

Programme of study for Year 12 Computer Science

Autumn (1 st term) Topic	Autumn (2 nd term) Topic	Spring (1 st term) Topic	Spring (2 nd Term) Topic	Summer (1 st term) Topic	Summer (2 nd term) Topic
<p>Component 1 Unit 1: 1.1.3 Input, output and storage. 1.1.1 Structure and function of the processor.</p> <p>Component 2: 2.2.1 Programming techniques(Basics) Linear search.</p> <p>Skills: Developing accurate technical language to describe code and processes.</p>	<p>Component 1: 1.1.2 Types of processor 1.4.2 Data Structures (Arrays of up to 3 dimensions, records, lists, tuples). 1.5.1 Computing related legislation.</p> <p>Component 2: 2.1 Elements of computational thinking 2.2.1 Programming techniques (Improving) binary search.</p> <p>Skills: Research, coding, computational thinking</p>	<p>Component 1: 1.5.2 Moral and ethical Issues. 1.2.4 LMC and methods of addressing memory 1.4.2 Data Structures linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table.</p> <p>Component 2: 2.2.1 Programming techniques (intermediate) reading/ writing to & from files.</p> <p>Skills: Research and knowledge of current affairs. Logical thinking</p>	<p>Component 1: 1.2.4 Types of Programming Language & OOP Theory 1.3.1 Compression, Encryption and Hashing</p> <p>Component 2: 2.2.1 Programming techniques(advanced) recursion (OOP)</p> <p>Skills: Study skills</p>	<p>Component 1: 1.3.2 Databases 1.2.3 Software Development (methodology.)</p> <p>Component 2: 2.2.1 Programming techniques(advanced) SQL and GUI</p> <p>Skills: Research and responsibility for self-study</p>	<p>Component 1: 1.2.3 Software Development (Testing)</p> <p>Component 2: 2.2.1 Programming techniques(advanced) 2.2.2 Computational methods (applying to NEA)</p> <p>Component 3: (NEA) Project analysis Project Design</p> <p>Skills: Research Computational thinking</p>
End of term 1 evidence to cover: Programming Skills. CPU knowledge		End of term 2 evidence to cover: Data Structures, knowledge of laws		End of year evidence to cover: Understand data transmission GUI and DB coding	
Rationale for sequence: Gentle introduction to both theory and programming.	Rationale for sequence: When programming there are legal responsibilities which are addressed at the same time as programming techniques are developed. Data Structures used in	Rationale for sequence: The wider impact of IT systems are considered and more complex data structures that could be used in solving more challenging problems. Building up on knowledge of processors instructions	Rationale for sequence: Starting to build up to the knowledge required for using OOP in the NEA. This allows us to also look at how OOP is used in implementing some data structures.	Rationale for sequence: Preparation for the NEA project - Understanding how to store persistent data and develop coding skills to link together theory and practical programming techniques.	Rationale for sequence: Students should be more aware of their coding ability, project requirements and choose a project of interest to them that will give them the opportunity to

	programming are introduced.	through assembly language.			produce a challenging yet feasible project.
Home – Learning: Related to topic covered. Programming.	Home – Learning: Related to topic covered. Programming.	Home – Learning: Related to topic covered. Programming.	Home – Learning: Related to topic covered. Programming.	Home – Learning: Related to topic covered. Programming.	Home – Learning: Related to topic covered. Programming.
Reading / High Quality Text: Technical language using manuals. Tech news Articles.	Reading / High Quality Text: Technical language using manuals. Tech news Articles.	Reading / High Quality Text: Technical language using manuals. Tech news Articles.	Reading / High Quality Text: Technical language using manuals. Tech news Articles.	Reading / High Quality Text: Technical language using manuals. Tech news Articles.	Reading / High Quality Text: Technical language using manuals. Tech news Articles.
Numeracy: Storage calculations, coding.	Numeracy: Coding calculations	Numeracy: Equations in coding	Numeracy: Mental maths – hashing techniques.	Numeracy: Equations in coding.	Numeracy: Coding calculations
Enrichment / opportunities to develop cultural capital (including careers, WRL and SMSC): Encourage students to mentor non coders, take advantage of online webinars and tutorials. Consider taking a MOOC to help with UCAS statements.					