COMPUTER SCIENCE Year 10 Curriculum End Points and Key Vocabulary

	Autumn Term 1	Autumn Term 2	Spring Term 1	Spring Term 2	Summer Term 1	Summer Term 2
Ethos Links	**STEM** - problem solving, programming, making links & applying knowledge, converting binary numbers, adding binary numbers, converting between different units, representing negative numbers in binary **Character** - risk-taking, resilience, perseverance & learning from mistakes, critical thinking, reasoning & making judgements	**STEM** - problem solving, programming, making links & applying knowledge, converting numbers to and from hexadecimal, applying logical shifts, understanding how characters are represented **Character** - risk-taking, resilience, perseverance & learning from mistakes, critical thinking, reasoning & making judgements	**STEM** - problem solving, programming, making links & applying knowledge, how CPU works, types of secondary storage devices **Character** - risk-taking, resilience, perseverance & learning from mistakes, critical thinking, reasoning & making judgements, analysis and evaluation, questioning and thinking, researching	**STEM** - problem solving, programming, making links & applying knowledge, how operating systems of a computer system work, different types of utility software and the purpose of them **Character** - risk-taking, resilience, perseverance & learning from mistakes, critical thinking, reasoning & making judgements, researching, analysis and evaluation, questioning and thinking	**STEM** - problem solving, programming, making links & applying knowledge, network security, cybersecurity, how to prevent cyber-attacks, how to protect data **Character** - risk-taking, resilience, perseverance & learning from mistakes, critical thinking, reasoning & making judgements, researching, analysis and evaluation, questioning and thinking	**STEM** - problem solving, programming, making links & applying knowledge, creating, imagining, innovating, types of networks, network performance **Character** - risk-taking, resilience, perseverance & learning from mistakes, critical thinking, reasoning & making judgements, analysis and evaluation, questioning and thinking, researching
Learning End Points	**CT: Problem Solving with Programming AT1** By the end of this unit students will know and understand: • The terms 'program', 'Decomposition', 'Algorithm', 'sequence', 'variable', 'runtime error' • Types of programs used every day • Python as a programming language **Principles: Data 1** By the end of this unit students will know and understand: • The terms 'digital computer', 'nibble', 'byte', 'overflow error' • Different types of computer • The effects of an overflow error	**CT: Problem Solving with Programming AT2** By the end of this unit students will know and understand: • The terms 'AND', 'NOT' and 'OR' **Principles: Data 2** By the end of this unit students will know and understand: • The terms 'hexadecimal', 'character set' • The range of values that can be represented in two's complement by a binary number of a given length • Why a number may be less precise after a binary shift right has been applied	**CT: Problem Solving with Programming SpT1** By the end of this unit students will know and understand: • The terms 'array', 'list', 'procedure', 'parameter', 'function' and 'return value' • The range() function generates a sequence of numbers **Principles: Hardware 1** By the end of this unit students will know and understand: • What is meant by the 'stored program concept' • The hardware components used in the von Neumann architecture and explain their role in the fetch-decode-execute cycle	**CT: Problem Solving with Programming SpT2** By the end of this unit students will know and understand: • The terms 'array', 'list' in relation to two-dimensional • How to apply a linear search to a one-dimensional list (paper) • How to apply a linear search to a two-dimensional list (paper) • How to complete a linear search algorithm in a flowchart **Principles: Software** By the end of this unit students will know and understand: • The role of the operating system in a computer system	**CT: Problem Solving with Programming SuT1** By the end of this unit students will know and understand: • The merge sort algorithm • The terms 'authentication' **Principles: Network Security, Cybersecurity & Programming Languages** By the end of this unit students will know and understand: • What is meant by the terms 'cyberattack', 'hacker', 'social engineering' and 'robust software' • The financial, reputational and legal damage that a cyberattack can cause	**CT: Problem Solving with Programming SuT2** By the end of this unit students will know and understand: • How to use the turtle module, programming constructs, and subprograms to create images. **Principles: Networks 1** By the end of this unit students will know and understand: • Why computers are connected on a network • The benefits to organisations of a WAN • Why protocols are needed on a network • The purpose of an IP address • The meanings of the terms 'bandwidth', 'latency' and 'topology'

	• The difference between signed and unsigned integers • How positive and negative numbers are represented in two's complement	• How an arithmetic right shift differs from a logical right shift • Why hexadecimal notation is used • How characters are represented in 7-bit ASCII • The shortcomings of ASCII and how encoding systems that use more bits overcome them	• How the speed of the clock impacts on performance • How pipelining improves the performance of the CPU • The relationship between the width of the address bus and the number of memory locations that can be addressed • Why secondary storage is needed • How data are stored on magnetic, optical and solid-state media	• How the OS organises files and allocates space on a hard drive • How file permissions are used to control access to files • How an OS uses scheduling to give each active process a share of CPU time • The features of the round-robin scheduling algorithm • How the OS uses a paging algorithm to swap programs in and out of main memory. • What is meant by the term 'peripheral' • How the OS uses drivers to communicate with and manage peripherals • The purpose of a user interface and describe features of a user interface • What is meant by the term 'access control' and 'utility software' • Commonly used methods of authentication • The purpose of: ○ file repair/recovery software ○ backup/recovery software ○ file compression software ○ disk defragmentation software	• The characteristics of and threat posed by different types of malware • How anti-malware works • Why it is important to keep anti-malware up-to-date • Why unpatched software is a target for hackers • The function of a firewall • How ethical hacking and penetration testing help identify vulnerabilities • Commonly used social engineering tactics (phishing, pretexting, baiting, quid pro quo) used by hackers • The purpose of an acceptable use policy and what it typically includes • How data is protected by encryption • How backup and recovery procedures protect against data loss • How access control helps to protect systems and data • How a hacker can exploit a code vulnerability • Examples of bad coding practices and secure coding practices • How code reviews and audit trails help to identify vulnerabilities	• How bandwidth and latency affect the performance of a network • How to use bits per second (bps) to describe network speed • How data is transmitted along copper and fibre-optic cables • How high-speed broadband is delivered • How devices are connected on a wireless network • The characteristic of Wi-Fi, Bluetooth, RFID, Zigbee and NFC and give examples of their use • The characteristics of bus, star and mesh network topologies
Key Vocabulary	• CT: Problem Solving with Programming AT1 ○ Decomposition & Abstraction ○ Algorithms ○ Develop Code ○ Constructs ○ Data Types ○ Operators ○ Types of Errors	• CT: Problem Solving with Programming AT2 ○ Algorithms ○ Logic Gates ○ Operators • Principles: Data 2 ○ Binary ○ Data Representation	• CT: Problem Solving with Programming SpT1 ○ Algorithms ○ Subprograms • Principles: Hardware 1 ○ Hardware ○ Data Storage	• CT: Problem Solving with Programming SpT2 ○ Algorithms • Principles: Software	• CT: Problem Solving with Programming SuT1 ○ Algorithms ○ Input & Output ○ Searching & Sorting • Principles: Network Security, Cybersecurity & Programming Languages	• CT: Problem Solving with Programming SuT2 ○ Decomposition & Abstraction ○ Algorithms • Principles: Networks 1

	• Principles: Data 1
	  ○ Binary
	  ○ Data Storage					