Computer Science curriculum overview

	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Year 7	How to use a computer safely and effectively. Using MS Office. The components of a computer system.	Data representation including binary conversion and units. Introduction to algorithms and programming.	Python programming fundamentals - Sequence, selection and data types.	Networks – components, the internet, threats, and how to stay safe online.	Python programming iteration – count controlled and condition controlled.	The moral, legal and environmental issues surrounding the increased use of technology.
Year 8	How to use a computer safely and effectively. Spreadsheets. The components of a computer system.	Data representation including hexadecimal conversion and units. Programming fundamentals.	Introduction to web design – HTML and CSS.	Website creation. Programming iteration – count controlled and condition controlled.	CAD – 2D and 3D sketching and modelling.	Designing, modelling and prototyping a product using CAD.
Year 9	How to use a computer safely and effectively. Comp systems - CPU and the FDE cycle, secondary storage, binary shift	Data representation including images, sound and character sets. Logic circuits.	Python programming, flowcharts and trace tables. Extended programming task.	Programming arrays. Searching and sorting algorithms.	Artificial Intelligence unit – how it works, the issues surrounding the increased use, how to use AI responsibly.	Mobile app development.
Year 10	Computer systems. Data representation. Programming – sequence and selection.	Binary operations and logic circuits. Programming – operators, string handling, iteration.	Primary and secondary memory. Data representation. Programming – arrays and file handling.	Hexadecimal. Software. Programming – sub programs. Testing.	Networks. Programming – pseudocode and flowcharts, maintainability, 2D arrays.	Ethical, legal and cultural issues. SQL. Programming – defensive design, trace tables, searching and sorting.
Year 11	Computer systems. Data representation. Programming – selection and iteration.	Logic. Primary and secondary memory. Programming – arrays, file handling and sub programs.	Hexadecimal, Software and networks. Programming – pseudocode and flowcharts, maintainability, 2D arrays.	Ethical, legal and cultural issues. SQL. Programming – defensive design, trace tables, searching and sorting.	Revision and exam practice.	
Year 12	Computer systems. Programming constructs: sequence, iteration, branching.	Software. Data representation. Computational thinking. Sub Routines.	Data structures. Boolean logic. Searching and sorting algorithms.	Assembly language. Compression and encryption. Databases.	Networks and protocols. HTML, CSS and Javascript.	NEA preparation. Object oriented programming.
Year 13	Computer legislation. Object oriented programming. NEA	Processors. Memory management. Computational thinking. NEA	Scheduling. Translators. NEA	NEA	Revision and exam practice.	

KS3

Throughout Key Stage 3, students develop a strong foundation in digital literacy, computer systems, and programming. They learn how to use technology responsibly and safely, understand how data is represented and processed, and build practical skills in coding, web design, and digital modelling. By the end of Year 9, students are equipped with a broad understanding of computing concepts and are prepared for further study in computer science.

In Year 7, students learn how to use a computer safely and effectively. They explore MS Office applications and the components of a computer system. They also begin to understand how data is represented using binary and are introduced to algorithms and basic Python programming, including sequence, selection, and data types. As the year progresses, they deepen their programming skills by learning iteration and consider the ethical, legal, and environmental impact of technology use.

Year 8 builds on the foundations of Year 7. Students continue to use computers safely and effectively, work with spreadsheets, and revisit computer components. They explore data representation through hexadecimal, develop programming skills, and are introduced to web design using HTML and CSS. They also engage in CAD-based design projects, creating 2D and 3D models and prototypes, and refine their programming through further practice with iteration.

In Year 9, students gain a more advanced understanding of computer systems, including the CPU, FDE cycle, and binary shifts. They study complex data representation such as images, sound, and character sets, and are introduced to logic circuits. Programming is extended to include arrays, flowcharts, and trace tables. They also study searching and sorting algorithms, explore artificial intelligence, and apply their skills through mobile app development.

KS4

The GCSE curriculum is split into two main areas which run simultaneously and complement each other in their delivery. Namely, these are ‘Computer Systems’ and ‘Algorithms and Programming’. At the start of the GCSE course, students learn how the different components of a computer system work together and how data is processed. This is a crucial starting point in developing a deeper understanding of how a computer works. Concurrently, students learn the basics of programming. This includes the theory behind the different programming constructs and how computers translate programming languages. This falls within the wider theme of data representation and students quickly become familiar with the idea that all digital information is represented by binary data.

By the end of the GCSE course, students are proficient programmers who are able to create efficient solutions to complex programming problems. Students are not only able to describe how a computer processes data, but how multiple computers and devices connect through networks and the internet. Furthermore, students are aware of both the power and the limitations of technology and the importance of creating technology that has a moral responsibility.