Computing Curriculum Map 2025-2026

Subject Intent
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.

Key Stage 2	Year	HT1	HT2	HT3	HT4	HT5	HT6	Beyond Elton
By the end of KS2, students should: - Design, debug, and improve programs using sequences, loops, and conditions to solve problems. - Understand algorithms, fix errors, and explain how code works logically. - Explore how networks (like the internet) enable communication and collaboration. - Use search tools critically and evaluate online information. - Combine software to create projects, analyse data, and present findings. - Use technology safely, spotting risks and knowing how to report concerns.	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	**Key Stage 5** 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 **Careers** Potential career paths include: software developer, programmer, IT technician, IT support, web developer, data analyst, cybersecurity specialist as well as many tech apprenticeships
	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/Streamlistic Project	Spreadsheets	3D Printing and App Development	
	10	Architecture of the CPU/Programming /Computer Memory/Secondary Storage/Cables	Types and topologies of Networks/ Programming Fundamentals (Continue) - Arrays/lists	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/Pseudocode/Inputs, Processes & Outputs	

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

Year 10		
Overview of the Year Students begin their GCSE journey by establishing foundational knowledge of computer systems, architecture, and data representation. They develop practical programming skills alongside theoretical understanding, preparing them for both components of the OCR specification. The year balances theory (Paper 1 content) with practical programming (Paper 2 skills).		
Autumn 1	**Autumn 2**	**Spring 1**
Content: 1.1 Systems Architecture - Purpose of the CPU - CPU components (ALU, CU, cache, registers) - Fetch-Decode-Execute Cycle - Common CPU characteristics - Embedded systems 2.2 Programming Fundamentals (Start) - Variables, constants, data types - Sequence, selection, iteration - Arithmetic operations - String manipulation - File handling operations	**Content:** 1.3 Computer Networks, Connections and Protocols - Types of networks (LAN/WAN) - Factors affecting network performance - Client-server and peer-to-peer - Hardware (NIC, switch, router, etc.) - The internet as a network of networks - Star and mesh topologies - Wi-Fi and Ethernet - IP addressing and MAC addressing - Standards and protocols (TCP/IP, HTTP, HTTPS, FTP, POP, IMAP, SMTP) - The concept of layers 2.2 Programming Fundamentals (Continue) - Arrays/lists	**Content:** 2.4 Computational Logic - Logic gates (AND, OR, NOT) - Truth tables - Binary logic diagrams - Boolean expressions 2.6 Data Representation (Part 1) - Binary to denary conversions - Denary to binary conversions - Binary addition and shifts - Hexadecimal system - Hexadecimal conversions - Binary representation of characters (ASCII, Unicode)

	- Subprograms (functions/procedures)	
Concepts/Generalisations/Skills - Understanding how all processing begins with the CPU's FDE cycle - Different CPU characteristics affect performance - Programming is built on fundamental constructs - Developing trace tables to follow program flow	**Concepts/Generalisations/Skills** - Networks enable communication and resource sharing - Protocols provide the "rules" for data transmission - Layering helps manage network complexity - Practical configuration of simple networks (simulated) - Programming with data structures	**Concepts/Generalisations/Skills** - Logical decisions in computing are based on Boolean algebra - All data must be represented in binary for computer processing - Different data types require different representation methods - Creating truth tables for given logic scenarios - Converting between number bases
Assessment - End of topic test: Systems Architecture - Programming task: Basic input/output program - FDE Cycle diagram labelling	**Assessment** - End of topic test: Networks - Programming task: Array-based program with subprograms - Network topology comparison task - Protocol research task	**Assessment** - End of topic test: Logic & Data Representation - Practical logic gate worksheet - Number conversion quiz
Revisit/Review - KS3 understanding of hardware/software - Basic programming from Year 9 Python - How computers execute instructions	**Revisit/Review** - KS3 networks understanding - Variables and data types from Autumn 1 - IP addressing revisit from network hardware	**Revisit/Review** - KS3 binary work (Year 7/8) - Basic logic from KS3 problem solving - ALU role from CPU component study

