

## Curriculum Plans: Year 12 (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?
Michaelmas 1	<p style="color: red;">NEA (Mixed in throughout the term)</p> <ul style="list-style-type: none"> <li>2.3 Basic Algorithms</li> <li>2.1 Elements of computational thinking</li> </ul>	<p><b>2.3.1 Algorithms – Basic Sorting and Searching Algorithms</b> (Linear, Binary searches – Bubble, Insertion, Merge, Quick Sorting Algorithms)</p> <p><b>2.1.1-2.1.4 Elements of Computational Thinking:</b> (Think abstractly, Thinking Ahead, Thinking Procedurally, Thinking logically, Thinking concurrently)</p>	<ul style="list-style-type: none"> <li>Implementation of basic search and sort algorithms.</li> <li>Using abstraction, thinking ahead, and recognizing patterns in problem-solving.</li> <li>Application of procedural, logical, and concurrent thinking approaches to problem solving.</li> </ul>	Linear Search Binary Search Bubble Sort Insertion Sort Merge Sort Quick Sort Abstraction Concurrency Logical Thinking Procedural Thinking	<p><b>Written Assessment –</b> End of topic test</p>
Michaelmas 2	<ul style="list-style-type: none"> <li>2.2.2 Software Development Lifecycle</li> </ul> <p>Mini NEA</p>	<p><b>2.2.2 Software Development:</b> (a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development. (b) The relative merits and drawbacks of different methodologies and when they might be used. (c) Writing and following algorithms. (d) Different test</p>	Various software development methodologies such as Waterfall, Agile, Extreme Programming, Spiral Model, and Rapid Application Development (RAD). Relative merits and drawbacks of each methodology, including when to use them. Writing algorithms and understanding	<ul style="list-style-type: none"> <li><b>Waterfall Model</b></li> <li><b>Agile</b></li> <li><b>Extreme Programming</b></li> <li><b>Spiral Model</b></li> <li><b>Rapid Application Development (RAD)</b></li> <li><b>Black Box Testing</b></li> <li><b>White Box Testing</b></li> <li><b>Alpha/Beta Testing</b></li> </ul>	<p><b>Written Assessment –</b> Written Exam</p>

## Curriculum Plans: Year 12 (Computer Science)

		<p>strategies, including black and white box testing and alpha and beta testing. (e) Test programs that solve problems using suitable test data and end user feedback, justify a test strategy for a given situation). IDE</p>	<p>different testing strategies (black box, white box, alpha, and beta testing).</p> <p>Ability to follow and write algorithms. Using integrated development environments (IDEs) for testing and debugging. Applying different test strategies and gathering feedback to improve software.</p>		
Lent 1	<ul style="list-style-type: none"> <li>• <b>2.2.2 Applied Computational Methods</b></li> <li>• <b>2.2.1 Programming Techniques</b> <ul style="list-style-type: none"> <li>○ C#</li> </ul> </li> </ul>	<p><b>2.2.2 Applied Computational Methods:</b> (Problem recognition, Decomposition, Divide and Conquer, Backtracking, Performance Modelling)</p> <p>Problem Solving and Programming (Sequence, Branching, Iteration, Global and Local Variables, Functions and Procedures, Modules, Class Structure, Recursion), IDE</p>	<ul style="list-style-type: none"> <li>• Problem-solving using computational methods like decomposition and divide and conquer.</li> <li>• Implementing recursion, and using local and global variables.</li> <li>• Structuring code using functions, procedures, and modules.</li> </ul>	<ul style="list-style-type: none"> <li>• Decomposition</li> <li>• Divide and Conquer</li> <li>• Backtracking</li> <li>• Recursion</li> <li>• Performance Modeling</li> <li>• Global/Local Variables</li> <li>• Functions and Procedures</li> <li>• Modules</li> </ul>	<p><b>Written Assessment –</b> End of topic test</p>

## Curriculum Plans: Year 12 (Computer Science)

Lent 2	<ul style="list-style-type: none"> <li>• <b>1.4.3 Boolean Algebra</b></li> </ul>	Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions	<ul style="list-style-type: none"> <li>• Ability to simplify and manipulate Boolean expressions.</li> <li>• Utilizing Karnaugh maps for logical simplifications in programming.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Boolean Algebra</b></li> <li>• <b>Karnaugh Maps</b></li> <li>• <b>Simplification</b></li> </ul>	<b>Written Assessment –</b> Written Exam
Trinity 1	<ul style="list-style-type: none"> <li>• <b>2.3.1 Algorithms: Data Structures –</b></li> </ul>	<ul style="list-style-type: none"> <li>• Stacks and Queues</li> <li>• Trees, Binary Trees</li> <li>• Graphs</li> </ul>	Implementing and traversing data structures (stacks, queues, trees, graphs).	<ul style="list-style-type: none"> <li>• <b>Stacks</b></li> <li>• <b>Queues</b></li> <li>• <b>Binary Trees</b></li> <li>• <b>Graphs</b></li> <li>• <b>Dequeue</b></li> <li>• <b>Enqueue</b></li> <li>• <b>Peek</b></li> <li>• <b>Pop</b></li> <li>• <b>Push</b></li> <li>• <b>IsFull</b></li> <li>• IsEmpty</li> <li>• Inorder, PreOrder, PostOrder</li> <li>• Breadth First, Depth First</li> </ul>	<b>Written Assessment –</b> Essay Question  Stack and Queues Assessment – Tower of Hanoi (40 marks)
Trinity 2	<ul style="list-style-type: none"> <li>• <b>2.2.1 Programming Techniques</b></li> </ul>	Windows Forms to display data, handle user input, and deploy applications	Developing applications using Windows Forms. Managing user input and output through GUI design.	<ul style="list-style-type: none"> <li>• <b>Windows Forms</b></li> <li>• <b>User Input</b></li> <li>• <b>Application Deployment</b></li> </ul>	

## **Curriculum Plans: Year 12 (Computer Science)**