Curriculum Plans: Year 10 (Computer Science)

	Topic	Knowledge: By the end of the unit students will know:	Skills: What skills will students have developed by the end of this unit?	Key terms: What new key terms and vocabulary will be learnt in this unit?	Summative Assessment: How will pupils be assessed in this unit?
Term 1	Systems Architecture	• Understanding the purpose of the CPU and the fetch-execute cycle. • Familiarity with the components of the CPU (ALU, CU, cache, registers). • Understanding the Von Neumann architecture (MAR, MDR, program counter, accumulator). • The impact of common CPU characteristics (clock speed, cache size, number of cores) on performance. • Recognizing the purpose and characteristics of embedded systems. • The function and role of various registers in the CPU.	• Describing and explaining the stages of the fetch-execute cycle. • Identifying and labeling CPU components and explaining their functions. • Explaining how the Von Neumann architecture stores and processes data. • Can code using Assembly language • Analyzing how clock speed, cache size, and the number of cores affect CPU performance. • Identifying examples of embedded systems and describing how they process data and produce output. • Differentiating between the roles of various registers like the program counter and accumulator.	• CPU (Central Processing Unit) • ALU (Arithmetic Logic Unit) • CU (Control Unit) • Cache • MAR (Memory Address Register) • MDR (Memory Data Register) • Program Counter • Embedded Systems	**Written Assessment –** End of topic test

			• Applying knowledge to scenarios involving embedded systems and their components.		
	Memory and Storage – part 1	• The need for primary storage in computer systems. • The difference between RAM and ROM, including their characteristics (volatile/non-volatile). • Understanding the purpose and function of virtual memory. • The need for secondary storage and the types of devices used (optical, magnetic, solid-state). • Common types of secondary storage and how they store data (magnetism, semiconductor chips, etc.). • Understanding the advantages and disadvantages of different storage media and devices. • The purpose of tertiary storage for backups and data archiving.	• Differentiating between RAM and ROM and explaining their purposes. • Illustrating how virtual memory functions and why it is needed. • Explaining scenarios where virtual memory is used and how it affects system performance. • Comparing storage devices in terms of capacity, speed, portability, durability, reliability, and cost. • Evaluating storage devices for specific applications (e.g., solid-state drives for portability). • Creating flowcharts or Top Trump cards to compare storage devices based on their characteristics. • Making recommendations for	• RAM (Random Access Memory) • ROM (Read-Only Memory) • Virtual Memory • Secondary Storage • Optical Storage • Magnetic Storage • Solid-State Storage • Tertiary Storage	

			the most suitable storage devices based on user needs. • Explaining how secondary storage complements primary storage in computer systems.		
	Memory and Storage – part 2	• Understanding units of data storage (bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte). • The necessity of converting data into binary format for computer processing. • Calculating data capacity requirements for storage. • Binary addition and understanding overflow errors. • Converting between binary and hexadecimal, and denary and hexadecimal. • How character sets (ASCII, UNICODE) are represented in binary. • Understanding how images are represented in binary and the effect of color depth and resolution. • The importance of compression (lossy and lossless) for reducing file size.	• Converting between different units of data storage (e.g., from bytes to kilobytes). • Performing binary to denary, hexadecimal, and vice versa conversions. • Adding binary integers and understanding overflow errors. • Applying binary shifts to manipulate data in binary format. • Calculating file sizes for images and audio based on color depth, resolution, and sample rate. • Explaining how sound is sampled and stored digitally, considering bit depth and sample rate. • Using binary shifts to manipulate numbers	• Bit • Byte • Nibble • Binary • Denary • Hexadecimal • Overflow Error • Character Set • ASCII • UNICODE • Pixel, Metadata • Compression • Lossy • Lossless	**Written Assessment –** End of topic test with MCQ on MS Forms

