Year 10	**Autumn Term**		**Spring Term**		**Summer Term**	
Topic(s)	**Topic 3 - Computers**	**Python Programming**	**Topic 1 – Computational Thinking**	**Python Programming**	**Topic 2 - Data**	**Python Programming**
Key Learning Content	• Systems architecture • Busses and the fetch decode execute cycle • Secondary storage • Operating systems • Utility software • Identifying vulnerabilities • Programming languages	• Comments • Identifiers • Data types and conversion • Operators • Programming constructs ○ Assignment ○ Sequencing ○ Blocking • Able to write programs that make appropriate use of primitive data types (integer, real, Boolean, char) and one and two-dimensional structured data types (string, array, record)	• Decomposition and abstraction • Algorithms ○ Searching ○ Sorting • Flowcharts • Pseudocode • Trace tables • Truth tables	• Programming constructs including sequencing and selection and iteration • Be able to write programs that make appropriate use of sequencing, selection, repetition (count-controlled, condition-controlled), iteration (over every item in a data structure) and single	• Storage units • Binary numbers • Binary arithmetic • Hexadecimal • Binary shits • Twos complement • ASCII • Images • Sound • Compression	• Able to write programs that use pre-existing (built-in, library) and user-devised subprograms (procedures, functions) • Able to write functions that may or may not take parameters but must return values, and procedures that may or may not take parameters but do not return values

		- Able to write programs that accept and respond appropriately to user input		entry/exit points from code blocks and subprograms		- Understand the difference between and be able to write programs that make appropriate use of global and local variables
Specification Reference	3.1 Hardware 3.2 Software 3.3 Programming languages	6.1 Develop code 6.2 Constructs 6.3 Data types and structures 6.4 Input/output 6.5 Operators	1.1 Decomposition and abstraction 1.2 Algorithms 1.3 Truth tables	6.1 Develop code 6.2 Constructs 6.3 Data types and structures 6.4 Input/output 6.5 Operators	2.1 Binary 2.2 Data representation 2.3 Data storage and compression	6.6 Subprograms
Assessment	Written assessment 50 marks total		Written assessment 50 marks total		Written assessment 50 marks total	

Year 11	**Autumn Term**		**Spring Term**		**Summer Term**
Topic(s)	**Topic 4 - Networks**	**Python Programming**	**Topic 5 – Issues and Impact**	**Python Programming**	**Revision**
Key Learning Content	• LANS & WANS • The Internet • Wired and Wireless • Protocols and layers • Network topologies • Network security	• Comments • Identifiers • Data types and conversion • Operators • Programming constructs ○ Assignment ○ Sequencing ○ Blocking • Able to write programs that make appropriate use of primitive data types (integer, real, Boolean, char) and one and two-dimensional structured data types (string, array, record) • Able to write programs that accept and	• Environmental issues • Ethical issues • Legislation and privacy • Cyber security	• Programming constructs including sequencing and selection and iteration • Be able to write programs that make appropriate use of sequencing, selection, repetition (count-controlled, condition-controlled), iteration (over every item in a data structure) and single entry/exit points from code blocks and subprograms • Turtle graphics library module • Turtle window and drawing canvas • Turtle creation, visibility and movement	

		respond appropriately to user input		• Turtle positioning and direction • Turtle filling shapes • Turtle controlling the pen • Turtle circles • Turtle colours	
Specification Reference	4.1 Networks 4.2 Network Security	6.1 Develop code 6.2 Constructs 6.3 Data types and structures 6.4 Input/output 6.5 Operators	5.1 Environmental 5.2 Ethical and legal 5.3 Cybersecurity	6.1 Develop code 6.2 Constructs 6.3 Data types and structures 6.4 Input/output 6.5 Operators	
Assessment	Written assessment 40 marks total		Written assessment 30 marks total		