

Computer Science

Computer Science							
		Term1		Term2		Term3	
		Term 1.1	Term 1.2	Term 2.1	Term 2.2	Term 3.1	Term 3.2
12 & 13	Theme	Computer Systems Algorithms & Programming		Computer Systems Algorithms & Programming		Computer Systems Algorithms & Programming	
	Concept	<p>Components of a computer and their uses</p> <p>Legal, moral, cultural and ethical issues</p> <p>Elements of computational thinking</p> <p>Project</p>	<p>Components of a computer and their uses</p> <p>Exchanging data</p> <p>Data types, data structures and algorithms</p> <p>Project</p>	<p>Data Types</p> <p>Software and software development</p> <p>Data types, data structures and algorithms</p> <p>Project</p>	<p>Data Types</p> <p>Software and software development</p> <p>Data types, data structures and algorithms</p> <p>Project</p>	<p>Data Structures</p> <p>Exchanging data</p> <p>Data types, data structures and algorithms</p> <p>Project</p>	<p>Boolean Algebra</p> <p>Exchanging data</p> <p>Data types, data structures and algorithms</p> <p>Project</p>
	Skills Knowledge	<p><u>Structure and function of the processor</u></p> <ul style="list-style-type: none"> <li>The Arithmetic and Logic Unit; ALU, Control Unit and Registers, Buses: data, address, and control</li> <li>The factors affecting the performance of the CPU, clock speed, number of cores, cache</li> <li>The fetch-decode-execute cycle, including its effect on registers,</li> <li>The use of pipelining in a processor to improve efficiency</li> <li>Von Neumann, Harvard, and contemporary processor architecture.</li> </ul> <p><u>Computing related legislation</u></p> <ul style="list-style-type: none"> <li>The Data Protection Act 1998</li> <li>The Computer Misuse Act 1990</li> <li>The Copyright Design and Patents Act 1988</li> <li>The Regulation of Investigatory Powers Act 2000.</li> </ul> <p><u>Moral and ethical Issues</u></p> <ul style="list-style-type: none"> <li>The individual moral, social, ethical, and cultural opportunities and risks of digital technology:               <ul style="list-style-type: none"> <li>computers in the workforce</li> <li>automated decision making</li> <li>artificial intelligence</li> <li>environmental effects</li> <li>censorship and the Internet</li> <li>monitor behaviour</li> <li>analyse personal information</li> <li>Piracy and offensive communications.</li> </ul> </li> </ul>	<p><u>Types of processors</u></p> <ul style="list-style-type: none"> <li>The differences between and uses of CISC and RISC processors, GPUs, and their uses (including those not related to graphics), Multicore and Parallel systems.</li> </ul> <p><u>Input, output, and storage</u></p> <ul style="list-style-type: none"> <li>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, RAM and ROM, Virtual storage.</li> </ul> <p><u>Networks</u></p> <ul style="list-style-type: none"> <li>Characteristics of networks and the importance of protocols and standards.</li> <li>The internet structure:               <ul style="list-style-type: none"> <li>The TCP/IP Stack.</li> <li>DNS</li> <li>Protocol layering.</li> <li>LANs and WANs.</li> <li>Packet and circuit switching.</li> </ul> </li> <li>Network security and threats, use of firewalls, proxies, and encryption.</li> <li>Network hardware.</li> <li>Client-server and peer to peer.</li> </ul>	<p><u>Data Types</u></p> <ul style="list-style-type: none"> <li>Primitive data types, integer, real/floating point, character, string, and Boolean,</li> <li>Represent positive integers in binary</li> <li>Use of sign and magnitude and two's complement to represent negative numbers in binary,</li> <li>Addition and subtraction of binary integers</li> <li>Represent positive integers in hexadecimal</li> </ul> <p><u>Systems Software</u></p> <ul style="list-style-type: none"> <li>The need for, function and purpose of operating systems</li> <li>Memory management (paging, segmentation, and virtual memory), Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the Fetch-Decode-Execute