Programme of study for Year 12 Computer Science

Autumn (1 st term) Topic	Autumn (2 nd term) Topic	Spring (1 st term)	Spring (2 nd Term)	Summer (1 st term)	Summer (2 nd term) Topic
		Торіс	Торіс	Торіс	
Component 1 Unit 1:	Component 1:	Component 1:	Component 1:	Component 1:	Component 1:
 1.1.3 Input, output and storage. 1.1.1 Structure and function of the processor. Component 2: 2.2.1 Programming techniques(Basics) Linear search. Skills: Developing accurate technical language to describe code and processes. 	 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. Component 2: 2.2.2 Computational methods. 2.2.1 Programming techniques(Improving) binary search. Skills: Research, coding, compuational thinking 	 1.5.2 Moral and ethical Issues. 1.4.2 Data Structures linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table. Component 2: 2.2.1 Programming techniques (intermediate) reading/ writing to & from files. Skills: Research and knowledge of current affairs. Logical thinking 	 1.2.4 Types of Programming Language (OOP) 1.3.1 Compression, Encryption and Hashing Component 2: 2.2.1 Programming techniques(advanced) recursion (OOP) Skills: Study skills 	 1.3.2 Databases 1.2.3 Software Development. Component 2: 2.2.1 Programming techniques(advanced) SQL and GUI Skills: Research and responsibility for self study 	 1.2.3 Software Development. Component 2: 2.2.1 Programming techniques(advanced) 2.2.2 Computational methods (applying to NEA) Component 3: (NEA) Project analysis Project Design Skills: Research Computational thinking
End of term 1 evidence to cover:		End of term 2 evidence to cover:		End of year evidence to cover:	
Programming Skills. CPU knowledge		Data Structures, knowledge of laws		Understand data transmission GUI and DB coding	
Rationale for sequence:	Rationale for sequence:	Rationale for sequence:	Rationale for sequence:	Rationale for sequence:	Rationale for sequence:

Gentle introduction to both theory and programming.	When programming there are legal responsibilities which are addressed at the same time as programming techniques are developed. Data Structures used in programming are introduced.	The wider impact of IT systems are considered and more complex data structures that could be used in solving more challenging problems	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.	Preparation for the NEA project - Understanding how to store persistent data and develop coding skills to link together theory and practical programming techniques.	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 produce a challenging yet feasible project.				
Home – Learning:	Home – Learning:	Home – Learning:	Home – Learning:	Home – Learning:	Home – Learning:				
Related to topic covered. Programming.	Related to topic covered. Programming.	Related to topic covered. Programming.	Related to topic covered. Programming.	Related to topic covered. Programming.	Related to topic covered. Programming.				
Reading / High Quality Text:	Reading / High Quality Text:	Reading / High Quality Text:	Reading / High Quality Text:	Reading / High Quality Text:	Reading / High Quality Text:				
Technical language using manuals. Tech news Articles.	Technical language using manuals. Tech news Articles.	Technical language using manuals. Tech news Articles.	Technical language using manuals. Tech news Articles.	Technical language using manuals. Tech news Articles.	Technical language using manuals. Tech news Articles.				
Numeracy:	Numeracy:	Numeracy:	Numeracy:	Numeracy:	Numeracy:				
Storage calculations, coding.	Coding calculations	Equations in coding	Mental maths – hashing techniques.	Equations in coding.	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.									