Computer Science KS5 Curriculum Overview

	Links to KS4	Computer Science at A level builds straight upon the skills and knowledge learnt at GCSE as well as building entirely new skills in new areas. Students will dive deeper and gain and even stronger grasp of the theory of computer science, whilst also vastly improving their programming skills using different programming paradigms such as procedural (which they would have focused on at GCSE) and introducing new paradigms like an Object-Oriented Programming approach.					
Intent	**Statement of Intent**	In Computer Science our intent is for students to be fascinated and well-informed of how computers perform the incredibly complex tasks they perform every second whilst also giving students the tools to develop their very own incredible and complex tasks. Computer Science is a practical subject where students can apply the academic principles learned in the classroom to real-world systems. It's an intensely creative subject that combines invention and excitement. Students have the fantastic opportunity of devising their own programming project where they must identify a problem that would benefit by being solved computationally and actually going through the project life cycle to eventually produce a finished product for that problem.					
	Timeline	**Term 1 - 7 Weeks**	**Term 2 - 7 Weeks**	**Term 3 - 6 Weeks**	**Term 4 - 6 Weeks**	**Term 5 - 6 Weeks**	**Term 6 - 7 Weeks**
Implementation (Year 12)	**Year 12 Overview**	Students cover unit 1 in Year 12 which primarily prepares them for Paper 1 – Computer Systems. Students immediately begin building on their GCSE knowledge of data, learning about the brand new topic of floating point binary – the concept of how computers store decimal numbers, whilst also learning about the vast and complex methods that data is stored such as trees, graphs, stacks and queues and many more and how and when these should be used.					
	SOW	**1.1. Components of a computer and their uses**		**1.2. Software and software development**	**1.3. Exchanging data**	**1.4. Data types, data structures and algorithms**	**1.5. Legal, moral, cultural and ethical issues**
	Assessment Type & Unit Focus	RP1 - Terminology / initial content. 1 x 40 min test (Sept/Oct). The characteristics of contemporary processors, input, output and storage devices. Students will gain a deeper understanding into hardware, for example the types of computer processor, as well as understanding how exactly the computer performs the FDE cycle.		RP2/PPE1 - topic test made up of related past paper questions. Types of software and the different methodologies used to develop software. Students will gain a further understanding into the tasks performed by the OS, such as memory management, whilst further understanding how code is translation during the stages of compilation.	How data is exchanged between different systems. Student's will expend their knowledge on networks, databases, compression and encryption.	How data is represented and stored within different structures. Different algorithms that can be applied to these structures	RP3/PPE2 – Almost full paper 1 mock based on the 5 topics visited in Year 12. The individual moral, social, ethical and cultural opportunities and risks of digital technology. Legislation surrounding the use of computers and ethical issues that can or may in the future arise from the use of computers.
Implementation (Year 13)	**Year 13 Overview**	In year 2 of the course, students will need to continuously revisit the content taught in year 1, this will be done in the classroom through retrieval activities but must also be covered independently. The paper 1 content, mainly taught in Year 12 and revisited throughout Year 13 will also be tested again in the first PPE of Year 13. The majority of classroom activities will focus around Paper 2 - Algorithms and programming and also the support and development of the student's Programming project.					
	SOW	**2.1. Elements of computational thinking**		**2.2. Problem solving and programming**		**2.3 Algorithms**	
	Assessment Type & Unit Focus	PR1/PPE1 - Students must understand what is meant by computational thinking, looking at the 5 areas and how each can be used when solving problems and programming. - Thinking abstractly - Thinking ahead - Thinking procedurally - Thinking logically - Thinking concurrently		RP2 - Topic test made up of related past paper questions. How computers can be used to solve problems and programs can be written to solve them (Learners will benefit from being able to program in a procedure/imperative language and object oriented Language). Students will learn a variety of programming techniques such as modularity (functions and procedures) OOP, recursion, and more.		RP3/PPE 2 – Full paper 2 mock based on the 3 topics visited in Year 13. Students will learn about and write algorithms to solve complex problems. These include popular searching and sorting algorithms (bubble sort, insertion sort, merge sort, quick sort, Dijkstra's shortest path algorithm, A* algorithm, binary search and linear search). Students will also learn about the suitability of different algorithms for a given task and data set, in terms of execution time and space. They will learn about measures and methods to determine the efficiency of different algorithms such as the Big O notation (constant, linear, polynomial, exponential and logarithmic complexity).	
	Topic Texts	Computational Fairy Tales Paperback - *Kubica, Jeremy* A collection of Academic Magazines created by Cs4fn (Queen Mary University of London)					
Impact	**Year 12 Review Points**	RP1: T1, Wk3		RP2 (PPE1): T3, Wk2			RP3 (PPE2): T6, Wk4
	Year 13 Review Points		RP1 (PPE1): T2, Wk1		RP2: T4, Wk1	RP3 (PPE2): T5, Wk1	
	How It Is Used / Skills Set Developed / Outcomes	Throughout A level Computer Science, students develop an understanding and ability to apply the fundamental principles and concepts of computer science, including: abstraction, decomposition, logic, algorithms and data representation. The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so. The capacity to think creatively, innovatively, analytically, logically and critically. The capacity to see relationships between different aspects of computer science.					
	Links to Higher Education	Studying A level Computer Science can offer pathways into a variety of different high education routes. It is still considered one of the most challenging and diverse A levels available. Due to the nature of Computer Science, the university pathways are vast and varied, from strict computer science to cyber security to software development to even games design. Students will make significant strides with their Programming Project, providing links to higher education courses that require significant programming skills. Furthermore, the skills of project development – analysis, design, development, testing and evaluation.					
	Careers in the Curriculum	Throughout the course, we update our careers board using genuine examples of occupations. We also incorporate careers into starter activities looking at Computer Science related careers.					