Year 12 Curriculum Overview

Rationale: The Year 12 curriculum is designed to encourage learners to be inspired, motivated and challenged by following a broad, coherent, practical, satisfying and worthwhile course of study. It will provide insight into, and experience of how computer science works, stimulating students' curiosity and encouraging them to engage with computer science in their everyday lives and to make informed choices about further study or career choices.

Term/Length of Time	Outline	Assessment/Teacher Feedback Opportunities	Homework and Literacy resources
Autumn 1	**The Characteristics of Contemporary Processors, Input, Output and Storage Devices** Students will be required to understand the different components of a computer and their uses - The structure and function of the processor (Arithmetic Logic Unit, Control Unit, Registers, Program Counter, Accumulator, Memory Address Register, Memory Data Register, Current Instruction Register, Buses) and how this relates to assembly language programs. The Fetch-Decode-Execute Cycle (including its effects on registers). The factors affecting the performance of the Central Processing Unit (CPU). The use of pipelining in a processor to improve efficiency. The Von Neumann, Harvard and contemporary processor	Sample examination questions at the end of each sub-topic completed as part of classwork. Formal end of topic assessments that include a mixture of open and closed questions with an additional focus on keywords/literacy. Completion of a set of Cornell Notes on the theory topics covered. A selection of written questions completed in class to assess understanding of programming A selection of programming challenges completed in class to assess understanding of programming techniques.	**Minimum homework expectation - to be set on G4S** Completion of six 30-minute revision/recall activities using an online platform called 'Smart Revise' that is bespoke for OCR A Level Computer Science. An additional 3 hours using the Computer Science Textbook completing chapter questions **Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list Complete some 'Quiz, Terms and Advance' questions using your Smart Revise platform login Access the Physics and Maths Tutor Computer Science revision section and complete revision tasks/activities on the topics covered plus access the past papers section and complete additional exam

	architectures. The differences between and uses of Complex Instruction Set Computer and Reduced Instruction Set Computer processors. Graphical Processing Units and their uses. Multicore and Parallel systems. How different input, output and storage devices can be applied to the solution of different problems. The uses of magnetic, flash and optical storage devices. Random Access Memory and Read Only Memory. Virtual storage. **Software** Students will be required to understand the types of software and the different methodologies used to develop software - The need for, function and purpose of operating systems. Memory Management. Interrupt Service Routines. Scheduling. Distributed, embedded, multi-tasking, multi-user and Real Time operating systems. BIOS. Device drivers. Virtual machines. The nature of applications. Utilities. Open-source vs closed-source software. Translators, Interpreters, compilers and assemblers. Stages		questions on topics covered (**pages are sometimes slow to load … be patient!**) Access W3Schools and learn a new Python programming technique

	of compilation. Linkers, loaders and use of libraries. Students will also be required to understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development including the relative merits and drawbacks of different methodologies and when they might be used. **Elements of Computational Thinking** Students will be required to understand what is meant by computational thinking including thinking abstractly, thinking ahead, thinking procedurally, thinking logically, thinking concurrently **Recap on Programming Techniques** Students will review, recap, develop and embed prior programming knowledge which will then be a precursor to an independent project utilising all of these skills.		

Autumn 2	**Software Development** Students will also be expected to understand and use assembly language (including following and writing simple programs with the Little Man Computer instruction set). **Exchanging Data** Students will be required to understand how data is exchanged between different systems by learning about compression, encryption and hashing techniques, and databases (flat file and relational). **Programming Techniques** Students will continue to embed prior programming knowledge by completing an independent project. Students will then be required to understand the use of Structured Query Language (SQL) when querying databases (theory and practical using Python).	Sample examination questions at the end of each sub-topic completed as part of classwork. Formal end of topic assessments that include a mixture of open and closed questions with an additional focus on keywords/literacy. Completion of a set of Cornell Notes on the theory topics covered. A selection of written questions completed in class to assess understanding of programming A selection of programming challenges completed in class to assess understanding of programming techniques.	**Minimum homework expectation - to be set on G4S** Completion of six 30-minute revision/recall activities using an online platform called 'Smart Revise' that is bespoke for OCR A Level Computer Science. An additional 3 hours using the Computer Science Textbook completing chapter questions **Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list Complete some 'Quiz, Terms and Advance' questions using your Smart Revise platform login Access the Physics and Maths Tutor Computer Science revision section and complete revision tasks/activities on the topics covered plus access the past papers section and complete additional exam questions on topics covered (**pages are sometimes slow to load … be patient!**) Access W3Schools and learn a new Python programming technique

Spring 1	**Exchanging Data** Students will gain a detailed understanding of networks including the importance of protocols and standards, the TCP/IP Stack, Domain Name System, Protocol layering, Local Area Networks, Wide Area Networks, Packet and circuit switching, Network security and threats, use of firewalls, proxies and encryption, Network hardware, Client-server and peer to peer, how Search Engine Indexing occurs including Page Rank Algorithm. **Revision** There will be an opportunity for students to revisit, embed and recap learning so far in order to successfully access the mock examination **Programming Techniques** Students will investigate and learn about more advanced programming techniques including recursion and Object Orientated Programming (OOP).	Sample examination questions at the end of each sub-topic completed as part of classwork. Formal end of topic assessments that include a mixture of open and closed questions with an additional focus on keywords/literacy. Completion of a set of Cornell Notes on the theory topics covered. A selection of written questions completed in class to assess understanding of programming A selection of programming challenges completed in class to assess understanding of programming techniques. Mock examination.	**Minimum homework expectation - to be set on G4S** Completion of six 30-minute revision/recall activities using an online platform called 'Smart Revise' that is bespoke for OCR A Level Computer Science. An additional 3 hours using the Computer Science Textbook completing chapter questions **Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list Complete some 'Quiz, Terms and Advance' questions using your Smart Revise platform login Access the Physics and Maths Tutor Computer Science revision section and complete revision tasks/activities on the topics covered plus access the past papers section and complete additional exam questions on topics covered (**pages are sometimes slow to load … be patient!**) Access W3Schools and learn a new Python programming technique

