, formats); researching assets; exporting appropriately; documenting work.</li> </ul> <p><b>Key Generalizations</b></p> <ul style="list-style-type: none"> <li>• Purpose determines format: Choice of raster/vector, resolution, and file format depends on intended use (web vs. print).</li> <li>• Layers enable non-destructive editing: Elements can be modified without altering the original image.</li> <li>• Selection isolates for precision: Tools allow editing specific areas while leaving others unchanged.</li> <li>• Retouching enhances reality: Digital tools can improve or transform images for commercial purposes.</li> <li>• Consistency builds brand recognition: A cohesive visual identity creates memorable branding.</li> <li>• Following a brief is essential: Professional work requires meeting client specifications.</li> </ul>	<ul style="list-style-type: none"> <li>• Formulas and Functions: Creating calculations; using SUM, AVERAGE, MAX, and MIN.</li> <li>• Chart Creation: Selecting appropriate chart types; creating and labelling charts for clarity.</li> <li>• Practical Application: Applying technical skills to complete real-world spreadsheet projects.</li> </ul> <p><b>Key Generalizations</b></p> <ul style="list-style-type: none"> <li>• Spreadsheets model real-world problems: They store, calculate, and analyse data for business scenarios.</li> <li>• Formulas automate calculations: They reduce manual effort and minimise errors.</li> <li>• Visualisation aids understanding: Charts make complex data more accessible.</li> <li>• Formatting enhances usability: Well-formatted spreadsheets are easier to read and use.</li> <li>• Purpose determines design: Choices should be guided by the audience and problem being solved.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluation: Assessing if models are structurally fit for printing.</li> </ul> <p><b>Key Generalizations</b></p> <ul style="list-style-type: none"> <li>• Design must consider manufacturing: Screen-ready models need preparation for successful printing.</li> <li>• Software translates design to instruction: Tinkercad designs; Cura creates machine instructions.</li> <li>• 3D printing has diverse applications: Used in automotive, medicine, fashion, construction, and food.</li> <li>• Iterative learning builds mastery: Tutorials develop foundational skills progressively.</li> <li>• Existing designs can be adapted: Online galleries provide modifiable starting points.</li> </ul> <p><b>App Development</b></p> <p><b>Key Concepts</b></p> <ul style="list-style-type: none"> <li>• Decomposition: Breaking large problems into smaller, manageable steps.</li> <li>• Event-Driven Programming: Code executes in response to user interactions (clicks).</li> <li>• UI Elements: Apps use screens, buttons, labels with customizable properties (id, text, colour).</li> <li>• Variables: Store and update data during program execution (e.g., score).</li> <li>• User Needs: Success criteria help evaluate if a project meets requirements.</li> <li>• Error Identification: Code can contain bugs needing systematic fixes.</li> </ul> <p><b>Key Skills</b></p> <ul style="list-style-type: none"> <li>• Problem Decomposition: Breaking app requirements into component parts.</li> </ul>
--	--	---

- App Lab Proficiency: Creating projects; adding/customising screens, buttons, labels; setting properties; importing images.
- Event Handling: Creating onEvent blocks linked to specific element IDs.
- Variable Management: Creating, updating, and displaying variables.
- Coding Techniques: Using setScreen, setTimeout, random positions, and getText().
- Collaborative Development: Pair programming with driver/navigator roles.
- Debugging: Identifying and fixing common errors (wrong IDs, incorrect event types).
- Project Documentation: Establishing success criteria; designing screens; reflecting on progress.

#### Key Generalizations

- Decomposition makes complex problems manageable: Breaking projects into steps provides a clear path forward.
- Events control program flow: Code runs in response to user actions, not linearly.
- IDs connect UI to code: Every element needs a unique ID for events to reference it.
- Variables track changing data: Essential for values that update during execution.
- Order matters inside events: Instruction sequence within an event determines outcome.
- Errors are fixable: Common mistakes can be systematically identified and corrected.
- Collaboration improves outcomes: Pair programming promotes shared understanding.

		<ul style="list-style-type: none"> <li>• Planning precedes building: Thinking through screens and criteria leads to efficient development.</li> </ul>
<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment (inc. Streamlistic Project)</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>	<p><b>Assessment</b></p> <ul style="list-style-type: none"> <li>• End of Topic Assessment</li> </ul>
<p><b>Revisit/Review</b></p> <p>This project-based unit brings together many prior skills. Students apply their understanding of image representation (pixels, resolution from Year 8 Autumn 2) in a practical design context. They follow a client brief, revisiting the principles of audience and purpose (Year 8 Summer 1). They manage files and assets, building on Year 7 Autumn 1 skills, to create a professional product for a specific scenario.</p>	<p><b>Revisit/Review</b></p> <p>Students return to spreadsheets (first introduced in Year 7 Autumn 2) with a more advanced and professional focus. They consolidate and extend their knowledge of formulae, functions, and charts. The emphasis on practical application and modelling "what-if" scenarios builds directly on the analytical skills developed in the Spy School unit, preparing them for data analysis in GCSE and beyond.</p>	<p><b>Revisit/Review</b></p> <p>This final term applies learning from across KS3 to two distinct projects. <b>3D Printing</b> applies computational thinking (decomposition) to a design workflow, revisiting concepts of planning and evaluation (from Year 8 Summer 1). <b>App Development</b> is the culmination of all programming learning (Years 7, 8, and 9). It uses event-driven programming (Scratch, Microbits), variables, and decomposition in a new environment (App Lab) to create a substantial project, mirroring the development cycle used in website design and digital graphics.</p>