Computer Science Curriculum Overview J277

	Year 10	Year 11
Autumn	**Overview – Component 1 - Computer Systems:** 1.1 - Systems Architecture 1.2 - Memory and storage 1.3 - Computer networks, connections and protocols	**Overview – Component 2 - Computational Thinking, Algorithms & Programming:** **Students will receive one lesson per week continuing the development of programming fundamentals moving later in the term to algorithm practice.** 2.2 - Programming fundamentals – *Runs alongside the delivery of the following* 2.4 - Boolean logic 2.1 - Algorithms 2.3 - Producing robust programs 2.5 - Programming languages and integrated development environments
	Skills **Architecture of the CPU (1.1)** The purpose of the CPU: The fetch-execute cycle, Common CPU components and their function: ((ALU (Arithmetic Logic Unit), CU (Control Unit), Cache, Registers)) Von Neumann architecture: ((MAR (Memory Address Register), MDR (Memory Data Register), Program Counter, Accumulator)) **CPU performance (1.1.)** How common characteristics of CPUs affect their performance: (Clock speed, Cache size, Number of cores) **Embedded systems (1.1)** The purpose and characteristics of embedded systems, Examples of embedded systems **Primary storage (Memory) (1.2)** The need for primary storage, The difference between RAM and ROM, The purpose of ROM in a computer system, The purpose of RAM in a computer system, Virtual memory. **Secondary storage (1.2)** The need for secondary storage, Common types of storage: (Optical, Magnetic, Solid state), Suitable storage devices and storage media for a given application, the advantages and disadvantages of different storage devices and storage media relating to these characteristics: (Capacity, Speed, Portability, Durability, Reliability, Cost). **Units (1.2)** The units of data storage: ((Bit, Nibble (4 bits), Byte (8 bits), Kilobyte (1,000 bytes or 1 KB), Megabyte (1,000 KB), Gigabyte (1,000 MB), Terabyte (1,000 GB), Petabyte (1,000 TB)), How data needs to be converted into a binary format to be processed by a computer, Data capacity and calculation of data capacity requirements. **Data storage (1.2)** Numbers How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa, How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur, 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. Characters The use of binary codes to represent characters, The term 'character set', 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) Images How an image is represented as a series of pixels, represented in binary, Metadata, The effect of colour depth and resolution on: (The quality of the image, The size of an image file) Sound	**Skills** **Programming fundamentals (2.2)** The use of variables, constants, operators, inputs, outputs and assignments, The use of the three basic programming constructs used to control the flow of a program: (Sequence, Selection, Iteration (count- and condition-controlled loops), The common arithmetic operators, The common Boolean operators AND, OR and NOT. **Data types (2.2)** The use of data types: (Integer, Real, Boolean, Character and string, Casting) **Additional programming techniques (2.2)** 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 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. **Boolean logic (2.4)** Simple logic diagrams using the operators AND, OR and NOT, Truth tables, Combining Boolean operators using AND, OR and NOT, applying logical operators in truth tables to solve problems. **Computational thinking (2.1)** Principles of computational thinking: (Abstraction, Decomposition, Algorithmic thinking) **Designing, creating and refining algorithms (2.1)** Identify the inputs, processes, and outputs for a problem, Structure diagrams, Create, interpret, correct, complete, and refine algorithms using: (Pseudocode, Flowcharts, Reference language/high-level programming language), Identify common errors, Trace tables. **Searching and sorting algorithms (2.1)** Standard searching algorithms: (Binary search, Linear search), Standard sorting algorithms: (Bubble sort, Merge sort, Insertion sort). **Testing (2.3)** 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.

