



Topic Name	Term	Skills Developed	Next link in curriculum	Other Notes
<i>Computer Architecture: Structure / function of processor and processor types &amp; Input / output and storage devices</i>	<i>Autumn 1</i>	<ul style="list-style-type: none"><li>• CPU function/components/cycle.</li><li>• Von Neumann vs Harvard architecture.</li><li>• Processor cycle – Assembly language overview.</li><li>• CPU performance factors – pipelining, multicore etc.</li><li>• Parallel processing (SIMD/MIMD)</li><li>• RISC vs CISC processors.</li><li>• Categorising devices and selecting use for specific purposes.</li><li>• Flash/Optical/Magnetic storage media comparison.</li></ul>		<b>Links to Prior Learning:</b>  Y10 System Architecture Y10 Memory and Storage
<i>System Software: Operating Systems and utility software. Nature of applications</i>	<i>Autumn 2</i>	<ul style="list-style-type: none"><li>• Operating system purpose.</li><li>• Operating system functions – processor scheduling algorithms, memory management etc.</li><li>• Operating system types.</li><li>• Application generation – stages of compilation.</li></ul>		<b>Links to Prior Learning:</b>  Y11 – System Software



<i>Programming techniques/paradigms</i>	<i>Autumn 1 / Autumn 2</i>	<ul style="list-style-type: none"><li>• <i>Basic procedural language concepts (loops/lists/strings/files)</i></li><li>• <i>Procedures/functions/program flow</i></li><li>• <i>Variables/constants/scope</i></li><li>• <i>File handling / Use of IDE</i></li><li>• <i>OOP concepts, classes, methods, attributes</i></li><li>• <i>OOP - Inheritance and Polymorphism</i></li><li>• <i>Assembly language</i></li><li>• <i>Assembly addressing modes.</i></li><li>• <i>Declarative / functional programming.</i></li></ul>	<ul style="list-style-type: none"><li>• <i>NEA Programming Project (Year 12 – Summer 2)</i></li><li>• <i>Data structures and Algorithms – Spring 1/2</i></li></ul>	<b>Links to Prior Learning:</b> Y10 Programming Techniques Y11 Creating Robust Programs
<i>Networks and Web technologies</i>	<i>Spring 1 / Spring 2</i>	<ul style="list-style-type: none"><li>• <i>LAN / WAN / Hardware</i></li><li>• <i>CS/P2P/Ethernet/Wifi</i></li><li>• <i>Internet Structure and Protocols</i></li><li>• <i>Web Forms / JavaScript / php</i></li><li>• <i>Search Engine Indexing</i></li></ul>		<b>Links to Prior Learning:</b> Year 10 Network Communications and Protocols Year 11 Network Security
<i>Data Structures and Algorithms / Algorithm analysis and design</i>	<i>Spring 1/2</i>	<ul style="list-style-type: none"><li>• <i>Tuples and records</i></li><li>• <i>Linear, Circular and priority queues</i></li><li>• <i>Stacks vs Queues</i></li><li>• <i>Bubble / Insert / Merge / Quick Sorts</i></li></ul>	<ul style="list-style-type: none"><li>• <i>NEA Programming Project (Year 12)</i></li></ul>	<b>Links to Prior Learning:</b> Y11 Algorithms Y10 Programming Fundamentals



		<ul style="list-style-type: none"><li>• <i>Linear Search vs Binary Search (recursion)</i></li><li>• <i>Linked Lists</i></li><li>• <i>Hash Tables</i></li><li>• <i>Graphs - Implementation / Traversal</i></li><li>• <i>Trees - Implementation / Traversal</i></li><li>• <i>Trees - Binary Search Tree</i></li><li>• <i>Binary Search Tree using recursion</i></li><li>• <i>Breadth first vs Depth first graph traversal</i></li></ul>		
Algorithms	Summer 1	<ul style="list-style-type: none"><li>• <i>Searching Algorithms (Binary/Linear)</i></li><li>• <i>Sorting Algorithms (Bubble/Insert/Merge/Quick)</i></li><li>• <i>Optimisation Algorithms - Dijkstra's / A*</i></li><li>• <i>Big O Notation</i></li></ul>		<b>Links to Prior Learning:</b>  Y11 Algorithms Y10 Programming Fundamentals
System Lifecycle / Testing Methods	Summer 2	<ul style="list-style-type: none"><li>• <i>Waterfall / Spiral / Agile etc methods of software development.</i> <i>Testing strategies / test data / test plans.</i></li></ul>	NEA Programming Project (Year 12)	<b>Links to Prior Learning:</b>  Y11 – Creating Robust Programs (elements of testing)



<i>Computational Thinking</i>	<i>Summer 1</i>	<ul style="list-style-type: none"><li>• <i>Abstraction</i></li><li>• <i>Thinking Ahead</i></li><li>• <i>Thinking Procedurally</i></li><li>• <i>Thinking concurrently/Computational Methods</i></li></ul>	<ul style="list-style-type: none"><li>• <i>NEA Programming Project (Year 12)</i></li></ul>	<b>Links to Prior Learning:</b>  Y10 - Algorithms
<i>NEA Programming Project</i>	<i>Summer 2</i>	<ul style="list-style-type: none"><li>• <i>Independent NEA project.</i></li><li>• <i>Analyse, design, implement, test and evaluate a system developed for a real end user to fulfil a specific need.</i></li></ul>		<b>Links to Prior Learning:</b>  Y10 Algorithms Y12 Programming Techniques Y11 Creating Robust Programs Y12 Computational Thinking Y12 Data Structures and Algorithms