KS5 Curriculum Overview: A Level Computer Science Year 13					
Rationale: Students have completed the majority of the coursework and the topics that support the coursework. In year 2 the focus changes from a practical element to a more a theory based to incorporate the remaining of the topics with a consolidation of everything covered to concentrate on the exams in preparation for the final exam. The programming project component draws on all the elements of the other components in order to enable learners to develop synoptic thought processes in their approach to the solving of a real life scenario. As part of this process, learners will draw on knowledge, understanding and skills that have been embedded in components 01 and 02.					
Term / Length of Unit	**Outline**	**Assessment**	**Home Learning**	**Resources**	**Knowledge/Skills End Points**
Y13 Autumn	**Unit 3** – Coursework, Introduce the evaluation and final testing of the coursework. Students to submit final project. **(WI lead)** **Unit 1.1 (BI lead)** The Characteristics of contemporary processors, input, output, and storage devices. Looking at the FDE cycle and the registers that are associated. The types of processors and the difference. With a recap from KS4 on input, outputs and storage. Make use of diagrams which depicts the fetch-decode-execute cycle, diagrams which shows the direction and connections of the 3 busses and diagrams which illustrates how	Internal assessment of the coursework. End of topic assessment – Zig Zag	Coursework Exam questions and revision for end of topic test	• Project resources • Individual action plan for final coursework submission • Types of secondary storage ppt. • Input output storage ppt. • Types of storage ppt • RAM and ROM ppt. • LMC ppt. • RAM, cpu, fde ppt	• Demonstrate understanding of the arithmetic logic unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Busses: data, address and control: How this relates to assembly language program. • Demonstrate understanding of fetch-decode-execute cycle, including its effect on registers.

	the various registers interact during a typical fetch-decode-execute cycle. Compare and contrast the differences between, and uses of CISC and RISC processors and the different types of processor systems such as multicore systems, parallel systems and co-processor systems (e.g. GPUs). Construct comparison which shows the differences (Adv. & Dis.) of RAM & ROM. Make use of diagrams to demonstrate a clear understanding of how magnetic, flash and optical medium are able to store representations of digital 1's & 0's. Virtual storage and under what situations it might be utilised.			• Intro to Von Neumann ppt. • Component parts of CPU PPT. • CPU registers in detail ppt. Complemented with Craig and Dave tasks. Zig zag resources. End of unit assessment task	• Be able to describe the factors affecting the performance of CPU, clock speed, number of cores, cache. • Demonstrate understanding of Von Neumann, Harvard and contemporary processor architecture. • Explain the differences between, and uses of, CISC and RISC processors • Demonstrate understanding of multicore and parallel systems • How different input output and storage devices can be applied as a solution of different problems • Demonstrate how different input output and storage devices can be applied as a solution of different problems • Explain the uses of magnetic, flash and optical storage devices. • Demonstrate understanding of RAM/ROM and virtual storage.

	Unit 1.5 (WI Lead) Legal, moral and ethical issues. The opportunity to explore how digital technology has an effects on the workforce, looking at automated system and artificial intelligence. Covering the laws that are relevant to computing. Start Consolidations Unit 2 – See below	End of topic assessment – Zig Zag Timetable plan	Exam questions and revision for end of topic test Students should produce an exam timetable to start revision	• Computer mis use act ppt. • Data protection Act ppt. • Copyright Design and patents act and RIPA ppt. • Ethical and cultural issues ppt. • Computers in the modern worls ppt. Complemented with Craig and Dave tasks. Zig zag resources. End of unit assessment task	• Demonstrate the ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology. • Pupils must be able to discuss moral/ethical and social implications regarding the areas below: - Computers in the workforce - Automated decision making - Artificial intelligence - Environmental effects - Censorship and the Internet - Monitor behaviour - Analyse personal information - Piracy, offensive communications. - Layout, colour • Explain the key points and facts, of each of the following acts of legislation: -The Data Protection Act (1998) -The Computer Misuse Act (1990) -The Copyright Design and Patents Act (1988) -The Regulation of Investigatory Powers Act (2000)

Y12 Spring	**Unit 1.2** Software and Software Development. Focusing on the role of system software and how it manages the operating system, memory and how data is scheduled. Alongside this, students will explore the range of additional applications that support the smooth running of a computer system. Finally looking at a range and types of programming. Pupils will study the need for, function and purposes of operating systems, Memory management (paging, segmentation and virtual memory), Interrupts, the role of interrupts and Interrupt Service Routines, Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time, Distributed, embedded, multi-tasking, multi-user and real time operating systems. Bios Device drivers and Virtual machines. Produce diagrammatic depictions of the various SDLC models. Produce comparisons (e.g. in a table) of the Adv. & Dis. of the various SDLC methodologies. Outlines the main testing strategies and	End of topic assessment – Zig Zag	Exam questions and papers.	• Software types ppt. • Software scheduling ppt. • Software development ppt. • Software development life cycles ppt. • Waterfall model ppt. • Alpha and beta testing ppt. Complemented with Craig and Dave tasks. Zig zag resources. End of unit assessment task	• Explain the advantages & disadvantages of paging vs segmentation. • Demonstrate understanding (e.g. via a diagram) how / when an interrupt is checked for during the fetch-decode-execute cycle. • Make comparisons of the key features of the various scheduling algorithms. • Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development • The relative merits and drawbacks of different methodologies and when they might be used • Writing and following algorithms • Different test strategies, including black and white box testing and alpha and beta testing

	describes when these test strategies might occur during the software development process. **Consolidation Unit 1** Students should RAG rate themselves and consolidation will be based on their personal checklist. Harder topics will be teacher led, but students will be expected to pull everything together using the topic booklets to help them plan their revision **Consolidation Unit 2** Students should RAG rate themselves and consolidation will be based on their personal checklist. Harder topics will be teacher led, but students will be expected to pull everything together using the topic booklets to help them plan their revision	Exam questions throughout both booklets	Students to take home an algorithm booklet to practice writing pseudocode. Independent revision	Topic booklets for each of each area. Overall workbook for revision for both areas. Mind map key areas of each.	• understand the internal workings of a CPU • How data is exchanged and is exchanged • Software development • Different data types and ethical issues. • Understand what is meant by computational thinking • The benefits of applying computational thinking to solving a wide variety of problems • Understand the principles of solving problems by computational methods • Be able to use algorithms to describe problems • Be able to analyse a problem by identifying its component parts.

Y12Summer	Exam practice and exam technique.		Independent revision	Variety of exam papers	