Computer Science Curriculum Overview J277

	How sound can be sampled and stored in digital form, The effect of sample rate, duration and bit depth on: (The playback quality, The size of a sound file) **Compression (1.2)** The need for compression, Types of compression: (Lossy, Lossless) **Networks and topologies (1.3)** Types of network: LAN, WAN, 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 Internet as a worldwide collection of computer networks: ((DNS (Domain Name Server), Hosting, The Cloud, Web servers and clients, Star and Mesh network topologies))	
	Assessments *Mini Assessment 1 (1.1)* *Mini Assessment 2 (1.1 - 1.2)* *Supported by regular class tests / reflective tasks*	**Assessments** *Mini Assessment 1 (2.4)* *Mini Assessment 2 (2.4 & 2.1)* *Bi Weekly Mini Algorithm tests.* *November – Y11 MOCKS – Series 1 (Component 1 & 2)*
Spring	**Overview – Component 1 - Computer Systems:** 1.3 – Computer networks, connections and protocols 1.4 – Network security	**Overview – Component 2 - Computational Thinking, Algorithms & Programming:** 2.3 - Producing robust programs 2.5 - Programming languages and integrated development environments **Retrieval Practise** **Overview - Revision / Independent focused learning** Students use the tracking and monitoring in place to support their development through independent learning. This may take place as part of an identified group, pair work or through independent revision techniques. ***Exam questions will be used regularly throughout all lessons in line with retrieval practice activities to identify and develop areas of required attention.***
	Skills **Wired and wireless networks, protocols and layers (1.3)** 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. **Threats to computer systems and networks (1.4)** 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. **Identifying and preventing vulnerabilities (1.4)** Common prevention methods: (Penetration testing, Anti-malware software, Firewalls, User access levels, Passwords, Encryption, Physical security)	**Skills** **Defensive design (2.3)** Defensive design considerations: (Anticipating misuse, Authentication, Input validation, Maintainability: (Use of sub programs, Naming conventions, Indentation, Commenting) **Testing (2.3)** 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. **Languages (2.5)** 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 Integrated Development Environment – "IDE" (2.5)** Common tools and facilities available in an Integrated, Development Environment (IDE): (Editors, Error diagnostics, Run-time environment, Translators) All components
	Assessments *Mini Assessment 3 (1.1 - 1.3)* *Mini Assessment 4 (1.1 - 1.4)*	**Assessments** *Mini Assessment 3 (2.4 & 2.1 & 2.3)* *Final Assessment (Component 2 – Mock Paper)*

Computer Science Curriculum Overview J277

	Supported by regular class tests / reflective tasks	*March – Y11 MOCKS – Series 2 (Component 1 &2)*
Summer	**Overview – Component 1 - Computer Systems:** 1.5 – System software 1.6 – Ethical, legal, cultural and environmental impacts of digital technology **Overview – Component 2 - Computational Thinking, Algorithms & Programming** Programming Practice: 2.2 - Programming fundamentals Supporting learning of: 2.1 – Algorithms 2.3 - Producing robust programs	**Overview – Final Exam Preparation** Paired Exam paper Exam questions
	Skills **Operating systems (1.5)** The purpose and functionality of operating systems: (User interface, Memory management and multitasking, Peripheral management and drivers, User management, File management) **Utility software (1.5)** The purpose and functionality of utility software, Utility system software: (Encryption software, Defragmentation, Data compression) **Ethical, legal, cultural and environmental impact (1.6)** Impacts of digital technology on wider society including: (Ethical issues, Legal issues, Cultural issues, Environmental issues, Privacy issues), Legislation relevant to Computer Science: ((The Data Protection Act 2018, Computer Misuse Act 1990, Copyright Designs and Patents Act 1988, Software licences (i.e. open source and proprietary)) **Programming fundamentals (2.2)** The use of variables, constants, operators, inputs, outputs and assignments, The use of the three basic programming constructs used to control the flow of a program: (Sequence, Selection, Iteration (count- and condition-controlled loops), The common arithmetic operators, The common Boolean operators AND, OR and NOT. **Data types (2.2)** The use of data types: (Integer, Real, Boolean, Character and string, Casting) **Additional programming techniques (2.2)** 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 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. **Computational thinking (2.1)** Principles of computational thinking: (Abstraction, Decomposition, Algorithmic thinking) **Designing, creating and refining algorithms (2.1)** Identify the inputs, processes, and outputs for a problem, Structure diagrams, Create, interpret, correct, complete, and refine algorithms using: (Pseudocode, Flowcharts, Reference language/high-level programming language), Identify common errors, Trace tables. **Testing (2.3)** 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.	**Skills** All components ➢ **Retrieval Practice used throughout lessons to re-enforce learning** ➢ **Online application shown in each SOW** ➢ **Component 1 - Retrieval Practice used throughout lessons of component 2 teaching to re-enforce / strengthen learning** ➢ **Assessments delivered will be holistic to represent students' knowledge of all areas currently studied**
	Assessments ***Mini Assessment 5 (1.1 - 1.5)*** ***Final Assessment (Component 1 – Mock Paper)*** ***June / July – Y10 End of year exam (Component 1)***	**Assessments** ***Component 1 – External Exam MAY*** ***Component 2 – External Exam MAY***