Cycle</li> <li>Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time</li> <li>Distributed, embedded, multi-tasking, multi-user and Real Time operating systems</li> <li>BIOS, device drivers, virtual machines</li> </ul>	<p><u>Data Types</u></p> <ul style="list-style-type: none"> <li>Convert positive integers between binary hexadecimal and denary</li> <li>Representation and normalisation of floating-point numbers in binary,</li> <li>Floating point arithmetic, positive and negative numbers, addition, and subtraction</li> <li>Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR,</li> <li>How character sets (ASCII and UNICODE) are used to represent text</li> </ul> <p><u>Applications Generation</u></p> <ul style="list-style-type: none"> <li>The nature of applications, justifying suitable applications for a specific purpose</li> <li>Utilities</li> <li>Open-source vs closed source</li> <li>Translators: Interpreters, compilers, and assemblers</li> <li>Stages of compilation (lexical analysis, syntax analysis, code generation and optimisation),</li> <li>Linkers and loaders and use of libraries.</li> </ul> <p><u>Software Development</u></p> <ul style="list-style-type: none"> <li>Waterfall lifecycle</li> <li>Agile methodologies</li> <li>Extreme programming</li> <li>Spiral model</li> <li>Rapid application development</li> </ul> <p><u>Types of Programming Language</u></p> <ul style="list-style-type: none"> <li>Need for and characteristics of a variety of programming paradigm</li> <li>Procedural languages</li> <li>Assembly</li> <li>Modes of addressing memory</li> <li>Object-oriented languages with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism.</li> </ul>	<p><u>Data Structures</u></p> <ul style="list-style-type: none"> <li>Arrays (of up to 3 dimensions), records, lists, tuples</li> <li>The following structures to store data: stack, queue, how to create, add data to and remove data from the data structures mentioned above (using arrays and procedural programming), the following structures to store data: linked-list, graph (directed and undirected), tree, binary search tree, hash table,</li> <li>How to create, traverse, add data to and remove data from the data structures mentioned above. (NB this can be either using arrays and procedural programming or an object-oriented approach).</li> </ul> <p><u>Databases</u></p> <ul style="list-style-type: none"> <li>Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation, and indexing.</li> <li>Methods of capturing, selecting, managing, and exchanging data.</li> <li>Normalisation to 3NF.</li> <li>SQL – Interpret and modify. See appendix 5d.</li> <li>Referential integrity.</li> <li>Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy.</li> </ul> <p><u>Compression, Encryption and Hashing</u></p> <ul style="list-style-type: none"> <li>Lossy vs Lossless compression run length encoding and dictionary coding for lossless compression,</li> <li>Symmetric and asymmetric encryption,</li> <li>Different uses of hashing.</li> </ul> <p><u>Web Technologies</u></p> <ul style="list-style-type: none"> <li>HTML, CSS, and JavaScript.</li> <li>Search engine indexing</li> <li>PageRank algorithm</li> <li>Server and client-side processing.</li> </ul>	<p><u>Boolean Algebra</u></p> <ul style="list-style-type: none"> <li>Define problems using Boolean logic. See appendix 5d.</li> <li>Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions.</li> <li>Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation.</li> <li>Using logic gate diagrams and truth tables. See appendix 5d.</li> <li>The logic associated with D type flip flops, half, and full adders.</li> </ul> <p><u>Compression, Encryption and Hashing</u></p> <ul style="list-style-type: none"> <li>Lossy vs Lossless compression run length encoding and dictionary coding for lossless compression,</li> <li>Symmetric and asymmetric encryption,</li> <li>Different uses of hashing.</li> </ul> <p><u>Web Technologies</u></p> <ul style="list-style-type: none"> <li>HTML, CSS, and JavaScript.</li> <li>Search engine indexing</li> <li>PageRank algorithm</li> <li>Server and client-side processing.</li> </ul>

