Subject: Computer Science KS4 Core Knowledge

Topic	Learners should know and understand:
Year 10	
1.1 Systems architecture	• Understand the purpose of the CPU and the fetch-execute cycle • Explain the function of the ALU, CU, cache and registers (MAR, MDR, program counter, accumulator) within the Von Neumann architecture • Explain how clock speed, cache size and number of cores affect CPU performance • Understand the purpose and characteristics of embedded systems, with examples
1.2 Memory and storage (primary and secondary storage)	• Understand the need for primary storage and the differences between RAM and ROM • Understand why virtual memory is needed and how it works • Understand the need for secondary storage and the common types: optical, magnetic and solid state • Compare storage devices using capacity, speed, portability, durability, reliability and cost • Know the units of data storage from bit to petabyte, and calculate storage requirements
1.2 Memory and storage (data storage)	• Convert between binary, denary and hexadecimal (0–255 / 00–FF) • Add two binary integers (up to 8 bits) and explain overflow • Apply binary shifts and describe their effect • Understand character sets and the differences between ASCII and Unicode • Explain how colour depth and resolution affect image quality and file size • Explain how sample rate, duration and bit depth affect sound quality and file size • Calculate image and sound file sizes • Understand the need for compression and the difference between lossy and lossless compression
2.4 Boolean logic	• Know the logic gate symbols and truth tables for AND, OR and NOT • Create and complete logic diagrams and truth tables, including combinations of gates
1.3 Networks, connections and protocols	• Understand LANs and WANs and the factors that affect network performance • Understand the roles of computers in client-server and peer-to-peer networks • Know the purpose of network hardware: wireless access point, router, switch, NIC and transmission media • Understand the internet, including DNS, hosting and the Cloud • Compare star and mesh topologies, and wired and wireless connections • Understand IP and MAC addressing, standards, common protocols (TCP/IP, HTTP, HTTPS, FTP, POP, IMAP, SMTP) and the concept of layers
2.1 Algorithms	• Understand abstraction, decomposition and algorithmic thinking • Create, interpret, correct and refine algorithms using pseudocode, flowcharts and code • Use trace tables to follow an algorithm • Understand and apply linear search and binary search • Understand and apply bubble sort, merge sort and insertion sort
2.2 Programming fundamentals (Part 1)	• Use variables, constants, inputs, outputs and assignments • Use sequence, selection and iteration (count-controlled and condition-controlled) • Use arithmetic (including MOD, DIV and exponentiation), comparison and Boolean operators • Use the data types integer, real, Boolean, character and string, and casting • Use string manipulation (concatenation and slicing) and 1D and 2D arrays • Use SQL (SELECT, FROM, WHERE) to search for data

Subject: Computer Science KS4 Core Knowledge

Topic	Learners should know and understand:
Year 11	
1.4 Network security	• Understand the forms of attack: malware, social engineering, brute-force attacks, denial of service, data interception and theft, and SQL injection • Understand how penetration testing, anti-malware, firewalls, user access levels, passwords, encryption and physical security limit threats
1.5 Systems software	• Understand the functions of an operating system: user interface, memory management and multitasking, peripheral management and drivers, user management and file management • Understand the purpose of utility software: encryption, defragmentation and data compression
1.6 Ethical, legal, cultural and environmental impacts	• Discuss the ethical, legal, cultural, environmental and privacy impacts of digital technology on society • Know the purpose of the Data Protection Act 2018, Computer Misuse Act 1990 and Copyright Designs and Patents Act 1988 • Understand the difference between open source and proprietary software licences
2.2 Programming fundamentals (Part 2)	• Use file handling operations: open, read, write and close • Use functions and procedures with parameters and return values • Understand the difference between local and global variables • Use random number generation
2.3 Producing robust programs	• Understand defensive design: anticipating misuse, authentication and input validation • Understand how to make programs maintainable: subprograms, naming conventions, indentation and commenting • Understand iterative and final testing, and the difference between syntax and logic errors • Select suitable test data: normal, boundary, invalid and erroneous
2.5 Programming languages and IDEs	• Understand the differences between high-level and low-level languages • Understand the purpose of translators and the differences between compilers and interpreters • Know the common tools in an IDE: editors, error diagnostics, run-time environment and translators