Year 10 Topics

In year 10 we teach the following topics over the course of the year. Each topic draws on prior learning from previous years and builds on understanding from the KS3 programme of study. Each topic develops and deepens the Core knowledge that will underpin all areas of the curriculum at KS4 and KS5.

Topic	Rationale	Knowledge acquisition	Key vocabulary	Skills and enrichment
1.1 Systems Architecture	This component will introduce learners to the Central Processing Unit (CPU). Learners will need to be able to describe the purpose and rationale of the CPU as a FDE cycle.	• the purpose of the CPU • Von Neumann architecture: MAR (Memory Address Register) MDR (Memory Data Register) Program Counter Accumulator • common CPU components and their function: ALU (Arithmetic Logic Unit) CU (Control Unit) Cache • the function of the CPU as fetch and execute instructions stored in memory • embedded systems: purpose of embedded systems Examples of embedded systems.	• Von Neumann Architecture • CPU • Bus • Registers • Memory Address Register (MAR) • Memory Data Register (MDR) • Program Counter • Accumulator • Arithmetic and Logic Unit (ALU) • Control Unit (CU) • Cache	• Independence • Evaluation • Analysis • Literacy • Oracy • Research skills • Note taking skills

Topic	Rationale	Knowledge acquisition	Key vocabulary	Skills and enrichment
1.2 Memory	This component will introduce learners to the How a machine uses different types of memory	• the difference between RAM and ROM • the purpose of ROM in a computer system • the purpose of RAM in a computer system • the need for virtual memory • Flash memory	• CPU – Central processing unit • RAM – Random Access memory • VOLATILE – Gets emptied when the power is turned off • ROM – Read Only Memory • NON VOLATILE - Doesn't lose the data when the device is turned off • BIOS – Basic input/output system contains programs to load the hardware • FIRMWARE – Permanent software programmed into ROM • DISK THRASHING - when a computer's virtual memory subsystem is in a constant state of paging	• Independence • Evaluation • Analysis • Literacy • Oracy • Research skills • Note taking skills

Topic	Rationale	Knowledge acquisition	Key vocabulary	Skills and enrichment
1.3 Storage	This component will introduce learners to the Different types of storage and how they work	• the need for secondary storage • data capacity and calculation of data capacity requirements • common types of storage: optical magnetic solid state • suitable storage devices and storage media for a given application, and the advantages and disadvantages of these, using characteristics: capacity, speed, portability, durability, reliability, cost	• Storage • Hardware • Secondary storage • *Optical* • *Magnetic* • *Solid state*	Independence Evaluation Analysis Literacy Oracy Research skills Note taking skills
1.7 Systems Software	This component will introduce learners to different types of software	• the purpose and functionality of systems software • operating systems: user interface memory management/multitasking peripheral management and drivers user management file management • utility system software: encryption so fare defragmentation data compression, the role and methods of backup	• System Software • Operating System (OS) • Virtual Machine • Utility Programs • Device Drivers • Multitasking • Time Slice • Processor Management • User Interface • Graphical User Interface • Command Line Interface • Peripheral Management • File Management	Independence Evaluation Analysis Literacy Oracy Research skills Note taking skills

Topic	Rationale	Knowledge acquisition	Key vocabulary	Skills and enrichment
2.1 Algorithms	This component will support learners with their coding by creating algorithms in pseudocode and flow diagrams	• computational thinking abstraction decomposition algorithmic thinking • standard searching algorithms: binary search, linear search • standard sorting algorithms: bubble sort merge sort insertion sort • how to produce algorithms using: pseudocode using flow diagrams • interpret, correct or complete algorithms	• Computational thinking – the use of computers to solve problems. • Abstraction – representing 'real world' problems in a computer using variables and symbols and removing unnecessary elements from the problem. • Decomposition – breaking down a large problem into smaller sub-problems. • Algorithmic Thinking - identifying the steps involved in solving a problem.	Independence Evaluation Analysis Literacy Oracy Research skills Note taking skills
2.2 Programming Techniques	This component will introduce learners to the basics of programming	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 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	• Input • Output • Assignment • Pseudocode • Flowchart • Identifier • Variable • CONSTANT • Concatenation • RAM • Memory locations • Arithmetic	Independence Evaluation Analysis Literacy Oracy Research skills Note taking skills

		• the use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays • how to use sub programs (functions and procedures) to produce structured code • the use of data types: integer real Boolean character and string casting • the common arithmetic operators • the common Boolean operators	• Boolean • String • Concatenate (join) • Substring • Delimiter • SPLIT • Left • Mid • Right • Addition • Subtraction • Division modular division • integer division • negation • AND • OR • NOT • Greater than • Less than • Equal • not equal • Modulus • Quotient • Exponentiation	

Topic	Rationale	Knowledge acquisition	Key vocabulary	Skills and enrichment
2.3 Producing robust programs	This component will enable learners to become defensive programmers	• Defensive design considerations input sanitation/validation planning for contingencies anticipating misuse authentication • maintainability: comments Indentation • the purpose of testing • types of testing iterative final/terminal • how to identify syntax and logic errors • selecting and using suitable test data.	• Data Validation • Data Sanitisation • Authentication • Maintainability • Comments • Indentation	Independence Evaluation Analysis Literacy Oracy Research skills Note taking skills
2.4 Computational logic	This component will introduce learners to why data needs to be in binary form	• why data is represented in computer systems in binary form • simple logic diagrams using the operations AN, OR and NOT • truth tables • combining Boolean operators using AND, OR and NOT to two levels • applying logical operators in appropriate truth tables to solve problems • applying computing-elated mathematics + – / * Exponentition (^ MOD DIV)	• OR • NOT (Inverter) • A ∧ B • A ∨ B • ¬A • Logic Gate • Transistor • Bit (Binary Digit) • Logic Circuit • AND	Independence Evaluation Analysis Literacy Oracy Research skills Note taking skills

Topic	Rationale	Knowledge acquisition	Key vocabulary	Skills and enrichment
2.6 Data Representation	This component will introduce learners to how different types of data is represented by a computer	<u>Units – what is:</u> • bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte• how data needs to be converted into a binary format to be processed by a computer. <u>Numbers</u> • how to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa • how to add two 8 bit binary integers and explain overflow errors which may occur • binary shift • how to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa • how to convert from binary to hexadecimal equivalents and vice versa • check digits. <u>Characters</u> • 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	Binary Denary Conversion Transistor Bit Binary Denary Conversion Transistor Bit Nibble Byte Kilobyte Megabyte Gigabyte Terabyte Petabyte	Independence Evaluation Analysis Literacy Oracy Research skills Note taking skills

		(for example ASCII, extended ASCII and Unicode).Images • how an image is represented as a series of pixels represented in binary • metadata included in the file• the effect of colour depth and resolution on the size of an image <u>Sound</u> • how sound can be sampled and stored in digital form • how sampling intervals and other factors affect the size of a sound file and the quality of its playback: sample size, bit rate, sampling frequency. <u>Compression</u> • need for compression • types of compression: lossy lossless		
3.2 Analysis **3.2 Design** **3.3 Implementation** **3.4 Testing** **3.5 Evaluate**	This component will introduce learners to the systems life cycle	• how to analyse and identify the requirements for a solution to the problem • how to set clear objectives that show an awareness of the need for real world utility • how to use abstraction and decomposition design the solution to a problem • how to identify the data requirements for their system	Analysis Design Implementation Testing Evaluation	Independence Evaluation Analysis Literacy Oracy Research skills Note taking skills

		• how to identify test procedures to be used during and after development to check their system against the success criteria • how to use validation to ensure a robust solution to a problem.		