

Year 13 Computer Science

Year Calendar Plan

Dates	Lesson Focus	Assessment/Revision
	<p>Red text identifies additions to the calendar plan since September 2019</p> <p>Green text identifies topics that have been moved January 2020. Topics are revisited as needed throughout the year, for example by refreshing basics of operating systems from year 12 in term 2 of year 13 before moving on to more complex functions. Where topics are not specifically revisited in this plan, they should fall into spaced repetition or can be requested by students.</p>	
Term 1	<p>Coursework feedback – returns marked work with feedback, explain next steps and act on feedback</p> <p>Continue with coursework, looking at project work undertaken over the summer.</p> <p>Write up for development section of coursework</p> <p>1.4.2 Data Structures</p> <p>Recap 1.4.1 data types</p> <p>Linked list, graph (direct and undirected), stack, queue, tree, binary search tree, hash table. 1.4.2 Data Structures</p> <p>Arrays (of up to 3 dimensions), records, lists, tuples. 1.4.2 Data Structures Stacks and queues.</p> <p>How to create, travers, add data to and remove data from data structures.</p> <p>1.1.2 Types of processor – difference between uses of CISC and RISC processors</p> <p>Coursework – project development</p> <p>1.1.2 Types of processor – Multicore and Parallel systems, GPUs and their uses</p> <p>AW 1</p>	<p>Project research</p> <p>Questionnaires and interviews</p> <p>Coursework design</p> <p>Programming skills</p> <p>Project development</p> <p>Weekly spaced repetition test, quiz or recall session</p>
Term 2	<p>Coursework – project development / Evaluation</p> <p>1.2.1 Operating Systems – memory management, interrupts, the role of interrupts and interrupt service routines (ISR), role with the Fetch-Decode-Execute Cycle</p> <p>1.2.1 Operating Systems – scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time.</p> <p>BIOS</p> <p>Coursework – complete project development</p> <p>1.2.1 Operating systems – device drivers, virtual machines</p> <p>Coursework – Project testing</p> <p>1.2.2 Applications Generation – translators: interpreters, compilers and assemblers, stages of compilation (lexical analysis, code generation and optimisation), linkers and loaders and use of libraries</p> <p>Coursework – project evaluation</p> <p>1.2.4 Types of programming – need for and characteristics of a variety of programming paradigms, procedural language, assembly language, modes of addressing memory, OOP languages and understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism</p> <p>2.2.2 Computation methods</p> <p>Features that make a problem solvable by computational methods.</p> <p>Problem recognition</p> <p>Problem decomposition.</p>	<p>Project development</p> <p>Project development</p> <p>Project testing</p> <p>On-going revision</p> <p>Revision for PC2</p> <p>AW1</p> <p>Full paper 1</p> <p>Weekly spaced repetition test, quiz or recall session</p>
Term 3	<p>2.1 Elements of Computational Thinking</p> <p>Recap of computation thinking methods (2.1.1-2.1.4)</p> <p>2.1.5 Thinking Concurrently</p> <p>Determine the parts of a problem that can be tackled at the same time</p> <p>Outline the benefits and trade-offs that might result in concurrent processing in a particular situation.</p> <p>2.3.1 Algorithms</p> <p>Dijkstra's shortest path algorithm</p> <p>A* algorithm</p> <p>Traversing and finding shortest path</p> <p>Measures and methods to determine the efficiency of different algorithms, Big O notation (constant, linear, polynomial, exponential and logarithmic complexity)</p> <p>1.3.4 Web Technologies</p> <p>Search engine indexing</p> <p>PageRank algorithm</p> <p>Server and client side processing</p>	<p>Coursework good draft hand in</p> <p>PC3 part paper 1, part paper 2</p> <p>Weekly spaced repetition test, quiz or recall session</p>
Term 4	<p>1.3.1 Compression, Encryption and Hashing</p> <p>Lossy vs. Lossless compression</p> <p>Run length encoding and dictionary coding for lossless compression</p> <p>Symmetric and asymmetric encryption</p> <p>Programming OOP techniques through Java</p> <p>2.2.1 Programming Techniques</p> <p>Use of an IDE to develop/debug a program</p> <p>1.2.4 Software Development</p> <p>(a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development.</p> <p>(b) The relative merits and drawbacks of different methodologies and when they might be used.</p> <p>(c) Writing and following algorithms.</p> <p>2.3.1 Algorithms</p> <p>Algorithms for the main data structures, (stacks, queues, trees, linked lists, depth-first (post order) and breadth-first traversal of trees).</p> <p>Comparison of the complexity of algorithms.</p> <p>Logic associated with D type flip flops, half and full adders</p>	<p>PC4</p> <p>Coursework final hand in.</p> <p>Weekly spaced repetition test, quiz or recall session</p>
Term 5	Revision in preparation for the exam.	Weekly spaced repetition test, quiz or recall session
Term 6		

