Timeline	Topic	Key concepts and knowledge	Skills development	Rationale
Autumn 1 7 lessons	2.1 – Algorithms 2.2 – Programming Fundamentals 2.1.2 – Designing, creating and refining algorithms 2.2.1 – Programming Fundamentals	Students know: • the importance of using 'problem solving' techniques when solving problems. • the meaning of 'Abstraction', 'Decomposition' and 'Algorithmic Thinking'. • that most modern programming languages have 3 main constructs (sequencing, selection and iteration). • the differences between sequencing, selection and iteration. • the purpose of algorithms • how to create and read flowcharts • how to create and read pseudocode • the main programming constructs of sequencing, selection and iteration • the main uses of arithmetic operators Students know how to: • explain what 'Abstraction', 'Decomposition' and 'Algorithmic Thinking' is. • list and describe the 3 main programming constructs. • identify programming constructs in a given set of code. • discuss the suitability of various construct implementations (e.g. FOR vs WHILE) for given scenarios • apply these techniques when solving a range of problems • explain the purpose of an algorithm and to be able to read / create simple algorithms in the form of flowcharts and pseudocode. • discuss the benefits and drawbacks of the various methods of presenting algorithms. • read / create algorithms for more complex problems • program sequences of code using selection and iteration	**Subject-specific skills:** Python programming skills and computational thinking skills Exam technique **Employability skills:** Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall **Career links:** Computer Programmer	Success in the J277/02 element of GCSE assessment depends upon a core understanding and application of programming skills. The introduction to the GCSE commences with knowledge and skill development centring around algorithmic thinking and programming fundamentals. This unit builds from KS3 learning of computational thinking skills and Python programming techniques. This learning will then allow learners to continue to develop their understanding of algorithms, and programming skills to develop more complex programs late in the course, including using file handling and sub-programs

Autumn 2 7 lessons	2.2 Programming fundamentals 2.4 Boolean logic 1.2 Systems Architecture 2.2.2 Data types 2.2.3 Additional programming techniques 2.4.1 Boolean Logic 1.2.3 Units of storage 1.2.4 Data storage	Students know: • the various types of operators that are used to perform an action on data. • the use and results of various built in string manipulation methods. • what arrays are. • how arrays can be multi-dimensional. • the difference between static and dynamic data structures. • how programs can be written to read from and write to files. • the role of the file handler • how NOT, AND & OR gates process their inputs. • how to work out the output of a logic circuit for a given set of inputs. • how to draw logic circuits for given expressions • why computers use the binary number system. • how the binary number system works. • the various binary units • how the hexadecimal number system works. • how to convert between decimal, hexadecimal and binary. • why the hexadecimal number system is used in computer science. Students know how to: • explain the effects of various operators used in programs and the effects of various string manipulation methods for given strings. • select the appropriate operators / methods for given scenarios. • explain what an array is and how data can be accessed from within an array. • read an array with more than one dimension. • describe the process of coding programs to read from and write to external files. • read and write simple pseudocode algorithms to read from and write to files. • complete truth tables for each logic gate and various logic circuits and to draw logic circuit diagrams from simple logic expressions. • draw logic circuits from more complex expressions. • describe how the binary number system works and be able to describe the size of standard binary units. • explain why computer systems use the binary number system • choose appropriate units to describe the size of various types of file (document, audio etc). • describe how the hexadecimal number system works and convert simple values between denary and hexadecimal and binary. • convert more complex values. • explain the importance of hexadecimal in computer science.	**Subject-specific skills:** Python programming skills and computational thinking skills Exam technique **Employability skills:** Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall **Career links:** Computer Programmer, Electrical engineer, physicist	Building from knowledge and skills in Autumn 1, students continue to explore programming fundamentals and link learning of logic gates from year 9 learning scheme and from binary knowledge developed from years 7 and 8. By understanding how computers store data and represent numbers as well as using logic expressions, will allow learners to move into exploring how images, characters and sound are represented in the Spring 2 term.