Curriculum Plans: Year 10 (Computer Science)

			and understanding the impact on data representation. • Comparing lossy and lossless compression techniques and their impact on quality and file size.		
	Computer Networks	• Understanding different types of networks: LAN (Local Area Network) and WAN (Wide Area Network). • Factors that affect network performance, such as bandwidth, error rate, and latency. • The role of hardware in connecting computers to a network (e.g., routers, switches, NICs). • Client-server vs. peer-to-peer network models. • Different network topologies: star and mesh. • The internet as a global network and concepts like DNS, hosting, and cloud computing. • Understanding wired (Ethernet) and wireless (Wi-Fi, Bluetooth) modes of connection.	• Describing and illustrating the differences between LAN and WAN. • Analyzing how network performance is impacted by different factors (e.g., number of users, bandwidth). • Setting up and configuring network hardware like routers and wireless access points. • Differentiating between client-server and peer-to-peer models and explaining their use cases. • Comparing the advantages and disadvantages of star and mesh topologies. • Explaining how the internet works,	• LAN (Local Area Network) • WAN (Wide Area Network) • Bandwidth • Latency • Router • Switch • IP Address • MAC Address • TCP/IP • HTTP/HTTPS • Protocols	**Written Assessment –** End of topic test with MCQ on MS Forms

		• Encryption, IP addressing (IPv4, IPv6), and MAC addressing. • Network protocols such as TCP/IP, HTTP, HTTPS, FTP, POP, IMAP, and SMTP. • - The concept of layering in networking and its advantages.	including the role of DNS, IP addresses, and hosting. • Recommending appropriate network connection modes for different scenarios (wired, wireless, or hybrid). • Configuring network encryption and using IP/MAC addresses for device identification. • Applying and explaining common network protocols and the concept of layers in networking. • - Illustrating how layering simplifies network design and maintenance.		
	Network Security	• Understanding different forms of attack: malware, social engineering, brute-force attacks, denial-of-service, data interception, and SQL injection. • The risks posed by social engineering, including phishing and people as a weak point.	• Identifying different forms of cyberattacks and understanding how each one works. • Explaining how social engineering exploits human behavior to compromise systems. • Identifying and preventing brute-force attacks and	• Malware • Phishing • Brute-force Attack • Denial-of-service (DoS) • SQL Injection • Encryption • Firewall	

		• The purpose of brute-force attacks and how to recognize and prevent them. • How denial-of-service (DoS) attacks work and how to protect against them. • The role of SQL injection in data theft and how to protect databases from it. • Common network security measures such as penetration testing, anti-malware, firewalls, encryption, and physical security. • The importance of user access levels, passwords, and two-factor authentication for system security. • - Different types of security software and hardware (e.g., firewalls, anti-malware, physical security).	implementing password policies. • Applying techniques like firewalls and traffic filtering to prevent DoS attacks. • Implementing validation and security checks to prevent SQL injection. • Explaining and applying various security measures to protect systems from attacks. • Implementing user access levels and strong password policies to safeguard sensitive data. • - Conducting penetration testing to identify vulnerabilities and improve system security.	• Penetration Testing	
	Systems Software	• Understanding the purpose and functionality of operating systems, including user interfaces, memory management, peripheral management, and file management. • Different types of user interfaces: graphical user	• Describing how operating systems manage hardware and software resources efficiently. • Comparing and explaining the features and uses of different types of user interfaces.	• Operating System • User Interface • GUI • CLI • Multitasking • Drivers	**Written Assessment -** Written Exam Assessed as part of the end of year exam

		interface (GUI), command-line interface (CLI), natural language, and menu interfaces. • How operating systems handle multitasking and memory management. • The role of drivers in peripheral management and how the OS communicates with different hardware devices. • The importance of user management in operating systems, including login systems and permissions. • File management and how the OS organizes, moves, and edits files. • The purpose and functionality of utility software, including encryption, defragmentation, and data compression. • How encryption, defragmentation, and data compression work and their importance in maintaining system performance.	• Demonstrating how multitasking works and explaining how the OS manages memory and processes. • Explaining how operating systems use drivers to manage communication between software and peripherals. • Managing user access and permissions within an operating system to ensure security and efficiency. • Using file management features of an OS to organize, edit, and manage files and folders. • Using and explaining utility software to maintain and optimize system performance. • Applying encryption, defragmentation, and data compression techniques to improve system security and efficiency.	• User Management • File Management • Utility Software • Encryption • Defragmentation • Data Compression	