Year 10		
Spring 2	**Summer 1**	**Summer 2**
Content: 1.2 Memory and Storage (Part 1) Primary Storage - RAM and ROM - Virtual memory - Flash memory Data Storage (Images) - Pixels and colour depth - Bitmap image size calculation - Metadata	**Content:** 1.2 Memory and Storage (Part 2) Secondary Storage - Common types: magnetic, optical, solid state - Suitable storage devices for given applications - Capacity, speed, portability, durability, reliability, cost Data Storage (Sound) - Sampling rate and bit depth - Sound file size calculation Compression - Lossy vs lossless compression - Common file formats 1.4 Network Security - Forms of attack (malware, phishing, brute force, DoS) - Prevention methods (firewalls, encryption, access levels) - Penetration testing - Anti-malware software	**Content:** 2.1 Algorithms Computational Thinking - Abstraction - Decomposition - Algorithmic thinking Designing Algorithms - Structure diagrams - Flowcharts - Pseudocode Searching Algorithms - Linear search - Binary search 2.3 Robust Programs - Defensive design (validation, authentication) - Testing methodologies - Syntax vs logic errors - Selecting test data (normal, boundary, invalid)
Concepts/Generalisations/Skills - Memory types serve different purposes in a computer system - Images must be digitally encoded for storage and processing - Calculating image file sizes from specifications	**Concepts/Generalisations/Skills** - Storage requirements vary by use case - Sound digitisation involves sampling analogue waves - Compression balances quality against file size - Security threats require layered defences - Comparing storage media for specific scenarios - Identifying vulnerabilities and appropriate protections	**Concepts/Generalisations/Skills** - Computational thinking breaks down complex problems - Algorithms can be represented in multiple ways - Different search algorithms have different efficiencies - Robust programs anticipate and handle errors - Systematic testing ensures program reliability - Creating structure diagrams from problems - Writing and tracing algorithms

Assessment - End of topic test: Primary Memory & Images - RAM/ROM comparison table - Image size calculation practice	**Assessment** - End of topic test: Secondary Storage, Sound & Security - Storage device recommendation task - Network security poster	**Assessment** - End of Year Exam (full Paper 1 and Paper 2 content) - Programming project: Create a menu-driven program with validation - Algorithm tracing exercise - Searching algorithm comparison
Revisit/Review - CPU components and their interaction with memory - Binary from Spring 1 - How programs execute from Autumn 1	**Revisit/Review** - Binary/hex from Spring 1 - Units of data storage (bits, bytes, etc.) - Network threats (linked to Autumn 2)	**Revisit/Review** - All Year 10 theory content - Programming fundamentals from across the year - Logic and truth tables from Spring 1

Year 11

Overview of the Year

Students deepen their understanding of computational thinking, complete their study of programming and algorithms, and refine their knowledge of all specification content. The year focuses on consolidating Paper 2 concepts, mastering programming techniques, and comprehensive revision for both GCSE papers.

Autumn 1	Autumn 2	Spring 1
Content: 2.1 Algorithms (Part 2) Sorting Algorithms - Bubble sort - Merge sort - Insertion sort 2.3 Robust Programs (Continue) - Authentication - Input validation - Maintainability (comments, indentation) - Testing strategies	**Content:** 2.5 Programming Languages and IDEs Programming Languages - Characteristics of high/low level languages - Translators (compiler, interpreter, assembler) Integrated Development Environments (IDEs) - Common IDE features (editors, error diagnostics, run-time environment, translators) 2.2 Programming Fundamentals (Advanced) - Records and SQL - Arrays of records - Random number generation - Structured programming approach Programming Project - Extended programming task combining multiple skills	**Content:** 1.5 Systems Software Operating Systems - User interface - Memory management/multitasking - Peripheral management - User management - File management Utility Software - Encryption software - Defragmentation - Data compression - Backup and system restore 1.6 Ethical, Legal, Cultural and Environmental Impact - Impacts of technology on society - Legislation (Data Protection Act, Computer Misuse Act, Copyright Act) - Environmental impact - Open source vs proprietary software

