

## Curriculum Subject: Computer Science KS4

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
<b>YEAR 10</b>	<b>Components of a Computer System</b> <ul style="list-style-type: none"> <li>• The purpose of the CPU</li> <li>• Von Neumann Architecture</li> <li>• CPU registers and components</li> <li>• Fetch-Decode-Execute</li> <li>• CPU Performance</li> <li>• Embedded systems</li> <li>• Memory</li> <li>• Storage</li> </ul>	<b>Networks</b> <ul style="list-style-type: none"> <li>• LANs &amp; WANs</li> <li>• Network performance</li> <li>• Client-server, P2P networks</li> <li>• Network Hardware</li> <li>• The internet</li> <li>• Virtual Networks</li> <li>• Network topologies</li> <li>• Wifi</li> <li>• Ethernet</li> <li>• Protocols</li> <li>• Layers</li> <li>• Packet switching</li> </ul>	<b>Programming</b> <ul style="list-style-type: none"> <li>• Variables, constants, operators, inputs, outputs and assignments</li> <li>• Programming constructs and flow control</li> <li>• String manipulation</li> <li>• File handling</li> </ul>	<b>Programming</b> <ul style="list-style-type: none"> <li>• Storing data in records</li> <li>• SQL</li> <li>• Arrays (1-D and 2-D)</li> <li>• Functions and procedures</li> <li>• Data types</li> <li>• Arithmetic operators</li> <li>• Boolean operators</li> </ul>	<b>Data Representation &amp; Computational logic</b> <ul style="list-style-type: none"> <li>• Storage units</li> <li>• Binary numbers</li> <li>• Hexadecimal numbers</li> <li>• Characters in binary</li> <li>• Images in binary</li> <li>• Digital sound</li> <li>• Compression</li> <li>• Boolean operators</li> <li>• Logic diagrams</li> <li>• Truth tables</li> <li>• Computing-related mathematics</li> </ul>	<b>Systems Software &amp; Security</b> <ul style="list-style-type: none"> <li>• Systems software</li> <li>• Operating systems</li> <li>• Utility software</li> <li>• Forms of attack</li> <li>• Threats to networks</li> <li>• Preventing vulnerabilities</li> </ul>
	Options Round 2					
<b>YEAR 11</b>	<b>Algorithms</b> <ul style="list-style-type: none"> <li>• Computational thinking</li> <li>• Searching algorithms</li> <li>• Sorting algorithms</li> <li>• Algorithm design</li> <li>• Working with algorithms</li> </ul>	<b>Programming Project</b> <ul style="list-style-type: none"> <li>• Programming techniques</li> <li>• Analysis</li> <li>• Design</li> </ul>	<b>Programming Project</b> <ul style="list-style-type: none"> <li>• Development</li> <li>• Testing, evaluation and conclusions</li> </ul>	<b>Design, Testing and IDEs</b> <ul style="list-style-type: none"> <li>• Defensive design</li> <li>• Maintainability</li> <li>• Purpose of testing</li> <li>• Types of testing</li> <li>• Syntax and logic errors</li> <li>• Selecting test data</li> <li>• Programming language levels</li> <li>• Translators</li> <li>• Assemblers, compilers and Interpreters</li> <li>• IDE tools</li> </ul>	<b>Legal, Ethical, Cultural and Environmental Issues</b> <ul style="list-style-type: none"> <li>• Computer science technologies</li> <li>• Effects on key stakeholders</li> <li>• Environmental impact</li> <li>• Cultural implications</li> <li>• Open source vs proprietary software</li> <li>• Legislation</li> </ul>	<b>External exams</b>

### St Bede's Curriculum Design Principles

Within subjects: depth, relevance, sequencing, spacing

Between subjects: breadth, cultural capital, coherence, progression, interlinking