

## Curriculum Plans: Year 9 (Computer Science)

	Topic	Knowledge: By the end of the unit students will know:	Skills: What skills will students have developed by the end of this unit?	Key terms: What new key terms and vocabulary will be learnt in this unit?	Summative Assessment: How will pupils be assessed in this unit?
Term 1	Data Representation	<ul style="list-style-type: none"> <li>Understanding binary numbers and why computers use them.</li> <li>Converting between binary, denary, and hexadecimal number systems.</li> <li>How binary is used to represent images and sound in computers.</li> <li>Understanding how sound is sampled and the factors that affect sound quality (e.g., sample rate).</li> <li>Basic understanding of logic gates (AND, OR, NOT) and their truth tables.</li> <li>The difference between sample rate, bit depth, and frequency in sound representation.</li> <li>The use of ASCII to represent text in binary.</li> <li>Basic data storage concepts (e.g., bits, bytes, kilobytes, megabytes).</li> </ul>	<ul style="list-style-type: none"> <li>Converting binary to denary and vice versa.</li> <li>Using binary addition and understanding binary laws.</li> <li>Creating and interpreting pixel images using binary values.</li> <li>Sampling sound waves and representing them in binary.</li> <li>Creating truth tables and performing calculations using logic gates.</li> <li>Understanding and using bitwise operations.</li> <li>Creating pixel art using binary values.</li> <li>Converting larger binary numbers (e.g., 16-bit) to denary and vice versa.</li> </ul>	<ul style="list-style-type: none"> <li>Binary</li> <li>Denary</li> <li>Hexadecimal</li> <li>Bit</li> <li>Byte</li> <li>Sample rate</li> <li>Bit depth</li> <li>Frequency</li> <li>ASCII</li> <li>Logic gates</li> <li>AND</li> <li>OR</li> <li>NOT</li> <li>Truth table</li> </ul>	<b>Written Assessment</b> – End of topic test
Term 2 and Term 3	Python – Advanced	<ul style="list-style-type: none"> <li>Understanding advanced Python constructs (e.g., while</li> </ul>	<ul style="list-style-type: none"> <li>Writing Python programs using control</li> </ul>	<ul style="list-style-type: none"> <li>Variable</li> <li>Loop</li> </ul>	<b>Practical Task</b> – Programming Test

## Curriculum Plans: Year 9 (Computer Science)

		<p>loops, for loops, if-elif-else statements).</p> <ul style="list-style-type: none"> <li>• Applying Boolean expressions and validating user input within loops.</li> <li>• Basic file handling and interaction with user input.</li> <li>• Using functions and subroutines to structure Python programs effectively.</li> <li>• Managing data using lists, appending/removing items, and performing operations on lists.</li> <li>• Understanding different data types in Python (e.g., strings, integers, floats, booleans).</li> <li>• Implementing error handling and debugging techniques.</li> <li>• Utilising logical and comparison operators for decision-making in programs.</li> <li>• Working with Python libraries for random number generation and string operations.</li> <li>• Developing practical Python programs.</li> </ul>	<p>flow structures like loops and conditionals.</p> <ul style="list-style-type: none"> <li>• Debugging common errors such as syntax errors (e.g., missing colons or indentation issues).</li> <li>• Developing programs to solve practical problems</li> <li>• Writing and using functions/subroutines, including the use of parameters and return values.</li> <li>• Using Python to perform list operations such as appending, removing, and searching.</li> <li>• Implementing validation checks for user inputs.</li> <li>• Structuring programs using control flow and loops to efficiently manage user input and output.</li> <li>• Creating interactive programs using while loops to manage input validation and iterative tasks.</li> </ul>	<ul style="list-style-type: none"> <li>• For</li> <li>• While</li> <li>• Boolean</li> <li>• Function</li> <li>• List</li> <li>• String</li> <li>• Data types</li> <li>• Syntax</li> <li>• Debugging</li> <li>• Subroutine</li> <li>• Procedure</li> </ul>	
--	--	---	--	---	--

