

KS5 Curriculum Overview: A Level Computer Science Year 12

Rationale:

Students can study Computing A level without taking a GCSE in the subject, although the majority students will have potentially have studied it at GCSE. All students have had some form of computing from KS3 - so have the basics understanding of key terms, writing and following algorithms and some programming experience mainly in Python which will vary depending on if they took the subject at GSCE. With this in mind, in year 12 we recap some of the knowledge but introduce them to a more in-depth understanding of the subject. Students are introduced to another programming language in the form of VB, this levels out the playing field and gives students that have a good grasp of Python the opportunity to apply skills to a different programming language as well as build on existing knowledge, which means they have developed skills in two programming languages

Students will sit the exam at the end of KS5 with an aim of completing most of coursework before the start of year 13. This means that year 12 is designed to cover the theory topics that support the project.

Term / Length of Unit	Outline	Assessment	Home Learning	Resources	Knowledge/Skills End Points
Y12 Autumn	Unit 2 - 2.1 Elements of computational thinking Understand what is meant by computational thinking and the benefits of applying computational thinking to solving a wide variety of problems	Exam questions, and notes Followed by an end of topic module for 2.1	Each lesson has a homework sheet that complements the lesson a range of these can be picked from the main topic. Student will also be encourage to read the PG-Online book, make comprehensive notes as well as answer the exam questions in the book	<ul style="list-style-type: none"> Thinking abstractly Thinking ahead Thinking Procedurally Thinking logically , thinking concurrently, Thinking recognition Problem Solving Complemented with Craig and Dave tasks. End of unit assessment task	<ul style="list-style-type: none"> explain the differences between an abstraction and reality describe the need for reusable program components identify the inputs and outputs for a given situation interpret simple algorithms to describe their purpose give an example of how caching is used in a computer system Identify the components of a problem identify the points in a solution where a decision has to be taken give an example of a Divide and Conquer algorithm

	<p>2.2 Problem solving and programming</p> <p>How computers can be used to solve problems and programs can be written to solve them they should use the laptops or online text editors to demonstrating the skills in a practical manor</p>	<p>End of module assessment on: 2.2</p> <p>Completed programming tasks</p>	<p>Each lesson has a homework sheet that complements the lesson a range of these can be picked from the main topic. Student will also be encourage to read the PG-Online book, make comprehensive notes as well as answer the exam questions in the book as well as complete programming task at home. Students will need to be competent programmers to complete the project but also for Unit 2</p>	<ul style="list-style-type: none"> • Introduction to programming with VB PPT. Simple programming tasks with the Craig and Dave resources • Selection.PPT with practical VB task to complement the lessons. Speed checker, add and subtract programme • Iteration.PPT with a some example programming tasks • Subroutines.PPT using the tax program. • Completed a range of OPP from Craig and Dave and then programme 	<ul style="list-style-type: none"> • give examples of backtracking, data mining, heuristics, performance modelling, pipelining and visualisation • use an IDE to develop and debug a program • describe the use of an IDE to check for syntax errors • explain the difference between a variable and a constant • write a pseudocode solution for a simple problem involving iteration and selection (branching) • use nested selection and iteration statements • use arithmetic operations and Boolean operations NOT, AND and OR • use functions and library subroutines including random number generation • know how to define and call a subroutine (procedure or function) with parameters • construct algorithms using one-dimensional arrays • describe what is meant by recursion • define the OOP terms class, object, method, attribute, inheritance, encapsulation and polymorphism
--	--	--	---	--	--

	<p>Unit 1</p> <p>1.4 - Data types, data structures and algorithms.</p> <p>Understand how data is represented and stored within different structures.</p> <p>Characteristics of the primitive data types.</p> <p>Conversion between denary, hex and binary.</p> <p>Arithmetic (addition & subtraction), use of carries, lost carries, why computers don't use sign and magnitude for arithmetic, and performing floating point addition and subtraction.</p> <p>Also looking at how different algorithms that can be applied to a range of subjects. This can be taught alongside the programming unit so students have experience of programming the structures within</p>	<p>End of module assessments on:</p> <p>1.4.1</p> <p>1.4.2</p> <p>1.4.3</p>	<p>Each lesson has a homework sheet that complements the lesson a range of these can be picked from the main topic. Student will also be encourage to read the PG-Online book, make comprehensive notes as well as answer the exam questions in the book.</p>	<p>these Bloke, World Hardest game, BTEC calculator</p> <ul style="list-style-type: none"> • Binary to hexadecimal ppt. and worksheets to consolidate learning. • Binary arithmetic ppt. and worksheets to consolidate learning. • Karnaugh maps ppt. and worksheets to consolidate learning. • Data structures, tuples arrays lists ppt. and worksheets to consolidate learning. <p>Complemented with Craig and Dave tasks. Zig zag resources. End of</p>	<ul style="list-style-type: none"> • draw an inheritance diagram • Describe primitive data types, integer, real/floating point, character, string and Boolean • Represent positive integers in binary • Demonstrate the ability to use sign and magnitude and two's complement to represent negative numbers in binary • Demonstrate the ability to apply addition and subtraction of binary integers • Demonstrate the ability to represent positive integers in hexadecimal • Convert positive integers between binary hexadecimal and denary • Demonstrate understand positive and negative real numbers using normalised floating point representation • Demonstrate understanding of Arrays (of up to 3 dimensions), records, lists and tuples. • Demonstrate understanding of The properties of stacks and queues
--	--	---	---	--	---

