Mulberry Academy Woodside

Computer Science KS4

Curriculum Overview 2025 – 2026

At Mulberry Academy Woodside, the Computer Science curriculum is driven by ambition and the pursuit of academic excellence for all students. We are committed to delivering a challenging, inclusive, and knowledge-rich curriculum that equips students with the computational thinking, technical knowledge, and problem-solving skills needed to succeed in an increasingly digital world.

Our curriculum empowers all students, regardless of starting point, to understand how digital systems work, to think logically and creatively, and to apply their knowledge through purposeful programming and digital projects. Through regular reflection, evaluation, and refinement of their work, students build resilience, precision, and confidence, enabling them to achieve their full potential both within Computer Science and beyond.

	AUTUMN TERM		SPRING TERM		SUMMER TERM	
	TERM 1A	TERM 1B	TERM 2A	TERM 2B	TERM 3A	TERM 3B
Year 10 Overview	**(P1) Systems Architecture** • the purpose of the CPU • the fetch-execute cycle • common CPU components and their function: • - ALU (Arithmetic Logic Unit) • - CU (Control Unit) • - Cache • - Registers	**(P1) Systems Architecture** • the purpose of the CPU • the fetch-execute cycle • common CPU components and their function: • - ALU (Arithmetic Logic Unit)	**(P1) Primary storage (memory & Storage)** • 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 • ¨ Cache	**(P1) Secondary (memory & Storage)** • The need for secondary storage • Common types of storage: • Optical • Magnetic • Solid state • Suitable storage devices and	**(P1) Units** • How data needs to be converted into a binary format to be processed by a computer • Data capacity and calculation of data capacity requirements **(P1) Data storage**	**(P1)Ethical, legal, cultural, and environmental impacts of digital literacy** • Ethical Issues • Legal Issues • Cultural Issues • Environmental Issues • Privacy Issues

	• Von Neumann architecture: • - MAR (Memory Address Register) • - MDR (Memory Data Register) • - Program Counter • Accumulator **(P2) Computational thinking** • Principles of computational thinking: • Abstraction • Decomposition Algorithmic thinking **(P2) Programming fundamentals** • The use of variables, constants, operators, inputs, outputs and	• - CU (Control Unit) • - Cache • - Registers • Von Neumann architecture: • - MAR (Memory Address Register) • - MDR (Memory Data Register) • - Program Counter • Accumulator **(P2) Boolean Logic** • Simple Logic Diagrams using the operators AND, OR and NOT • Truth tables • Combining Boolean operators such as AND, OR and NOT. • Applying logical operators in truth tables to solve problems **(P2) Programming fundamentals** The use of the three basic programming constructs used to	**(P2) Designing, creating and refining algorithms** • 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	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 **(P2) Designing, creating and refining algorithms** • 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	• 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 **(P2) Programming Fundamentals** • String Manipulation • Random number	**(P2) Programming Fundamentals** • String Manipulation • Random number

		control the flow of a program: • Sequence • Selection • Iteration (count- and condition-controlled loops) • The common Boolean operators AND, OR and NOT		• Trace tables		
Year 11 Overview	**(P1) Ethical, legal, cultural, and Environmental impacts of digital literacy** • Ethical Issues • Legal Issues • Cultural Issues • Environmental Issues • Privacy Issues **(P2) Searching and sorting algorithms** • **Standard searching algorithms:** Binary search Linear search Standard	**(P1) Data storage** • **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	**(P1) Data storage Compression** • The need for compression • ¨ Types of compression: ➢ Lossy ➢ Lossless **(P1) Wired & Wireless Networks, Protocols & Layers** • Modes of connection: Wired Ethernet Wireless Wi-Fi Bluetooth Encryption	**(P1) Systems Software & Utility software** • The purpose and functionality of operating systems: User interface Memory management and multitasking Peripheral management and drivers User management File management • The purpose and functionality of utility software Utility system software:	**(P1) Network 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	Revision and Exam Preparation

	sorting algorithms: Bubble sort Merge sort Insertion sort **(P2) Programming Techniques** • The use of records to store data • The use of SQL to search for data	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** How sound can be sampled and stored in digital form The effect of sample rate, duration and bit depth on:	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 Forms of attack: Malware Social engineering, e.g. phishing, people	Encryption software Defragmentation **(P2) Producing Robust Programs** - Defensive Design - Testing - Anticipating Misuse - Authentication - Indentation - Commenting - Selecting & using suitable test data **(P1) Programming languages and Integrated Development Invironments** • 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 Revision and exam preparation Common tools and facilities available in an Integrated	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 **(P2) Testing** • The purpose of testing Types of testing: Iterative Final/terminal Identify syntax and logic errors Selecting and using suitable test data: Normal	

		The playback quality The size of a sound file **(P2) Programming Techniques** • The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D)	as the 'weak point' Brute-force attacks Denial of service attacks Data interception and theft The concept of SQL injection **(P2) Programming Techniques** • The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D) **(P2) Producing Robust Programs** - Defensive Design - Testing - Anticipating Misuse - Authentication - Indentation - Commenting - Selecting & using suitable test data	Development Environment (IDE): Editors Error diagnostics Run-time environment Translators	Boundary Invalid/Erroneous Refining algorithms Revision and Exam Preparation	