

Term		KS4 1 Year GCSE	
	Topic	Knowledge	Skills/Assessment
<b>Term 1</b>	<b>2.4 Boolean Logic</b>	Students should be able to construct and interpret truth tables for simple logic circuits using AND, OR, NOT gates, create, modify, and interpret simple logic circuit diagrams.	<p>All topics begin with an initial assessment, and prior knowledge gaps are filled in before moving on to new content.</p> <p>Pupils complete end of topic tests throughout each half term to be marked by their teacher. This will allow gaps to be closed before the end of half-term assessment.</p> <p>At the end of each half-term there will be an assessment on all of the topic's pupils have studied in that block.</p>
	<b>1.2 Memory &amp; Storage</b>	Students should understand how computers store data, convert positive denary whole numbers (0-255) into 8-bit binary numbers, convert positive denary whole numbers (0-255) into 2-digit hexadecimal numbers and vice versa. They should also have an understanding of how binary is integral to how computers work and apply this knowledge to image resolution and sound sampling. Students will have an understanding of the purpose and differences between RAM and ROM and why primary storage, secondary storage and virtual memory are all required for computers to function.	
	<b>2.2 Programming</b>	Students will begin to understand and use the five main data types in their programming lessons (integer, real, Boolean, character, and string). They should be able to draw upon knowledge from the three topics this term to begin to write programs from algorithms.	
	<b>2.5 Programming Languages &amp; IDEs</b>	Students will be introduced to assembly language and begin to be able to understand the differences between programming languages and what they are used for.	
	<b>2.2 Programming</b>	Students should be able to transfer knowledge from previous programming and theory lessons to include arithmetic operators, string handling, conversion functions, sequence, selection and iteration in their programming work. During this term students will also start to understand the importance of subroutines and basic file handling operations.	
	<b>2.1 Algorithms</b>	Students will begin to understand the principles of computational thinking including abstraction, decomposition, and algorithmic thinking. They will also begin using pseudocode to write algorithms which include sequence, selection and insertion. Students should also have an understanding how to identify and correct errors in algorithms. Students will begin to understand how searching and sorting algorithms are used through computational thinking, flowcharts and tracing algorithms.	
	<b>1.1 System Architecture</b>	Students should be able to describe common Central Processing Unit components and Von Neumann architecture (Memory Access Registers, Memory Data Registers, programme counters including Arithmetic Logic Unit, Control Unit, Cache, Registers, and accumulators) and how these characteristics can affect the performance of computers.	
<b>Term 2</b>	<b>2.2 Programming</b>	Students should be able to transfer knowledge from previous programming and theory lessons to include arithmetic operators, string handling, conversion functions, sequence, selection and iteration in their programming work. During this term students will also start to understand how to write simple procedures and functions alongside using local variables in their subroutines.	<p>All topics begin with an initial assessment, and prior knowledge gaps are filled in before moving on to new content.</p> <p>Pupils complete end of topic tests throughout each half term to be marked by their teacher. This will allow gaps to be closed before the end of half-term assessment.</p> <p>At the end of each half-term there will be an assessment on all of the topic's pupils have studied in that block.</p>
	<b>2.3 Building robust programs</b>	Students will be able to refine their code and understand how the introduction of subroutines, appropriately named variables and comments can improve their code. They will also understand the differences in the types of errors they encounter and why they appear as well as the differences between iterative and final testing.	
	<b>1.3 Computer Networks, Connections &amp; Protocols</b>	Students will be able to explain how computer networks, connections and protocols are integral in how computers work. They will also understand how the internet works and how routers are utilised across different user groups.	
	<b>1.4 Network systems</b>	Students will be able to understand the different forms of attack and threats posed to a network and identify how these vulnerabilities can be prevented.	
	<b>1.5 System software</b>	Students should be able to discuss and understand the purpose and functionality of operating systems and common utility software. including user interface, memory management and multitasking, peripheral management and drivers, user management and file management.	
	<b>1.6 Ethics</b>	Students will have an understanding of how digital technology impacts wider society including ethical, cultural and environmental issues. They should also be able to describe the legislation and clauses relevant to Computer Science.	

**Term 3****Revision and exam preparation**

During the final part of the year pupils will complete whole class revision on areas of weakness identified by their teachers. They will also be signposted to specific areas of weakness for them to address individually and complete exam paper practise on whole papers and specific topics.

During this part of the year pupils will complete formal mock examinations. The results from this will be used alongside other data to write personalised revision plans for each class.

Students will regularly complete past exam papers during this period, including additional papers completed in exam conditions during lessons, and these will be marked by the class teacher and used to refine their planning.