Year 10 Curriculum Overview

Rationale: The Year 10 curriculum is designed to complete knowledge acquisition for the final two topics of Unit 1 and re-introduce students to the concept of computational thinking from Year 8 including abstraction, decomposition and writing algorithms for familiar and unfamiliar scenarios. Students will be given the opportunity to utilise this new computational thinking and discover the different searching and sorting algorithms computers use to organise data and look at how Boolean logic is developed to create electronic circuits and decision-making. Students will also be given the opportunity to continue to undertake a range of programming tasks that will allow them to develop their skills to design, write, test and refine programs using a high-level programming language. Towards the end of the academic year students will begin to recap on prior learning from Year 9.

Term/Length of Time	Outline	Assessment/Teacher Feedback Opportunities	Homework and Literacy resources
Autumn 1	**System Software** Students will be required to understanding and gain knowledge on the purpose and functionality of operating systems including user interface, memory management and multitasking, peripheral management, drivers, user management and file management. This will lead into the purpose and functionality of utility software including encryption software, defragmentation and data compression. **Wider Issues Surrounding Computer Science** Students will develop their understanding of the impacts of digital technology on wider society including ethical, legal,	Differentiated recall 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. A selection of written questions completed in class to assess understanding of programming techniques.	**Minimum homework expectation - to be set on G4S** Completion of three (two theory + one programming) 30-minute revision/recall activities using an online platform called Smart Revise which is bespoke for OCR GCSE Computer Science. **Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Access BBC Bitesize and research more into System Software and Wider Computing Issues Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list some of which you may find in the department or the library Complete lesson 1 on System Software and lessons 1 through to 7 on Wider Impacts from the Oak National Academy Sign up to '100 Days of Code' online course and complete/embed more Python activities/knowledge

	cultural, environmental and privacy issues. This will include how to approach and answer essay style questions in the examination. This unit will also link to our International Schools focus by investigating moral and environmental issues surrounding develop and under developed countries. **Recap Sequence and Selection, Iteration and ID Arrays Programming Techniques** Students develop their use and understanding of: Variables, constants, operators, inputs, outputs and assignments. The three basic programming constructs used to control the flow of a program Sequence, Selection and Iteration (count- and condition-controlled loops). The common arithmetic and Boolean operators. The use of data types and casting.		Complete some 'Advance' questions using your Smart Revise platform login Choose another computing language to learn from W3Schools
Autumn 2	**Computational Thinking** Students will be required to understanding the principles of abstraction, decomposition and	Differentiated recall questions at the end of each sub-topic completed as part of classwork.	**Minimum homework expectation - to be set on G4S** Completion of three (two theory + one programming) 30-minute revision/recall activities using an online platform called Smart Revise which is bespoke for OCR GCSE Computer Science.

	algorithmic thinking and how they are used to define and refine problems. There will also be a requirement for students to produce simple diagrams to show the structure of a problem, subsections and their links to other subsections, to complete, write or refine an algorithm using the techniques learnt, to identify syntax/logic errors in code and suggest fixes and also create and use trace tables to follow an algorithm. **2D Arrays and Procedures Programming Techniques** Students will develop their programing skills further by investigating and coding 2D arrays to emulate database tables, fields, and records.	Formal end of topic assessments that include a mixture of open and closed questions with an additional focus on keywords/literacy. A selection of programming challenges completed in class to assess understanding of programming techniques.	**Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Access BBC Bitesize and research more into this topic Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list some of which you may find in the department or the library Complete lessons 1, 2 and 3 from Oak National Academy Sign up to '100 Days of Code' online course and complete/embed more Python activities/knowledge Complete some 'Terms' questions using your Smart Revise platform login Choose another computing language to learn from W3Schools
Spring 1	**Searching and Sorting Algorithms** Students will learn about the standard searching algorithms (Binary and Linear) and standard sorting algorithms (Bubble, Merge and Insertion). Students will need to be able to understand the main steps of each algorithm, any pre-	Differentiated recall 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	**Minimum homework expectation - to be set on G4S** Completion of three (one theory + two programming) 30-minute revision/recall activities using an online platform called Smart Revise which is bespoke for OCR GCSE Computer Science. **Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity

	requisites of an algorithm, apply the algorithm to a data set and identify an algorithm if given the code or pseudocode for it. **Functions and Parameter Passing Programming Techniques** Students learn how to use sub programs (functions and procedures) to produce structured efficient code. This would include an understanding of where to use functions and procedures effectively and the use of local variables, global variables and parameter passing. **Written Examination Question Technique - Programming** Students start to improve their ability to answer programming questions on Inputs, Outputs, Variables and Casting in an exam style setting.	additional focus on keywords/literacy. A selection of programming challenges completed in class to assess understanding of programming techniques.	Access BBC Bitesize and research more into this topic Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list some of which you may find in the department or the library Complete lessons 4 through to 10 from Oak National Academy Sign up to '100 Days of Code' online course and complete/embed more Python activities/knowledge Complete some 'Terms' questions using your Smart Revise platform login Choose another computing language to learn from W3Schools
Spring 2	**Defensive Program Design and Testing** Students will develop an understanding of the issues a programmer should consider to	Differentiated recall questions at the end of each sub-topic completed as part of classwork.	**Minimum homework expectation - to be set on G4S** Completion of three (two theory + one programming) 30-minute revision/recall activities using an online platform called Smart Revise which is bespoke for OCR GCSE Computer Science.

