Curriculum Map – Computer Science – Year 10

Immanuel College
Church of England Academy

	Half term 1	Half term 2	Half term 3	Half term 4	Half term 5	Half term 6
Key focus	1.1 Systems Architecture Practical Programming 2.1 Logic & Languages	1.2 Memory & Storage Practical Programming	1.5 Systems Software Practical Programming 2.1 Logic & Languages	1.5 Systems Software	1.3 Network connections & Protocol 1.4 Network Security Practical Programming	1.6 Impact of digital technology Programming fundamentals

Intent	**(2.1)**: Students will explore the fundamental theories of computer science, including data representation, algorithms, and the importance of systems thinking. They will learn to appreciate the role of computer science in solving real-world problems, fostering critical and logical thinking skills.
	(1.1): This section will introduce students to the hardware and software components of computers, including how they interact to perform tasks. Students will gain hands-on experience with various devices, understanding their functionality and the basic principles of how machines operate.
	(Programming): Students will develop practical programming skills through engaging projects. They will learn different programming paradigms, problem-solving strategies, and debugging techniques. By applying their knowledge to create programs, students will enhance their creativity and logical reasoning, preparing them for future
		(1.2): To understand the fundamental differences between memory and storage in computing, including their functions, capacities, types, and how they impact overall system performance.	**(1.5):** By the end of this section, students will be able to identify and describe key hardware components, differentiate between types of software, and understand the relationship between hardware and software in a computer system. This knowledge will form the basis for further exploration of more advanced topics in computer science.	**(1.5):** By the end of this section, students will be able to identify and describe key hardware components, differentiate between types of software, and understand the relationship between hardware and software in a computer system. This knowledge will form the basis for further exploration of more advanced topics in computer science.	**(1.3):**
					To equip students with a solid understanding of how computers communicate over networks. This includes the fundamental concepts of networking, the role of various types of networks (e.g., LAN, WAN), and the importance of protocols in ensuring efficient and secure data transmission.
					(1.4):
					To provide students with a deep understanding of the potential threats and vulnerabilities within networked systems, and the essential methods used to protect data and ensure secure communication. By learning about network security, students will be prepared to recognize the importance of safeguarding digital information in today's interconnected world and will develop the skills necessary to mitigate common security risks.	

"Perseverance produces **character,** and character, **hope**" (Romans 5:4)

	studies or careers in technology.					
Key knowledge and skills	**2.1:** • Understanding of binary systems and data representation (bits, bytes, etc.) • Awareness of algorithms and their characteristics (sequence, selection, iteration) • Familiarity with basic data structures (arrays, lists, etc.) • Concepts of computational thinking (decomposition, pattern recognition, abstraction) • Ethical and societal implications of technology • Analysing problems to identify appropriate computational solutions • Creating flowcharts and pseudocode to represent algorithms • Evaluating the efficiency of algorithms (time and space complexity) • Applying computational thinking to real-world scenarios **1.1:** • Overview of computer hardware components (CPU, memory, storage, input/output devices) • Understanding of operating systems and their functions	**1.2:** • Definitions of memory (RAM) and storage (e.g., SSD, HDD). • Differences between volatile and non-volatile memory. • The purpose and roles of memory and storage in a computer system. • Familiarity with different types of memory, such as RAM, ROM, and cache. • Knowledge of various storage options, including HDD, SSD, and cloud storage. • Pros and cons of SSDs versus HDDs in terms of speed, durability, and cost. • How memory supports CPU operations by holding active processes and data. • How storage enables long-term data retention and access. • Concepts of primary vs. secondary storage. • Understanding how memory size and speed affect system performance and multitasking. • Impact of storage type on boot times, data retrieval speeds, and application loading. • Ability to evaluate different memory and storage options for specific needs, such as high performance vs. high capacity. • Skill in assessing memory and storage requirements for various applications	**1.5:** • **Components**: Identify and describe the main components of a computer system (CPU, RAM, storage devices, motherboards, power supply). • **Functions**: Understand the role and function of each hardware component in the overall system. • **Types of Devices**: Differentiate between input devices (e.g., keyboard, mouse) and output devices (e.g., monitor, printer). • **Storage**: Explain the differences between volatile and non-volatile memory (RAM vs. hard drives/SSD). • **Types of Software**: Understand the distinction between system software (operating systems, utility programs) and application software (word processors, games). • **Operating Systems**: Explain the purpose of operating systems and their functions (managing hardware, providing a user interface). • **Installation and Maintenance**: Knowledge of how to install, update, and uninstall software, including troubleshooting common issues. • **Drivers**: Understand what drivers are and their role in enabling communication between hardware and software.	**1.5:** • **Components**: Identify and describe the main components of a computer system (CPU, RAM, storage devices, motherboards, power supply). • **Functions**: Understand the role and function of each hardware component in the overall system. • **Types of Devices**: Differentiate between input devices (e.g., keyboard, mouse) and output devices (e.g., monitor, printer). • **Storage**: Explain the differences between volatile and non-volatile memory (RAM vs. hard drives/SSD). • **Types of Software**: Understand the distinction between system software (operating systems, utility programs) and application software (word processors, games). • **Operating Systems**: Explain the purpose of operating systems and their functions (managing hardware, providing a user interface). • **Installation and Maintenance**: Knowledge of how to install, update, and uninstall software, including troubleshooting common issues. • **Drivers**: Understand what drivers are and their role in enabling communication between hardware and software.	**1.3:** • Understanding and Explaining Network Protocols: • Identify and explain the functions of various network protocols (TCP/IP, HTTP, FTP, SMTP, DNS, etc.). • Understand how protocols ensure that data is sent and received reliably, securely, and efficiently. • Understand how to configure devices to connect to wired or wireless networks (e.g., setting up Wi-Fi or Ethernet connections). • Know how to troubleshoot network connectivity issues (e.g., diagnosing IP address conflicts, weak Wi-Fi signals). • Understanding and applying the TCP/IP Models: • Be able to describe each layer of the TCP/IP model and their functions. • Understand how devices are assigned IP addresses in a network and the concept of static vs. dynamic IP addressing (e.g., DHCP). • Understand the role of DNS in resolving domain names to IP addresses. • Explain the process of data packetization, error checking, and reassembly in network communication. • Understand how HTTPS, SSL/TLS, and encryption protocols help protect data during transmission.	

