COMPUTING SCIENCE

Year 11

What are the aims and intentions of this curriculum?

The aim of our Key Stage 3 Curriculum is to ensure students experience a broad and balanced experience in Computing, which prepares them effectively for the workplace and as active participants in the digital world. The curriculum offers a balanced approach which will equip students to use computational thinking, principles of information, how digital systems work and how to put this knowledge to use through programming, the creation of systems and a range of content. This curriculum also ensures that students can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems and ultimately are responsible, digitally literate, confident and creative users of information and communication technology. This curriculum also covers e-safety, with progression in the content to reflect the different and escalating risks that young people face as they get older. To provide students with a holistic experience, prepare them for future success, help them aspire and value Computer Science, Personal Social Health and Economic (PSHE) education and Careers Education (CE) are incorporated into the curriculum.

Term	Topics	Knowledge and key terms	Skills developed	Assessment
Autumn 1	**Data storage**	**<u>Knowledge</u>** • Convert positive denary whole numbers to binary numbers(up to and including 8 bits) and vice versa. • Add two binary integers together (up to and including 8 bits) • Convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa. • Convert binary integers to their hexadecimal equivalents and vice versa • Binary shifts • Have a knowledge of the two main character sets that are used to the computer uses to represents characters. • Understand that each pixel has a specific colour, represented by a specific code. • Recognize the effect on image size and quality when changing colour depth and resolution • Use of variables, constants, operators,	• Understand how to convert positive denary whole numbers to binary numbers(up to and including 8 bits) and vice versa. • Add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur • Convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa. • Convert binary integers to their hexadecimal equivalents and vice versa • Use Binary shifts to multiply and divide binary digits. • Use of binary codes to represent characters • Understand the relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.: ○ ASCII ○ Unicode	**FORMATIVE:** • Seneca • End of topic quiz • Home Work • Worksheets • Class Discussions **SUMMATIVE:** • End of Unit Test

	Programming fundamentals	• Use inputs, outputs and assignments statements
		• Use of the three basic programming constructs used to control the flow of a program:
		  ○ Sequence
		  ○ Selection
		  ○ Iteration (count- and condition-controlled loops)
		• Use Boolean operators AND, OR and NOT
		
		• Recognise and use Comparison operators and Arithmetic operators
		
		Comparison operators
		== Equal to
		!= Not equal to
		< Less than
		<= Less than or equal to
		> Greater than
		>= Greater than or equal to
		
		Arithmetic operators
		+ Addition
		MOD Modulus
		/ Division
		– Subtraction
		* Multiplication
		DIV Quotient
		^ Exponentiation (to the power)
		
		• The use of data types:
		  ○ Integer
		  ○ Real
		  ○ Boolean
		  ○ Character and string
		  ○ Casting
		
		Key terms
		  ○ Denary
			• How an image is represented as a series of pixels, represented in binary
			• Understand what is Metadata
			• Know the effect of colour depth and resolution on:
			  ○ The quality of the image
			  ○ The size of an image file
			  ○ Sound
			• Understand how sound can be sampled and stored in digital form
			• State the effect of sample rate, duration and bit depth on:
			  ○ The playback quality
			  ○ The size of a sound file
			
			• Practical use of the three basic programming constructs used to control the flow of a program
			
			• Ability to choose suitable data types for data in a given scenario
			• Understand that data types may be temporarily changed through casting, and where this may be useful
			
			Career Links - Computer Programmer	

		○ Decimal ○ Binary ○ Overflow error ○ Hexadecimal ○ Binary shift ○ Most significant bit ○ Least significant bit ○ character set ○ Metadata ○ Pixel ○ Colour depth ○ Resolution ○ Sample rate ○ Duration ○ Bit depth ○ Boolean operators ○ Variable ○ Constants ○ Control structures/ programming constructs ○ Casting ○ Data types		
Autumn 2	**Additional programming techniques**	**Knowledge** • The use of basic string manipulation • The use of basic file handling operations: ○ Open ○ Read ○ Write ○ Close • The use of records to store data • The use of SQL to search for data • The difference between Arrays and Lists • The use of arrays/ Lists when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D)	• Practical use of the additional programming techniques in a high-level language within the classroom • Ability to manipulate strings, including: ○ Concatenation ○ Slicing ○ substring • Where to use functions and procedures effectively • The use of the following within functions and procedures: ○ local variables/constants ○ global variables/constants ○ arrays/list manipulation	**FORMATIVE:** • Seneca • End of topic quiz • Home Work • Worksheets • Class Discussions **SUMMATIVE:** • Trial Exam

		• The use of sub programs (functions and procedures) to produce structured code • Random number generation **Key terms** ○ String manipulation ○ File handling ○ SQL ○ List ○ Arrays ○ Subprograms/ subroutines ○ Random numbers	• Use SQL commands to retrieve data: ○ SELECT ○ FROM ○ WHERE • Create and use random numbers in a program **Careers Links –** Programmer, Software Engineer, Systems analyst, Database administrator	
Autumn 2	**Networks and topologies**	**Knowledge** For students to have a knowledge of: • Types of network : ○ LAN (Local Area Network) ○ WAN (Wide Area Network) • The characteristics of LANs and WANs • Factors that affect the performance of networks • The different roles of computers in a client-server and a peer-to peer network • The hardware needed to connect stand-alone computers into a Local Area Network: • The internet as a network • Network topologies (Star and Mesh) **Key terms** ○ Wireless access points ○ Routers ○ Switches	• Understanding of different factors that can affect the performance of a network, e.g.: Number of devices connected Bandwidth • State the tasks performed by each piece of hardware in a network. • Identify a DNS's role in the conversion of a URL to an IP address • State Advantages and disadvantages of the Cloud • State Advantages and disadvantages of the Star and Mesh topologies **Careers Links –** Network administrator, Software Engineer, Systems analyst, Database administrator	**FORMATIVE:** • Group Presentation • Worksheet • Home work • Class Discussions • Seneca **SUMMATIVE:** • End of Unit Test