Spring 2	**Data Types and Data Structures** Students will be required to understand how data is represented and stored within different structures including common algorithms associated with these structures. This would include a detailed knowledge of data types including primitive data types, integers, real/floating point, characters, string and Boolean. How to represent positive integers in binary. The use of sign and magnitude and two's complement to represent negative numbers in binary. How to add and subtract binary integers. The representation of positive integers in hexadecimal. How to convert positive integers between binary hexadecimal and denary. The normalisation of floating-point numbers in binary. Floating point arithmetic, positive and negative numbers, addition and subtraction. An understanding of Bitwise manipulation and masks and how character sets are used to represent text.	Sample examination questions at the end of each sub-topic completed as part of classwork. Formal end of topic assessments that include a mixture of open and closed questions with an additional focus on keywords/literacy. Completion of a set of Cornell Notes on the theory topics covered. A selection of written questions completed in class to assess understanding of programming A selection of programming challenges completed in class to assess understanding of programming techniques.	**Minimum homework expectation - to be set on G4S** Completion of six 30-minute revision/recall activities using an online platform called 'Smart Revise' that is bespoke for OCR A Level Computer Science. An additional 3 hours using the Computer Science Textbook completing chapter questions **Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list Complete some 'Quiz, Terms and Advance' questions using your Smart Revise platform login Access the Physics and Maths Tutor Computer Science revision section and complete revision tasks/activities on the topics covered plus access the past papers section and complete additional exam questions on topics covered (**pages are sometimes slow to load … be patient!**) Access W3Schools and learn a new Python programming technique

	Algorithms Students will be required to understand the analysis and design of algorithms for a given situation, the suitability of different algorithms for a given task and data set, the measures and methods to determine the efficiency of different algorithms, Big O notation, the algorithms for the main data structures, and standard algorithms.		
Summer 1	**Data Types and Data Structures** Students will be required to understand common data structures including arrays, records, lists, tuples. How structures are used to store data as a linked-list, graph (directed and undirected), stack, queue, tree, binary search tree and hash table. There is also the requirement to know how to create, traverse, add data to and remove data from the data structures mentioned. There will also be the requirement to understand how to define problems using Boolean logic, Manipulate Boolean expressions, including the use of Karnaugh maps to simplify	Sample examination questions at the end of each sub-topic completed as part of classwork. Formal end of topic assessments that include a mixture of open and closed questions with an additional focus on keywords/literacy. Completion of a set of Cornell Notes on the theory topics covered. A selection of written questions completed in class to assess understanding of programming A selection of programming challenges	**Minimum homework expectation - to be set on G4S** Completion of six 30-minute revision/recall activities using an online platform called 'Smart Revise' that is bespoke for OCR A Level Computer Science. An additional 3 hours using the Computer Science Textbook completing chapter questions **Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list Complete some 'Quiz, Terms and Advance' questions using your Smart Revise platform login

	Boolean expressions, use rules to derive or simplify statements in Boolean algebra, use logic gate diagrams and truth tables. **Programming Techniques** Students will be required to understand the different web technologies required when exchanging data including HTML, CSS and JavaScript (both theoretical and practical exercises on HTML, CSS and JavaScript).	completed in class to assess understanding of programming techniques.	Access the Physics and Maths Tutor Computer Science revision section and complete revision tasks/activities on the topics covered plus access the past papers section and complete additional exam questions on topics covered (**pages are sometimes slow to load … be patient!**) Access W3Schools and learn a new Python programming technique
Summer 2	**Data Types and Data Structures** Students will be required to identify and apply the logic associated with D type flip flops, half and full adders. **Legal, Moral, Cultural and Ethical Issues** Students will be required to understand 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.	Sample examination questions at the end of each sub-topic completed as part of classwork. Formal end of topic assessments that include a mixture of open and closed questions with an additional focus on keywords/literacy. Completion of a set of Cornell Notes on the theory topics covered. A selection of written questions completed in class to assess understanding of programming	**Minimum homework expectation - to be set on G4S** Completion of six 30-minute revision/recall activities using an online platform called 'Smart Revise' that is bespoke for OCR A Level Computer Science. An additional 3 hours using the Computer Science Textbook completing chapter questions **Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list

	Revision There will be an opportunity for students to revisit, embed and recap learning so far in order to successfully access the mock examination **Project** Students will begin to formulate a synopsis for their independent programming project and start to develop and write up their project analysis.	A selection of programming challenges completed in class to assess understanding of programming techniques. Mock examination.	Complete some 'Quiz, Terms and Advance' questions using your Smart Revise platform login Access the Physics and Maths Tutor Computer Science revision section and complete revision tasks/activities on the topics covered plus access the past papers section and complete additional exam questions on topics covered (**pages are sometimes slow to load … be patient!**) Access W3Schools and learn a new Python programming technique