

Long term planning grid

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 12	<p>UNIT 1 – Components of a Computer and their uses Duration: 9 hours What will be learnt:</p> <p>The basic internal components of the processor: ALU, Control Unit, registers and buses</p> <p>The different registers used in the Fetch-Decode-Execute cycle</p> <p>Factors which affect the performance of the CPU: clock speed, number of cores, cache</p> <p>Meaning of a multicore system and parallel processing</p> <p>The uses of RAM, ROM and virtual storage</p> <p>Typical uses of magnetic, flash and optical storage devices</p> <p>Assessment: Overall Unit assessment test</p> <p>UNIT 2 – Systems Software Duration: 9 hours What will be learnt:</p> <p>State the function and purpose of an operating system</p> <p>Describe scheduling algorithms: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time</p>	<p>UNIT 2 – Systems Software continued</p> <p>Assessment: Overall Unit assessment test</p> <p>Followed by:</p> <p>UNIT 3 – Software development Duration: 10 hours What will be learnt:</p> <p>list the stages in the waterfall lifecycle model</p> <p>name two other systems development models</p> <p>name and describe different types of testing</p> <p>write a pseudocode algorithm to solve a simple problem</p> <p>use a trace table to trace through an algorithm</p> <p>interpret simple algorithms to describe their purpose</p> <p>list two features of a good algorithm</p>	<p>Unit 4 Exchanging Data</p> <p>Duration: 10 hours What will be learnt:</p> <p>explain the difference between lossy and lossless compression and list advantages and disadvantages of each</p> <p>use basic encryption to create ciphertext</p> <p>encrypt and decrypt a message using the Caesar cipher</p> <p>explain the weaknesses of the Caesar cipher</p> <p>define the terms flat file, primary key, indexing</p> <p>define the terms relational database, foreign key, secondary key, entity</p> <p>draw a simple entity relationship diagram involving three or four entities</p> <p>state the properties of a database in Third Normal Form</p> <p>interpret a simple SQL statement</p>	<p>Unit 5 - Networks</p> <p>Duration: 12 hours What will be learnt:</p> <p>State the importance of protocols and standards</p> <p>Describe the structure of the Internet</p> <p>Explain the protocols used within the TCP/IP stack</p> <p>Demonstrate DNS in action using an IP address within a web browser</p> <p>Describe and identify examples of LANs and WANs</p> <p>Explain packet switching</p> <p>Provide examples of network threats and state methods to overcome these</p> <p>Explain the function of a firewall</p> <p>State the functions of a proxy server</p> <p>Create a basic webpage using HTML and some CSS</p> <p>Use JavaScript to make web form elements interactive and add validation</p>	<p>UNIT 6 – Data Types</p> <p>Duration: 10 hours What will be learnt:</p> <p>list primitive data types</p> <p>represent positive integers in binary</p> <p>use sign and magnitude and two's complement to represent negative numbers in binary</p> <p>add two unsigned binary numbers</p> <p>represent positive numbers in hexadecimal</p> <p>convert between denary, binary and hexadecimal number systems</p> <p>define bits and bytes, and use names, symbols and prefixes appropriately</p> <p>know how to use the ASCII table to represent text as binary</p> <p>explain why Unicode was introduced, and its advantages</p> <p>use arithmetic operations and Boolean operations AND, OR and XOR</p>	<p>Work on coursework development – programming</p> <p>Duration: 10 - 14 hours</p>