Concepts/Generalisations/Skills - Sorting organises data for efficient processing - Different sorting algorithms have different efficiencies - Robust programs require careful design and testing - Comparing algorithm efficiency through trace tables - Implementing search and sort algorithms in code	**Concepts/Generalisations/Skills** - Languages are chosen based on problem requirements - IDEs support the development process - Translators convert code into executable form - Structured programming improves code quality - Developing a complete program from specification	**Concepts/Generalisations/Skills** - The OS manages all system resources - Utility software maintains system performance - Technology has far-reaching impacts beyond functionality - Legal frameworks govern technology use - Evaluating ethical and environmental implications - Applying legislation to scenarios
Assessment - End of topic test: Sorting Algorithms - Algorithm implementation task - Test plan creation and execution - Code review exercise	**Assessment** - Mock Paper 2 (Computational Thinking) - End of topic test: Languages & IDEs - Extended programming project (2 weeks) - IDE features exploration task - Language translator comparison	**Assessment** - End of topic test: Systems Software & Legal/Ethical - OS function diagram - Legislation case study analysis - Debate: Ethical implications of AI
Revisit/Review - Searching algorithms from Summer 2 Year 10 - Validation techniques from Summer 2 Year 10 - Trace tables from Autumn 1 Year 10	**Revisit/Review** - All programming concepts from Year 10 - Data types and structures - File handling from Year 10 Autumn 1	**Revisit/Review** - CPU interaction with OS from Year 10 Autumn - Memory management links to virtual memory - Data representation links to compression

Year 11		
Spring 2	**Summer 1**	**Summer 2**
Content: Revision: Paper 1 Computer Systems 1.1 Systems Architecture 1.2 Memory and Storage 1.3 Computer Networks 1.4 Network Security 1.5 Systems Software 1.6 Ethical, Legal, Cultural and Environmental Impact Exam Technique - Command words - Mark scheme interpretation - Time management - Extended response questions	**Content:** Revision: Paper 2 Computational Thinking 2.1 Algorithms - Searching and sorting - Flowcharts and pseudocode - Computational thinking methods 2.2 Programming Fundamentals - All programming constructs - Data structures - File handling 2.3 Robust Programs - Defensive design - Testing 2.4 Computational Logic - Logic gates and truth tables 2.5 Programming Languages and IDEs Exam Technique - Past paper practice - Walk-throughs of challenging topics	**Content:** External Examinations - J277/01: Computer Systems - J277/02: Computational Thinking
Concepts/Generalisations/Skills - Making connections between Paper 1 topics - Applying knowledge to examination contexts - Developing examination techniques - Identifying areas for targeted revision - Creating revision resources	**Concepts/Generalisations/Skills** - Applying computational thinking to unfamiliar problems - Tracing and debugging algorithms - Writing clear pseudocode - Explaining code and algorithms - Full past paper practice under timed conditions	**Concepts/Generalisations/Skills** - Application of knowledge under timed conditions - Exam technique in practice
Assessment - Paper 1 Mock Exam (1 hour 30 minutes) - Targeted topic tests based on mock performance (focus on weak areas) - Extended writing practice (6-mark questions with mark schemes)	**Assessment** - Paper 2 Mock Exam - Final past paper practice - Peer marking and reflection	**Assessment** - OCR GCSE Computer Science Paper 1 - OCR GCSE Computer Science Paper 2

- Command word quiz		
Revisit/Review - Full Year 10 content - Year 11 Autumn/Spring content - Areas of weakness from mocks	**Revisit/Review** - Full Year 10 and 11 content - Exam technique refinement - Final knowledge checks	**Revisit/Review**