				unit assessment task	
Y12 Spring	<p>Unit 1</p> <p>1.4.3 Boolean Algebra</p> <p>Boolean Algebra in the OCR A Level in Computer The unit begins with lessons on logic gates and Boolean algebra and covers Karnaugh maps, flip flops, half and full adders</p> <p>1.3 – Exchanging Data</p> <p>Understand relational databases. Use diagrams which shows your understanding of a relational database, e.g. keys, relationships etc.</p> <p>Understand and compare differences in lossy vs. lossless compression.</p> <p>Understand Internet structure, LAN's WAN'S and network topologies. Include diagrams</p>	<p>End of Module 1.4.3</p> <p>End of module assessments on:</p> <p>1.3.1 1.3.2 1.3.3</p>	<p>PG online Resources</p> <p>Each lesson has a homework sheet that complements the lesson a range of these can be picked from the main topic. Student will also be encourage to read the PG-Online book, make</p>	<ul style="list-style-type: none"> Boolean Theorems ppt. and worksheets to consolidate learning. Boolean algebra logic gates ppt. and worksheets to consolidate learning. Boolean Algebra- Boolean expressions ppt. and worksheets to consolidate learning. <p>Boolean Algebra</p> <ul style="list-style-type: none"> Database concepts ppt. Relational Databases ppt. and worksheets to consolidate learning. Entity relationship diagrams ppt. and worksheets 	<ul style="list-style-type: none"> Define problems using Boolean logic Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions Using logic gate diagrams and truth tables <ul style="list-style-type: none"> Demonstrate understanding of relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation and indexing Understand methods of capturing, selecting, managing and exchanging data Demonstrate understanding of Lossy v lossless compression

	<p>which explains the difference between packet and circuit switching.</p> <p>A clear understanding of the main differences between client-server and peer to peer.</p> <p>Unit 2</p> <p>2.3- Algorithms</p> <p>Searching and sorting algorithms (bubble sort, insertion sort, merge sort, quick sort) are explained in an interactive and practical way, with reference to Big-O notation in terms of time and space complexity. Tackles standard algorithms for depth-first and breadth-first graph traversals. Optimisation algorithms, such as Dijkstra's shortest path algorithm and the A* algorithm are covered along with a discussion of intractable problems, in the final topic</p>		<p>comprehensive notes as well as answer the exam questions in the book.</p> <p>Exam question on 2.3. Notes from the PG online book.</p>	<p>to consolidate learning.</p> <ul style="list-style-type: none"> • SQL ppt. and worksheets to consolidate learning. • Compression ppt. • Structure of the internet ppt. • Client server and peer to peer ppt. <p>Complemented with Craig and Dave tasks. Zig zag resources. End of unit assessment task</p>	<ul style="list-style-type: none"> • Demonstrate understanding of characteristics of networks and the importance of protocols and standards • Be able to explain client-server and Peer to Peer networks • state the order in which nodes are visited in pre-order and post-order tree traversals • give examples of linear, polynomial, exponential and logarithmic functions • compare two algorithms in terms of efficiency • explain the principles of a linear and binary search • explain how an insertion sort works
--	--	--	--	---	--

				explain the structure <ul style="list-style-type: none"> • Object-oriented design • Algorithms • User interfaces • Usability Features • Test plans 	
Y12Summer	Unit 1 1.3 Exchanging Data Understand Internet structure. Protocols and layering. Use diagrams which clearly show understanding of protocol layering and show how that applies to the TCP/IP protocol.	End of module assessment on: 1.3.3	Each lesson has a homework sheet that complements the lesson a range of these can be picked from the main topic. Student will also be encourage to read the PG-Online book, make comprehensive notes as well as answer the exam questions in the book.	<ul style="list-style-type: none"> • Structure of the internet ppt. • Internet protocols ppt. • . 	<ul style="list-style-type: none"> • Explain characteristics of networks and the importance of protocols and standards • Internet structure: -The TCP/IP Stack - DNS -Protocol layering -LANs and WANs -Packet and circuit switching
	Unit 3 – Programming Project Development/Testing Students to work independently on developing project – This will continue over the summer holidays	Completed Development over the summer holidays	Work on the development and testing of the solution	Introduce best way to lay out the development. Must be an iterative approach with testing happening throughout	Plan and organise the development of software using an iterative approach