Computer Science Curriculum Overview - Year 12

	Unit	Details
Autumn One	**Introduction to NEA** **Computational Thinking** **Programming Techniques**	Pupils will be introduced to the Non-Exam Assessment (NEA), exploring its purpose, expectations, and formal requirements. Through Computational Thinking, they will develop core problem-solving skills, including decomposition, abstraction, pattern recognition, and algorithmic design. They will also build their Programming Techniques, learning to write efficient, maintainable code using a range of constructs such as selection, iteration, subroutines, and data handling.
Autumn Two	**NEA Formal Proposal** **Algorithms** **Exchanging Data**	Pupils will complete and refine their NEA Formal Proposal, ensuring it outlines a clear problem definition, rationale, success criteria, and initial design considerations. They will study Algorithms in greater depth, analysing standard algorithms and learning how to design and evaluate their own solutions. In Exchanging Data, pupils will examine methods of data transmission, file formats, compression, encryption, and how data is shared securely across digital systems.
Spring One	**NEA Design** **Data Structures** **OOP Paradigm**	Pupils will produce the full design for their NEA solution, including system architecture, data models, user interface design, and algorithmic planning. They will explore a range of Data Structures such as arrays, stacks, queues, linked lists, and trees, understanding when and why each structure is appropriate. Alongside this, pupils will develop their understanding of the Object-Oriented Programming Paradigm, applying concepts like classes, objects, inheritance, encapsulation, and polymorphism to create modular and extensible software.
Spring Two	**NEA Development** **Components of a Computer** **Software Development**	Pupils will begin developing their NEA solution, implementing their planned features and applying iterative development techniques. They will study the Components of a Computer, including the CPU, memory, storage, I/O, and system architecture. In Software Development, pupils will gain insight into development methodologies, testing strategies, debugging, documentation, and the full software development lifecycle.
Summer One	**NEA Development** **Systems Software and** **Applications Generation**	Pupils will continue developing their NEA product, refining functionality, improving efficiency, and producing evidence of their process. They will study Systems Software, including operating systems, utility programs, and the role of system management tools. They will also explore Applications Generation, understanding translators, compilers, interpreters, and how high-level programs are converted into executable machine code.
Summer Two	**NEA Development** **Data Types**	Pupils will continue developing their NEA solution, focusing on producing the first full draft of their project. They will consider how data is represented, including binary, hexadecimal, ASCII, and Unicode, as well as floating-point numbers for decimals. Pupils will apply binary arithmetic and bitwise manipulation to process and transform data efficiently.