COMPUTER SCIENCE

Year 10

What are the aims and intentions of this curriculum?

The aim of our Key Stage 4 Curriculum is to ensure students experience a holistic approach to Computer Science, enabling them to understand and apply the fundamental principles and concepts of the national curriculum for computing and the OCR J277 Computer Science, including computational thinking, data storage and logic, which will enable them to develop and apply their analytic, problem-solving, and design skills with repeated practical experience of writing computer programs in order to solve such problems. Additionally, PSHE and relevant career opportunities are incorporated within the curriculum to enable students to further develop a sense of responsibility, competence and confidence in the way they conduct themselves.

Term	Topics	Knowledge and key terms	Skills developed	Assessment
Autumn 1	**1.2.4 DATA STORAGE** • **Numbers**	• How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa • How to convert binary integers to their hexadecimal equivalents and vice versa • Binary shifts	• Denary number range 0 – 255 • Hexadecimal range 00 – FF • Binary number range 00000000 – 11111111 • Understanding of the terms 'most significant bit', and 'least significant bit' • Conversion of any number in these ranges to another number base • Ability to deal with binary numbers containing between 1 and 8 bits e.g. 11010 is the same as 00011010 • Understand the effect of a binary shift (both left or right) on a number • Carry out a binary shift (both left and right)	• Spelling test • Seneca Online activities • Case Studies • End of topic quiz • Microsoft Teams collaborative activities • Home work • Quizizz gamified questions • Class Discussions • Topic Worksheets • Past Paper question sheets
	1.2.5 COMPRESSION	• The need for compression Types of compression: ○ Lossy ○ Lossless	• Common scenarios where compression may be needed • Advantages and disadvantages of each type of compression • Effects on the file for each type of compression	

Autumn 2	**2.1 – ALGORITHMS** • **2.1.2 Designing, creating and refining algorithms**	• Structure diagrams • Create, interpret, correct, complete, and refine algorithms using: ○ Reference language/high-level programming language • Identify common errors • Trace tables	• Produce simple diagrams to show: ○ The structure of a problem ○ Subsections and their links to other subsections • Complete, write or refine an algorithm Identify syntax/logic errors in code and suggest fixes • Create and use trace tables to follow an algorithm • Know the characteristics of positive and healthy friendships including: trust, respect, honesty, kindness, generosity, boundaries, privacy, consent and the management of conflict and reconciliation.	• Seneca Online activities • End of topic quiz • Microsoft Teams collaborative activities • Padlet • Home work • Class Discussions • Topic Worksheets • Past Paper question sheets • Completion of CISCO Networking Academy course: Python Essentials
	1.3 – COMPUTER NETWORKS, CONNECTIONS AND PROTOCOLS • **1.3.1 Networks and topologies**	• Types of network: ○ LAN (Local Area Network) ○ WAN (Wide Area Network) • 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: ○ Wireless access points ○ Routers ○ Switches ○ NIC (Network Interface Controller/Card) ○ Transmission media	• The characteristics of LANs and WANs including common examples of each • Understanding of different factors that can affect the performance of a network, e.g.: ○ Number of devices connected ○ Bandwidth • The tasks performed by each piece of hardware • The concept of the Internet as a network of computer networks • Concept of servers providing services (e.g. Web server " Web pages, File server " file storage/retrieval) • Concept of clients requesting/using services from a server	• Individual Presentations • Models • Home work • Class Discussions • Topic Worksheets

Spring 1	**2.2 – PROGRAMMING FUNDAMENTALS** • **2.2.3 Additional programming techniques**	• The use of basic string manipulation • The use of basic file handling operations: ○ Open ○ Read ○ Write ○ Close	• Practical use of the additional programming techniques in a high-level language within the classroom • Ability to manipulate strings, including: ○ Concatenation ○ Slicing • how to identify and access appropriate advice and support • describe how the world of work is changing and the skills that promote employability	• Practical activities for Python and Exam Reference Language (ERL)
	• **1.3.1 Networks and topologies cont.**	• The Internet as a worldwide collection of computer networks: ○ DNS (Domain Name Server) ○ Hosting ○ The Cloud ○ Web servers and clients • Star and Mesh network topologies	• A Domain Name Service (DNS) is made up of multiple Domain Name Servers • A DNS's role in the conversion of a URL to an IP address • Concept of servers providing services (e.g. Web server " Web pages, File server " file storage/retrieval) • Concept of clients requesting/using services from a server • The Cloud: remote service provision (e.g. storage, software, processing) • Advantages and disadvantages of the Cloud • Advantages and disadvantages of the Star and Mesh topologies • Apply understanding of networks to a given scenario • identify and use a variety of sources of careers information, including ICT	• Individual Presentations • Spelling test • Seneca Online activities • Case Studies • End of topic quiz • End of term test • Microsoft Teams collaborative activities • Home work • Class Discussions • Topic Worksheets • Past Paper question sheets

