





## Curriculum Overview for Computer Science Year 11

<p><b>Half Term 4</b> <b>Logic, Defensive Design and Testing</b></p>		<p>Model reading Reading out loud Skim and Scan of source information Decoding terms Etymology of key terms</p>
<p><b>Substantive Knowledge:</b></p> <ul style="list-style-type: none"> <li>• Understand why logic is important in computer programming</li> <li>• Understand what AND means</li> <li>• Understand what OR means</li> <li>• Understand what NOT means</li> <li>• Understand what a Truth Table is used for</li> <li>• Understand how to combine logic gates into a logic diagram</li> <li>• Understand the purpose of testing</li> <li>• Understand the purpose of iterative testing</li> <li>• Understand the purpose of final/terminal testing</li> <li>• Understand the types of errors that happen in programming</li> <li>• Understand the concept of test data</li> <li>• Understand the need for boundary test data</li> <li>• Understand the need for Erroneous test data</li> <li>• Understand the concept of defensive design</li> <li>• Understand why a program may be misused</li> <li>• Understand the need for authentication</li> <li>• Understand the need for code maintainability</li> <li>• Understand the use of subprograms</li> <li>• Understand the difference between a subroutine and a function</li> <li>• Understand the need for naming conventions</li> <li>• Understand the need for indentation</li> <li>• Understand the need for comments</li> </ul>		<p></p> <p>Logic AND gate OR gate NOT gate Truth Table Logic diagram Testing Iterative Testing Terminal Testing Errors Syntax Errors Runtime Errors Logic Errors Normal Data Boundary data Erroneous data Defensive Design Misuse Authentication Maintainability Subprograms Subroutine Function Arguments Naming Convention Camel Case Snake Case Indentation Comments</p>
<p><b>Disciplinary Knowledge:</b></p>		<p><b>Formative assessment</b> Knowledge checks Smart Revise Practice questions <b>Summative assessment</b> End of unit assessment</p>



<ul style="list-style-type: none"> <li>• Be able to identify the gate symbols for AND, OR and NOT gates</li> <li>• Be able to complete a truth table for each gate</li> <li>• Be able to produce a logic diagram</li> <li>• Be able to complete a truth table for a logic diagram</li> <li>• Be able to identify syntax errors, runtime errors and logic errors</li> <li>• Be able to identify Normal, Boundary and Erroneous test data</li> <li>• Be able to design a program anticipating misuse</li> <li>• Be able to design a program to include authentication</li> <li>• Be able to write a program that includes a subroutine</li> <li>• Be able to run a subroutine</li> <li>• Be able to write a program that includes a function</li> <li>• Be able to run a function</li> <li>• Be able to include arguments into subprograms</li> <li>• Be able to comment code in Python</li> </ul>		<p>Practice questions Revision tasks Research tasks</p>
--	---	---