



# A-Level COMPUTER SCIENCE Plans for Year 12 Curriculum

Term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
<b>Year 12</b>	<p><b><u>The characteristics of contemporary processors, input, output and storage devices</u></b> To introduce the concept of a <b>computer system</b> and its component parts and understand the link between <b>hardware</b> and <b>software</b>.</p>		<p><b><u>Exchanging data</u></b> To understand how data can be stored in <b>databases</b> and transmitted over computer networks. To be able to explain the process of data <b>compression</b> and the use of <b>protocols</b>.</p>	<p><b><u>Network topologies and layers</u></b> To understand how <b>data packets</b> are transmitted between electronic devices and how they are encoded to enable their delivery.</p>	<p><b><u>Progression exams</u></b> Component 01 and Component 02 style examinations.</p>	<p><b><u>Elements of computational thinking</u></b> To understand the principles and elements of computational thinking, and to be able to identify <b>abstraction, decomposition, logic</b> and <b>procedures</b> to solve problems.</p>
	<p><b><u>Software and software development</u></b> To develop programming skills in <b>Python</b>, Little Man Computer (<b>LMC</b>), <b>HTML</b>, <b>CSS</b>, <b>JavaScript</b> and <b>SQL</b> to be able to identify <b>sequence, selection</b> and <b>iteration</b> in code. To simulate the operations of a processor and assembly language, create interactive web pages of a uniformed style, and query a database over the duration of the course.</p>		<p><b><u>Data types, data structures and algorithms to solve problems and standard algorithms</u></b> To understand how data is represented within a computer and how <b>binary</b> can be interpreted as <b>numbers, text, sound</b> and <b>images</b>. To be able to convert binary to <b>hexadecimal</b> and understand why this can be used as shorthand. To develop and review <b>algorithms</b> to complete standard tasks.</p>			<p><b><u>Programming Project</u></b> To research and develop a programming project based on the analysis of a student lead project.</p>
Assessed through	<p><i>Programming exercises, home learning tasks based on examination questions and end of unit tests.</i></p>	<p><i>Programming exercises, home learning tasks based on examination questions and end of unit tests.</i></p>	<p><i>Programming exercises, home learning tasks based on examination questions and end of unit tests.</i></p>	<p><i>Programming exercises, home learning tasks based on examination questions and end of unit tests.</i></p>	<p><i>Formal progression examinations.</i></p>	<p><i>Research and identify suitable project ideas.</i></p>



## A-Level COMPUTER SCIENCE Plans for Year 13 Curriculum

Term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
<b>Year 13</b>	<p><b>Databases</b></p> <p>To develop and make the connection between database theory and their use as potential solutions to the <b>Programming Project</b>.</p> <p><b>Programming Project</b></p> <p>The students will develop a computing problem to work through according to the guidance in the specification based on the <b>analysis, design, development, testing</b> and <b>evaluation</b> of their solution.</p>	<p><b>Legal, moral, cultural and ethical issues</b></p> <p>To explore the <b>wider implications</b> of the use of computers in the modern world and identifying the <b>opportunities</b> and <b>risks</b> it brings. This would include <b>artificial intelligence, environmental effects, censorship, personal data</b>, and <b>piracy</b>. To understand and be able to apply relevant <b>computer legislation</b> to situations and explain how the principles of the law might be broken.</p>		<p><b>Revision</b></p> <p>To review and revise the entire examination specification.</p>	<p><b>Examinations</b></p>	
Assessed through	Completion of project work and home learning tasks.	Home learning tasks based on examination questions and revision tasks.	Formal Component 01 and Component 02 examination papers	Home learning tasks based on examination questions and classroom timed unit tests.		