Long term planning grid

	<p>Describe distributed, embedded, multi-tasking, multi-user and real-time operating systems</p> <p>Describe the function of BIOS and device drivers</p> <p>Distinguish between systems software and applications software</p> <p>Describe what is meant by a utility program and give examples</p> <p>Be able to justify a suitable application for a specific purpose</p> <p>Distinguish between open source and closed source software</p> <p>State the roles of an assembler, compiler and interpreter</p> <p>Describe the use of libraries</p> <p>UNIT 10 – Computational Thinking</p> <p>Duration: 20 hours</p> <p>What will be learnt:</p> <p>Thinking abstractly, looking at the nature of abstraction, the need for abstraction and the differences between an abstraction and reality. Thinking ahead, identifying the inputs and outputs for a given situation. Determining the preconditions for devising a solution to a problem, the nature, benefits and drawbacks of caching and researching the need for reusable program components. Thinking procedurally, identifying the components of a problem and its solution and determining the order of</p>	<p>Define the term “programming paradigm” and give an example of two paradigms</p> <p>define the terms object, class, method, attribute, inheritance</p> <p>draw a simple inheritance diagram for a set of classes in an object-oriented approach</p> <p>follow through a simple program using the LMC instruction set</p> <p>Assessment:</p> <p>Overall Unit assessment test</p> <p>UNIT 11 – Programming Techniques</p> <p>Duration: 40 hours</p> <p>What will be learnt:</p> <p>Use an IDE to develop and debug a program, Describe the use of an IDE to check for syntax errors</p> <p>Explain the difference between a variable and a constant</p> <p>Write a pseudocode solution for a simple problem involving iteration and selection (branching)</p> <p>Use nested selection and iteration statements</p> <p>Use arithmetic operations and Boolean operations NOT, AND and OR</p>	<p>list methods of capturing data for input to a database</p> <p>list problems that can arise with a multi-user database</p> <p>Assessment:</p> <p>Overall Unit assessment test</p> <p>UNIT 11 – Programming Techniques - Continued</p>	<p>Describe the characteristics of the PageRank algorithm and state the factors that influence page ranking</p> <p>Assessment:</p> <p>Overall Unit assessment test</p> <p>UNIT 7 – Data Structures</p> <p>Duration: 20 hours</p> <p>What will be learnt:</p> <p>describe the concept and uses of a queue, stack, graph, tree, binary search tree and hash table</p> <p>list typical uses of each of these data structures</p> <p>know how an adjacency matrix and an adjacency list may be used to represent a graph</p> <p>traverse a binary tree in pre-order, in-order and post-order</p> <p>create a binary search tree</p> <p>be able to apply a simple hashing algorithm</p> <p>state what is meant by a collision and describe how collisions may be handled</p>	<p>show the effect of a logical shift left and shift right of a number of bits</p> <p>Assessment:</p> <p>Overall Unit assessment test</p> <p>UNIT 12 – Algorithms</p> <p>Duration: 20 hours</p> <p>What will be learnt:</p> <p>How to measure the time complexity of an algorithm using Big O notation</p> <p>Write an algorithm for a linear search and binary search, and measure how efficient they are</p> <p>Writing an algorithm for bubble sort, insertion sort, merge sort and quick sort, and tracing through them, and exploring their time complexities</p> <p>Outline the uses of graph traversal algorithms and their recursive nature</p> <p>Trace Dijkstra’s shortest path algorithm and explain how it works.</p> <p>Give examples of intractable problems</p> <p>The A* algorithm and its purpose</p>	
--	--	---	--	---	---	--

Long term planning grid

	<p>steps to get there, including sub-procedures. Thinking logically, identifying the points in a solution where a decision has to be taken. Thinking concurrently, determining the parts of a problem that can be tackled at the same time. Outline the benefits and trade offs that might result from concurrent processing in a particular situation Features that make a problem solvable by computational methods. Problem recognition, problem decomposition, use of divide and conquer, abstraction, learners should apply their knowledge of, backtracking, data mining, heuristics, performance modelling, pipelining, visualisation to solve problems</p>	<p>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 Draw an inheritance diagram Write complex algorithms involving data structures, subroutines and file-handling Interpret complex algorithms and determine the output Explain why using local variables makes a program easier to maintain Distinguish between passing parameters by value and by reference Write a recursive algorithm to solve a problem Use object oriented programming techniques to solve problems</p>		<p>Assessment: Overall Unit assessment test</p>		
--	--	---	--	---	--	--