	• **1.3.2 Wired and wireless networks, protocols and layers**	• Modes of connection: ○ Wired • Ethernet ○ Wireless • Wi-Fi • Bluetooth • Encryption • IP addressing and MAC addressing • Standards • 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) • The concept of layers	• Compare benefits and drawbacks of wired versus wireless connection • Recommend one or more connections for a given scenario • The principle of encryption to secure data across network connections • IP addressing and the format of an IP address (IPv4 and IPv6) • A MAC address is assigned to devices; its use within a network • The principle of a standard to provide rules for areas of computing • Standards allows hardware/software to interact across different manufacturers/producers • The principle of a (communication) protocol as a set of rules for transferring data • That different types of protocols are used for different purposes • The basic principles of each protocol i.e. its purpose and key features • How layers are used in protocols, and the benefits of using layers; for a teaching example, please refer to the 4-layer TCP/IP model • to understand how the inappropriate use of mobile phones can contribute to accidents	• Booklet creation • Puzzles • mind maps
Spring 2	• **2.2.3 Additional programming techniques…cont**	• The use of records to store data • The use of SQL to search for data • The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D) • How to use sub programs (functions and procedures) to produce structured code • Random number generation	• Arrays as fixed length or static structures • Use of 2D arrays to emulate database tables of a collection of fields, and records • The use of functions • The use of procedures • Where to use functions and procedures effectively • The use of the following within functions and procedures:	• Individual Presentations • Seneca Online activities • Case Studies • End of topic quiz • End of term test

Summer 1	**2.3 – PRODUCING ROBUST PROGRAMS** - **2.3.1 Defensive design**	- Defensive design considerations: ○ Anticipating misuse ○ Authentication - Input validation - Maintainability: ○ Use of sub programs ○ Naming conventions ○ Indentation ○ Commenting	○ local variables/constants ○ global variables/constants ○ arrays (passing and returning) - SQL commands: ○ SELECT ○ FROM ○ WHERE - Be able to create and use random numbers in a program - to recognise when others are using manipulation, persuasion or coercion and how to respond - Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values - Understanding of how to deal with invalid data in a program - Authentication to confirm the identity of a user Practical experience of designing input validation and simple authentication (e.g. username and password) - Understand why commenting is useful and apply this appropriately	- Microsoft Teams collaborative activities. - Programming Practical - Topic Worksheets - Home work - Class Discussions - Electronic and hardcopy Portfolio - Past Paper question sheets
	- **2.3.2 Testing**	- The purpose of testing - Types of testing: ○ Iterative ○ Final/terminal - Identify syntax and logic errors - Selecting and using suitable test data: ○ Normal ○ Boundary ○ Invalid/Erroneous - Refining algorithms	- The difference between testing modules of a program during development and testing the program at the end of production - Syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated - Logic errors as errors which produce unexpected output - Normal test data as data which should be accepted by a program without causing errors	- Group Presentations - Spelling test - Seneca Online activities - Case Studies - End of topic quiz - End of term test - Microsoft Teams collaborative activities - Home work - Class Discussions - Topic Worksheets

	- **2.5.2 The Integrated Development Environment (IDE)**	- Common tools and facilities available in an Integrated Development Environment (IDE): ○ Editors ○ Error diagnostics ○ Run-time environment ○ Translators	- Boundary test data as data of the correct type which is on the very edge of being valid - Invalid test data as data of the correct data type which should be rejected by a computer system Erroneous test data as data of the incorrect data type which should be rejected by a computer system - Ability to identify suitable test data for a given scenario - Ability to create/complete a test plan - Knowledge of the tools that an IDE provides - 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	- Past Paper question sheets
Summer 2	**REVISION AND PROJECT BASED LEARNING** - **Architecture of the CPU** - **1.1.2 CPU performance** - **1.2 – Memory and storage**	- The purpose of the CPU: - Common CPU components and their function. - Von Neumann architecture - How common characteristics of CPUs affect their performance. 1.2.1 Primary storage (Memory) 1.2.2 Secondary storage	- the role peers can play in supporting one another (including helping vulnerable friends to access reliable, accurate and appropriate support) - What actions occur at each stage of the fetch-execute cycle - The role/purpose of each component and what it manages, stores, or controls during the fetch-execute cycle - The purpose of each register, what it stores (data or address) - The difference between storing data and an address - Why computers have primary storage - How this usually consists of RAM and ROM - Key characteristics of RAM and ROM	- Individual Presentations - Seneca Online activities - Case Studies - Worksheets - End of topic quiz - End of term test - Microsoft Teams collaborative activities. - Programming Practical - Home work - Class Discussions - Electronic and hardcopy Portfolio - Past Paper question sheets

	• **Mini Programming Project**	• Must include defensive design • Must produce a test plan by using suitable test data: Normal, Boundary, Invalid/Erroneous • Include a combination of the following: ○ The use of the three basic programming constructs. ○ The common Boolean operators AND, OR and NOT ○ The use of basic string manipulation ¨ The use of basic file handling operations ○ The use of arrays ○ use sub programs ○ Random number generation	• Why virtual memory may be needed in a system How virtual memory works • Transfer of data between RAM and HDD when RAM is filled • Why computers have secondary storage • Recognise a range of secondary storage devices/media • Differences between each type of storage device/medium • Compare advantages/disadvantages for each storage device • Be able to apply their knowledge in context within scenarios • to recognise that the way in which personal qualities, attitudes, skills and achievements are evaluated by others, affects confidence and self-esteem • use the outcomes of self-assessment to identify areas for development, build self-confidence and develop a positive self image	