## Curriculum Plans: Year 9 (Computer Science)

			<ul style="list-style-type: none"> <li>• Testing Python programs, identifying logical/syntax errors, and applying fixes.</li> </ul>		
Term 4	Computational Thinking and Algorithms	<ul style="list-style-type: none"> <li>• Understanding the key concepts of computational thinking (decomposition, pattern recognition, abstraction, and algorithms).</li> <li>• How searching algorithms (linear and binary) work and their applications.</li> <li>• The process of solving problems using algorithmic thinking, including designing flowcharts.</li> <li>• Understanding and implementing searching and sorting algorithms (e.g., binary search, merge sort, bubble sort).</li> <li>• The difference between various sorting algorithms (e.g., merge sort, bubble sort, insertion sort).</li> <li>• Basics of flowchart symbols and pseudocode for algorithm development.</li> <li>• The advantages and disadvantages of using specific algorithms like binary search and merge sort.</li> </ul>	<ul style="list-style-type: none"> <li>• Breaking down complex problems using decomposition to identify individual components.</li> <li>• Using pattern recognition to identify similarities in data sets or problems.</li> <li>• Creating and interpreting flowcharts and pseudocode to solve problems systematically.</li> <li>• Developing and refining algorithms to solve problems more efficiently.</li> <li>• Applying algorithms such as binary search and sorting techniques like bubble sort to solve tasks.</li> <li>• Creating effective algorithms and explaining their purpose (e.g., for searching, sorting).</li> </ul>	<ul style="list-style-type: none"> <li>• Computational thinking</li> <li>• Decomposition</li> <li>• Pattern recognition</li> <li>• Abstraction</li> <li>• Algorithm</li> <li>• Linear search</li> <li>• Binary search</li> <li>• Merge sort</li> <li>• Bubble sort</li> <li>• Insertion sort</li> <li>• Flowchart</li> <li>• Pseudocode</li> </ul>	<b>Written Assessment - Written Exam</b>

## Curriculum Plans: Year 9 (Computer Science)

		<ul style="list-style-type: none"> <li>- How to interpret and apply logic through flowcharts to visualize solutions.</li> </ul>	<ul style="list-style-type: none"> <li>Comparing different algorithms based on efficiency, accuracy, and complexity.</li> <li>Using abstraction to focus on relevant details of a problem while ignoring unnecessary complexities.</li> </ul>		
Term 5 and 6	Cyber Security	<ul style="list-style-type: none"> <li>Understanding the difference between data and information.</li> <li>The role and importance of the Data Protection Act.</li> <li>How data is collected, processed, and used by online platforms.</li> <li>Recognizing human errors (social engineering) as a threat to cybersecurity.</li> <li>Understanding different types of cyberattacks (e.g., DDoS, brute force).</li> <li>Knowledge of the Computer Misuse Act and its applications.</li> <li>Understanding malware types (e.g., viruses, worms, trojans, ransomware) and their impact on systems.</li> <li>The importance of secure passwords and two-factor</li> </ul>	<ul style="list-style-type: none"> <li>Identifying and critiquing online services based on data privacy practices.</li> <li>Recognising potential threats and breaches in online data privacy.</li> <li>Evaluating cybersecurity risks and how personal data can be compromised.</li> <li>Identifying and protecting against phishing, blagging, and shouldering attacks.</li> <li>Implementing strategies to prevent social engineering attacks (e.g., two-factor authentication).</li> <li>Designing secure systems using user authentication and</li> </ul>	<ul style="list-style-type: none"> <li>Data</li> <li>Information</li> <li>Data Protection Act</li> <li>Phishing</li> <li>Blagging</li> <li>Social engineering</li> <li>DDoS</li> <li>Brute force attack</li> <li>Computer Misuse Act</li> <li>Ransomware</li> <li>Malware</li> <li>Botnet</li> </ul>	<p><b>Written Assessment - Written Exam</b></p> <p>Assessed as part of the end of year exam</p>

## Curriculum Plans: Year 9 (Computer Science)

		<p>authentication in protecting accounts.</p> <ul style="list-style-type: none"><li>• How to assess and respond to cybersecurity threats in organizations.</li><li>• The role of network security tools (e.g., firewalls, anti-virus, penetration testing).</li></ul>	<p>permissions to minimize data breaches.</p> <ul style="list-style-type: none"><li>• Developing prevention strategies for malware, including anti-virus software and firewalls.</li><li>• Recognising different types of malware and how they exploit vulnerabilities in systems.</li><li>• Assessing and prioritising cybersecurity measures based on risk and impact.</li><li>• Managing cybersecurity resources to protect a network from common threats.</li></ul>		
--	--	---	---	--	--