	ensure that a program caters for all likely input values. This will include an understanding of how to deal with invalid data in a program, authentication to confirm the identity of a user, practical experience of designing input validation and simple authentication (e.g. username and password) and an understanding of why commenting is useful and how to apply this appropriately. In addition students will look at testing techniques during development and at the end of production. Be able to spot syntax and logic errors and use test data (Normal, Boundary, Invalid and Erroneous). **Written Examination Question Technique - Programming** Students start to improve their ability to answer programming questions on Selection.	Formal end of topic assessments that include a mixture of open and closed questions with an additional focus on keywords/literacy. A selection of programming challenges completed in class to assess understanding of programming techniques.	**Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Access BBC Bitesize and research more into this topic Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list some of which you may find in the department or the library Complete lesson 4 from Oak National Academy Sign up to '100 Days of Code' online course and complete/embed more Python activities/knowledge Complete some 'Terms' questions using your Smart Revise platform login Choose another computing language to learn from W3Schools
Summer 1	**Boolean Logic** Students will study simple logic diagrams using the operators AND, OR and NOT and truth tables, They will use these skills	A series of knowledge based questions completed as part of classwork. Formal end of topic assessments that include a mixture of open and closed	**Minimum homework expectation - to be set on G4S** Completion of three (two theory + one programming) 30-minute revision/recall activities using an online platform called Smart Revise which is bespoke for OCR GCSE Computer Science. **Optional homework tasks and Literacy resources**

	to combine Boolean operators using AND, OR and NOT and apply logical operators in truth tables to solve logical problems. **Translators and IDE's** Students will develop and understanding of the characteristics and purpose of high-level and low-level languages. In addition understand the role and purpose of translators, compilers and interpreters when executing programs. **Written Examination Question Technique - Programming** Students start to improve their ability to answer programming questions on FOR Loops.	questions with an additional focus on keywords/literacy. Completion of a set of Cornell Notes on systems architecture and memory. A selection of programming challenges completed in class to assess understanding of programming techniques.	Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity Access BBC Bitesize and research more into Boolean Logic and Translators/IDE's topics Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list some of which you may find in the department or the library Complete lessons 9 and 10 from Oak National Academy Sign up to '100 Days of Code' online course and complete/embed more Python activities/knowledge Complete some 'Terms' questions using your Smart Revise platform login Choose another computing language to learn from W3Schools
Summer 2	**Recap on Systems Architecture and Memory** Students will revisit and gain a deeper understanding of the structure and purpose of the CPU which includes the fetch-execute cycle, common CPU components and the registers. Students will also look at the factors affecting the performance	A series of knowledge based questions 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.	**Minimum homework expectation - to be set on G4S** Completion of three (two theory + one programming) 30-minute revision/recall activities using an online platform called Smart Revise which is bespoke for OCR GCSE Computer Science. **Optional homework tasks and Literacy resources** Creation of revision resource (e.g. mind map) to be submitted alongside compulsory activity

	of a CPU such as Clock Speed, Number of Core and Cache Size. Students will be able to distinguish between a multi-purpose computer and an embedded system giving examples to demonstrate their understanding. Following this topic students will revisit various different types of primary and secondary storage methods and the need for these types of storage in computer systems. Their knowledge will be deepened further by understanding and explaining different storage devices and storage media suitable for a given application relating to capacity, speed, portability, durability, reliability and cost. **Written Examination Question Technique - Programming** Students start to improve their ability to answer programming questions on WHILE Loops.	Completion of a set of Cornell Notes on storage and networks. A selection of programming challenges completed in class to assess understanding of programming techniques. Progress exam covering components of paper 1 and paper 2.	Access BBC Bitesize and research more into Systems Architecture plus Memory and Storage topics Watch an episode of BBC Click on the BBC iPlayer Additional Reading for Budding Computer Scientists: Choose a book from this recommended reading list some of which you may find in the department or the library Complete lessons 1 through to 8 on Computer Systems pus lessons 1 through to 5 and lesson 9 on Memory from Oak National Academy Sign up to '100 Days of Code' online course and complete/embed more Python activities/knowledge Complete some 'Terms' questions using your Smart Revise platform login Choose another computing language to learn from W3Schools