Curriculum Map Year 11 GCSE COMPUTER SCIENCE

Topic Name	Term	Skills developed with link to NC Subject content	Reflection on previous link in the curriculum	Progress to future link in the curriculum
Algorithms	*Autumn HT1 and HT2*	- Be able to define the term algorithm. - Understand what is meant by abstraction. - Be able to provide examples of abstraction. - Know what is meant by the term 'abstraction'. - Be able to provide examples of abstraction. - Know what is meant by problem decomposition. - Know the advantages of decomposition when applied to programming. - Be able to provide an example of problem decomposition. - Understand how to solve computational problems by applying a systematic approach to problem solving. - Know the flow diagram symbols. - Know that flow diagrams are also called flowcharts. - Know how to make a flow diagram. - Understand how to construct a program from a flow diagram. - Know what is meant by the term pseudocode. - Understand how to write pseudocode. - Know how to make a flow diagram. - Understand how to write pseudocode. - Understand how to write a program from a flow diagram and pseudocode. - Know how to make a flow diagram. - Understand how to write pseudocode. - Understand how to write a program from a flow diagram and pseudocode. - Understand how to interpret algorithms. - Understand how to correct algorithms. - Be able to identity inputs, processing and outputs from an algorithm. - Understand how to use visual inspection to determine how simple algorithms work and what their purpose is. - Understand how to use trace tables to determine how simple algorithms work and what their purpose is. - Understand the linear search algorithm. - Understand it is not an efficient algorithm, but it is easier to program than alternatives and does not require the items to be in any order. - Understand the binary search algorithm. - Understand the required condition of the list of items in order for the binary search to work. - Understand which searching algorithm is quicker. - Understand the merge sort algorithm. - Understand the bubble sort algorithm. - Understand that more than one algorithm can be used to solve the same problem. - Understand that certain algorithms can be more efficient (quicker) at solving a problem than another.	*Year 10: Python programming* *Year 10: Basic programming concepts* *Year 10: Advanced programming concepts* *Year 10: Robust and secure programming*	*Year 11: Python programming examination work* *Year 11: Classification of programming languages* *Examination revision* *A Level: Algorithms and programming* *A Level: NEA* *A Level: Computational thinking*

Boolean logic	*Autumn HT2*	• Understand how to construct truth tables from logic gates NOT AND OR XOR. • Understand how to construct truth tables for simple logic circuits. • Be able to interpret the results of simple truth tables. • Know how to make simple logic circuit diagrams from Boolean expressions using AND, OR, NOT, XOR and vice versa. • Know how to modify and interpret simple logic circuit diagrams. • Understand how to create, complete or edit logic diagrams and truth tables for given scenarios.	*Year 8: Understanding computers* *Year 10: Data representation* *Year 10: Python programming* *Year 10: Basic programming concepts*	*Year 11: Python programming examination work* *Year 11: Relational databases and SQL* *Examination revision* *A Level: Boolean algebra*
Python programming: examination work	*Autumn HT2 and HT3*	• Past paper questions and challenges related to Basic and advanced programming concepts	*Year 10: Basic programming concepts* *Year 10: Advanced programming concepts* *Year 10: Robust and secure programming concepts*	
Classification of programming languages	*Spring HT3 and HT4*	• Know the characteristics of low-level and high-level programming languages. • Explain the main differences between low-level and high-level languages. • Understand the advantages and disadvantages of low-level language programming compared with high-level language programming. • Understand the relationship between machine code and low-level languages. • Explain the difference between machine code and low-level languages. • Know what a translator does. • Understand the differences between assemblers, compilers and interpreters. • Understand when it would be appropriate to use each type of translator.	*Year 10: Systems architecture* *Year 10: Memory and storage* *Year 10: Hardware and software* *Year 10: Data representation* *Year 10: Basic programming concepts* *Year 10: Advanced programming concepts* *Year 10: Robust and secure programming concepts*	*Year 11: Python programming examination work* *Examination revision* *A Level: programming languages, assembly language and object orientated programming*
Python programming: examination work	*Spring HT4 and Summer HT5*	• Past paper questions and challenges related to advanced and robust and secure programming concepts	*Year 10: Basic programming concepts* *Year 10: Advanced programming concepts* *Year 10: Robust and secure programming concepts*	*Examination revision* *A Level programming project*

Relational databases and SQL	*Spring HT4 and Summer HT5*	• Understand what a database is.
		• Understand what a relational database is.
		• Understand the difference between the two.
		• Understand the following database concepts:
		   • Table
		   • Record
		   • Field
		   • Primary key
		   • Foreign key
		• Understand that the use of a relational database facilitates the elimination of data inconsistency and data redundancy.
		• Understand how to retrieve data from a relational database by using the SQL command SELECT.
		• Understand how to order the data retrieved from a SQL SELECT command.
		• Understand how to insert data into a relational database by using the SQL INSERT command.
		• Understand how to update data in a relational database by using the SQL UPDATE command.
		• Understand how to delete data from a relational database by using the SQL DELETE command.
			Year 10: Advanced programming concepts
			Year 10: Robust and secure programming concepts
				A Level: Relational databases and SQL
				Examination revision