	• Basics of how software applications interact with hardware • Introduction to networking concepts and the Internet • Identifying and explaining the functions of different hardware components • Troubleshooting basic hardware and software issues • Understanding and applying basic networking principles (IP addresses, protocols) • Demonstrating safe and responsible use of technology **Programming:** • Familiarity with programming languages (Python) • Understanding of syntax and semantics in programming • Concepts of variables, data types, and control structures • Basic principles of object-oriented programming (if applicable) • Writing, testing, and debugging code to create functional programs • Using libraries and APIs to enhance programming capabilities • Developing problem-solving strategies through coding challenges	(e.g., gaming, video editing, general office work). • Recognizing situations where more RAM or a faster storage drive (like an SSD) would improve performance. • Identifying performance bottlenecks related to insufficient memory or slow storage.	• **User Interface**: Recognize different types of user interfaces (GUI vs. command line) and their impact on user experience. • **Resource Management**: Explain how operating systems manage hardware resources (memory management, process scheduling). • Ability to explain technical concepts clearly and effectively, both verbally and in writing. • Skill in documenting procedures for software installation and hardware setup.	• **User Interface**: Recognise different types of user interfaces (GUI vs. command line) and their impact on user experience. • **Resource Management**: Explain how operating systems manage hardware resources (memory management, process scheduling). • Ability to explain technical concepts clearly and effectively, both verbally and in writing. • Skill in documenting procedures for software installation and hardware setup. **2.1:** • Understanding of binary systems and data representation (bits, bytes, etc.) • Awareness of algorithms and their characteristics (sequence, selection, iteration) • Familiarity with basic data structures (arrays, lists, etc.) • Concepts of computational thinking (decomposition, pattern recognition, abstraction) • Ethical and societal implications of technology • Analysing problems to identify appropriate computational solutions • Creating flowcharts and pseudocode to represent algorithms • Evaluating the efficiency of algorithms (time and space complexity)	• Explain how security protocols prevent data interception, eavesdropping, and tampering. • Understand the significance of network performance metrics, such as latency, bandwidth, and packet loss. • Understand how network ports are used to identify specific services on a device (e.g., port 80 for HTTP, port 443 for HTTPS). • Know how ports are managed and how firewalls can block or allow specific port traffic. **1.4:** • Identify various **network security threats** and understand their potential impacts. • Set up **authentication mechanisms** and ensure proper access control to protect user data. • Secure **wireless networks** and understand protocols like HTTPS, SSL/TLS, and VPNs for protecting data in transit. • Respond effectively to **security incidents** and ensure that data privacy is maintained according to relevant regulations (e.g., GDPR). • Understand the role of **firewalls, encryption**, and other technologies in creating a secure network environment.	