	Wired and wireless network, protocols and layers	○ NIC (Network Interface Controller/Card) ○ Transmission media ○ The Internet ○ DNS (Domain Name Server) ○ Hosting ○ The Cloud ○ Web servers and clients ○ Network topologies **<u>Knowledge</u>** For students to have a knowledge of: • Modes of connection: ○ Wired ○ Ethernet ○ Wireless ○ Wi-Fi ○ Bluetooth • Encryption • IP addressing and MAC addressing • Common protocols including: ○ TCP/IP (Transmission Control Protocol/Internet Protocol) ○ HTTP (Hyper Text Transfer Protocol) ○ HTTPS (Hyper Text Transfer Protocol Secure) ○ FTP (File Transfer Protocol) ○ POP (Post Office Protocol) ○ IMAP (Internet Message Access Protocol) ○ SMTP (Simple Mail Transfer Protocol **<u>Key terms</u>** Wired and wireless connections Encryption Protocols	• Compare benefits and drawbacks of wired versus wireless connections • Recommend one or more connections for a given scenario • Explain the principle of encryption to secure data across network • Connections • Explain IP addressing and the format of an IP address (IPv4 and IPv6) • Explain how a MAC address is assigned to devices; its use within a network • Explain how standards allows hardware/software to interact across different manufacturers/producers • Discuss the different types of protocols and their key features of each. • Tell how layers are used in protocols, and the benefits of using layers (4-layer TCP/IP model)	

		IP Addressing Layers		
Spring 1	**Languages** **The Integrated Development Environment (IDE)** **Threats to computer systems and networks** **Identifying and preventing vulnerabilities**	**<u>Knowledge</u>** For students to have a knowledge of: • Characteristics and purpose of different levels of programming language: ○ High-level languages ○ Low-level languages • The purpose of translators • The characteristics of a compiler and an interpreter • the tools that an IDE provides • Common tools and facilities available in an Integrated Development Environment (IDE): ○ Editors ○ Error diagnostics ○ Run-time environment ○ Translators • Forms of attack: ○ Malware ○ Social engineering, e.g. phishing, people as the 'weak point' ○ Brute-force attacks ○ Denial of service attacks ○ Data interception and theft ○ The concept of SQL injection • Principles of each prevention method • Common prevention methods: ○ Penetration testing ○ Anti-malware software ○ Firewalls ○ User access levels ○ Passwords	• Tell the differences between high- and low-level programming languages • Sate the need for translators • Identify the differences, benefits and drawbacks of using a compiler or an interpreter • Identify the tools that an IDE provides • Discuss how each of the tools and facilities listed can be used to help a programmer develop a program • Practical experience of using a range of these tools within at least one IDE • Discuss threats posed to devices/systems • State principles of each form of attack including: ○ How the attack is used ○ The purpose of the attack. • Discuss how to limit the threats posed listed in forms of attack. • Identify methods to remove vulnerabilities and the attacks that they may limit/prevent	**FORMATIVE:** • Group Presentation • Worksheet • Home work • Reflection writing • Class Discussions • Seneca **SUMMATIVE:** • End of Unit Test •

		○ Encryption ○ Physical security		
	Operating systems **Utility software**	**<u>Knowledge</u>** For students to have a knowledge of: • The purpose and functionality of operating systems: ○ User interface ○ Memory management and multitasking ○ Peripheral management and drivers ○ User management ○ File management (naming, allocating to folders, moving files, saving, etc) • The purpose and functionality of utility software • Why is utility software important • Identify utility system software: o Encryption software o Defragmentation o Data compression **<u>Key terms</u>** ○ Malware ○ Social engineering ○ SQL injection ○ Anti-Malware ○ Firewalls ○ Penetration testing ○ System Software ○ Encryption software ○ Defragmentation ○ Data compression	• State the types of system software • Discuss each function of an operating system. • Understand that computers often come with utility software, and how this performs housekeeping tasks **PSHE Links –** **Careers Links –** Programmer, Software Engineer, webpage designer	
Spring 2	**Revision** **Practical Programming Skills**	**<u>Knowledge</u>** • Revise for mock examination • Complete programming project in Python. The programming	• Review and sit mock examination. • Design and Create a solution to a given problem using OCR ERL and python • **P**	**FORMATIVE:** • Project

		task(s) will require skills within the following areas when programming: ○ Design ○ Write ○ Test ○ Refine **key terms** ○ OCR Exam Reference Language	• **Career Links -** Graphic	**SUMMATIVE:** • Trial Examination
Summer 1	**Revision and Examination preparation.**	**Knowledge** Review past papers	• Complete past examination paper	**FORMATIVE:** Discussions
Summer 2	**GCSE EXAMINATIONS**			