		<p><u>Computational Thinking</u></p> <ul style="list-style-type: none"> <li>Thinking abstractly</li> <li>Thinking ahead</li> <li>Thinking procedurally</li> <li>Thinking logically</li> <li>Thinking concurrently</li> </ul> <p><u>Computational Methods</u></p> <ul style="list-style-type: none"> <li>Features that make a problem solvable by computational methods</li> <li>Problem recognition</li> <li>Problem decomposition</li> <li>Use of divide and conquer</li> <li>Use of abstraction</li> <li>Backtracking</li> <li>Data mining</li> <li>Heuristics</li> <li>Performance modelling</li> <li>Pipelining</li> <li>Visualisation to solve problems.</li> </ul> <p><u>A-Level Project – Yr12</u> Investigating programming languages</p> <p><u>A-Level Project – Yr13</u></p> <ul style="list-style-type: none"> <li>Development <ul style="list-style-type: none"> <li>Iterative development process</li> <li>Testing to inform development</li> </ul> </li> </ul>	<p><u>Algorithms</u></p> <ul style="list-style-type: none"> <li>Analysis and design of algorithms for a given situation</li> <li>The suitability of different algorithms for a given task and data set, in terms of execution time and space.</li> <li>Measures and methods to determine the efficiency of different algorithms</li> <li>Big O notation (constant, linear, polynomial, exponential and logarithmic complexity).</li> </ul> <p><u>A-Level Project – Yr12</u> Programming a variety of challenges using a language of choice</p> <p><u>A-Level Project – Yr13</u></p> <ul style="list-style-type: none"> <li>Development <ul style="list-style-type: none"> <li>Iterative development process</li> <li>Testing to inform development</li> </ul> </li> </ul>	<p><u>Algorithms</u></p> <ul style="list-style-type: none"> <li>Comparison of the complexity of algorithms.</li> <li>Algorithms for the main data structures, (stacks, queues, trees, linked lists, depth-first (post-order) and breadth-first traversal of trees).</li> <li>Standard algorithms (bubble sort, insertion sort, merge sort, quick sort, Dijkstra’s shortest path algorithm, A* algorithm, binary search and linear search).</li> </ul> <p><u>A-Level Project – Yr12</u> Programming a variety of challenges using a language of choice</p> <p><u>A-Level Project</u></p> <ul style="list-style-type: none"> <li>Evaluation <ul style="list-style-type: none"> <li>Testing to inform evaluation</li> <li>Success of the solution</li> </ul> </li> </ul>	<p><u>Programming Techniques</u></p> <ul style="list-style-type: none"> <li>Modularity, functions and procedures, parameter passing by value and by reference.</li> <li>Use of an IDE to develop/debug a program.</li> <li>Use of object-oriented techniques</li> </ul> <p><u>A-Level Project – Yr12</u> Deciding on a project idea</p>	<p><u>Programming Technique</u></p> <ul style="list-style-type: none"> <li>Modularity, functions and procedures, parameter passing by value and by reference.</li> <li>Use of an IDE to develop/debug a program.</li> <li>Use of object-oriented techniques</li> </ul> <p><u>A-Level Project – Yr12</u></p> <ul style="list-style-type: none"> <li>Analysis <ul style="list-style-type: none"> <li>Problem identification</li> <li>Stakeholders</li> <li>Research the problem</li> <li>Specify the proposed solution</li> </ul> </li> </ul>	<p><u>Programming Techniques</u></p> <ul style="list-style-type: none"> <li>Modularity, functions and procedures, parameter passing by value and by reference.</li> <li>Use of an IDE to develop/debug a program.</li> <li>Use of object-oriented techniques</li> </ul> <p><u>A-Level Project – Yr12</u></p> <ul style="list-style-type: none"> <li>Design <ul style="list-style-type: none"> <li>Decompose the problem</li> <li>Describe the solution</li> <li>Describe the approach to testing</li> </ul> </li> </ul>
	Wider Curriculum						