				- Applying computational thinking to real-world		
Key words/ vocabulary	**2.1:** Binary, Data Representation, Algorithm, Computational Thinking, Decomposition, Abstraction, Pattern Recognition, Data Structure, Efficiency, Ethics **1.1:** Hardware, Software, CPU (Central Processing Unit), Memory, Input/Output Devices, Operating System, Networking, Protocol, Storage, Troubleshooting **Programming:** Programming Language, Syntax, Variable, Data Type, Control Structure, Object-Oriented Programming, Function, Debugging, API (Application Programming Interface), Version Control	**1.2**: Memory, Storage, RAM (Random Access Memory), Cache Memory, Volatile Memory, Non-Volatile Memory, Primary Storage, Secondary Storage, HDD (Hard Disk Drive), SSD (Solid State Drive), Read/Write Speed, Data Retrieval, Access Time, Latency, Capacity, Performance, CPU (Central Processing Unit), Temporary Storage, Permanent Storage, Data Retention, System Responsiveness, Multitasking, Memory Hierarchy, Virtual Memory, Flash Memory, ROM (Read-Only Memory), Bandwidth, Data Bus, Cloud Storage, Data Transfer Rate	**1.5:** Operating System (OS), Kernel, Device Drivers, Utility Software, File Management, Memory Management, Multitasking, Task Scheduler, Security Software, Backup Software, Command Line, Interface (CLI), Graphical User Interface (GUI), System Software, Resource Allocation, Booting, Firmware, Virtual Memory, System Calls, User Interface (UI), File System, Process Management, System Configuration, Resource Sharing, Networking Utilities, System Updates	**2.1:** Binary, Data Representation, Algorithm, Computational Thinking, Decomposition, Abstraction, Pattern Recognition, Data Structure, Efficiency, Ethics **1.5:** Operating System (OS), Kernel, Device Drivers, Utility Software, File Management, Memory Management, Multitasking, Task Scheduler, Security Software, Backup Software, Command Line, Interface (CLI), Graphical User Interface (GUI), System Software, Resource Allocation, Booting, Firmware, Virtual Memory, System Calls, User Interface (UI), File System, Process Management, System Configuration, Resource Sharing, Networking Utilities, System Updates	**1.3**: LAN (Local Area Network), WAN (Wide Area Network), VPN (Virtual Private Network), Ethernet, Wi-Fi (Wireless Fidelity), Topology, Bandwidth, Latency, Firewall, Router, Switch, Modem, Access Point, Network Interface Card (NIC), TCP (Transmission Control Protocol), IP (Internet Protocol), TCP/IP, HTTP (Hypertext Transfer Protocol), HTTPS (Hypertext Transfer Protocol Secure), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3), IMAP (Internet Message Access Protocol), DNS (Domain Name System), DHCP (Dynamic Host Configuration Protocol), VoIP (Voice over Internet Protocol), Layer/Transport Layer, IPv4/IPv6	Data, Information, Cybersecurity, Cybercriminals, Profiling, User behaviour, Privacy policies, Data protection, Data subject, Data portability, Malware, Social engineering, Phishing, Blagging, Shouldering, Name generator attacks, Scam, Cyberthreats, Hacking, Ethical hacking, Penetration testing, Brute force attacks, Script kiddies, DoS (denial of service), DDoS (distributed denial of service), Computer Misuse Act (1990), Ransomware, Malware, Viruses, Trojans, Worms, Adware, Spyware, Bots, Botnet, Anti-malware, Firewall, End-user authentication, Folder permissions/privileges, Botnet, Trojans, Biometrics, Two-factor authentication (2FA), CAPTCHA, Internet Service Provider (ISP), Auto-updates
Assessment method	Summative test 1.1, 2.1 Programming Challenge	Summative test 1.2, 2.1 Programming Challenge	Summative test 1.5, 2.1 Programming Challenge	Summative test 1.5, 2.1 Programming Challenge	Summative test 1.3, 1.4 Programming Challenge	Summative test 1.6 Programming Challenge
Wider links	Maths, Physics, Engineering	Maths, Physics, Engineering, D&T	Maths, Physics, Engineering	Maths, Physics, Engineering	Maths, Physics, Engineering	Sociology, Psychology, Law
Enrichment opportunities	Bebras Challenge	Issac Computing Workshop	University Open Day		Look at academy infrastructure	Debating competition
Careers links	Software engineer, programmer	Systems administrator, Network manager, engineer	Systems administrator, Network manager, engineer	Systems administrator, Network manager, engineer	Network Manager, Cloud based engineer	Ethics, Lawyer, Data manager