Spring 1 6 lessons	2.2 Programming fundamentals 2.2.3 Additional Programming Techniques Practical programming skills (7 hours)	Students know: • the need to break down programs into smaller sections • the difference between a procedure and a function. • the difference between a local variable and a global variable. • what parameter passing is and how it works. • the purpose of SQL. • database terminology. • what is meant by the term 'query'. • how to read and write basic SQL syntax in order to create tables, add data and query databases. • How to apply basic and additional programming techniques to solve a variety of real world problems Students know how to: • explain the benefits that subroutines bring to programming and to be able to explain the difference between procedures and functions. • explain the difference between local and global variables and explain how parameter passing works. • demonstrate a thorough knowledge of parameter passing and returning values through written algorithm • define basic database terminology, describe the purpose of SQL and read basic SQL statements. • write SQL statements for a range of scenarios. • Use a range of programming techniques to plan and develop programs that solve real world problems	**Subject-specific skills:** Python programming skills and computational thinking skills Exam technique **Employability skills:** Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall **Career links:** Data handler	Drawing and building on knowledge and skills acquired in the first term, learners move into more complex programming techniques which should reinforce prior learning and skills whilst allowing opportunity for more complex programs to be developed using external files and sub programs. Learners have an opportunity to embed this learning in a period of practical programming skills which will underpin the component 2 learning moving into the more theoretical aspects of learning in component 1

Spring 2 6 lessons	1.1 – System Architecture 1.2 – Memory and storage 1.2.4 – Data storage (characters, images, sound) 1.2.5 – compression 1.1.1 – Architecture of the CPU 1.1.2 – CPU performance 1.1.3 – Embedded Systems	Students know: • how binary numbers can represent characters. • what the ASCII character set is. • the limitations with the ASCII character set. • the benefits that Unicode offers. • how images are represented in computer systems. • what is meant by ‘colour depth’. • how ‘colour depth’ affects an image’s quality and file size. • what is meant by ‘image resolution’. • how ‘image resolution’ affects an image’s quality and file size. • the purpose of meta data in image files. • how to calculate the file size of an image if its resolution and colour depth are known. • how sound is represented in computer systems. • the process of sampling. • is meant by ‘bit rate’. • how ‘bit rate’ affects a sound file’s quality and size. • what is meant by ‘sample rate’. • how ‘sample rate’ affects a sound file’s quality and size. • the purpose of meta data in sound files. • how to calculate the file size of a sound file if its ‘bit rate’ and ‘sample rate’ • the purpose of file compression. • the difference between lossy and lossless compression. • practical examples of both lossy and lossless compression (i.e.: ‘Run Length Encoding’, ‘Dictionary Coding’ and ‘Huffman Coding’). • the various characteristics of a CPU and how they each affect the performance of the CPU • the purpose of the CPU • how the CPU processes data • the various components that make up the CPU • the importance of the Von Neumann architecture in separating the program from the machine. • what is meant by an ‘Instruction Set’ • the relative roles of specialised CPU registers during the Fetch-Decode-Execute cycle • Students know how to: • describe how characters are represented in binary and to be able to explain what the ASCII character set is. • discuss the limitations of ASCII and the benefits of Unicode.	**Subject-specific skills:** computational thinking skills Exam technique **Employability skills:** Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall **Career links:** Data handler Sound engineer Media designer	Students are introduced to some of the more abstract concepts of computing by exploring the components of computing hardware and understanding their role in processing data. Initially students build on the Spring 1 exploration of data representation of numbers to now look at characters, images and sound, again building on concepts introduced back in year 9. Once students understand more about how computers represent different types of data, they then move to explore the CPU components and processes, including the FDE cycle. Once this unit has been completed learners can then move to exploring both primary and secondary storage in the next half term.

		• explain how images are represented in binary, explain colour depth and resolution and their effects on file size and image quality. • calculate the file size of an image from an image’s colour depth and resolution. • explain the purpose of meta data in image files. • explain how sounds are represented in binary, explain bit rate and sample rate and their effects on file size and sound quality. • calculate the file size of a sound file from the file’s bit rate and sample rate. • explain the purpose of meta data in sound files. • explain the reasons for compression and explain the difference between lossy and lossless compression. • explain ‘Run Length Encoding’ & ‘Dictionary Coding’. • explain and carry out ‘Huffman Coding’. • explain the various characteristics of a CPU • explain how these characteristics can affect the performance of a CPU • begin to discuss the current limitations of processor speeds. • explain the fetch-decode-execute cycle. • explain the fetch-decode-execute cycle in relation to the various components of the CPU • discuss the performance of the CPU in relation to the speed of the F-D-E cycle • describe the roles of various specialised registers inside the CPU during the F-D-E cycle. • explain what the registers do during F-D-E cycles when provided with a given set of instructions. • discuss the benefits that the Von Neumann architecture provided to computing systems.		

Summer 1 5 lessons	1.2 – Memory & Storage 1.3 – Computer networks, connections and protocols 1.2.1 – Primary storage 1.2.2 – Secondary storage 1.3.1 – Networks and topologies	Students know: • the characteristics and roles of RAM, ROM, Cache and Virtual Memory. • the difference between volatile and non-volatile memory • the need for secondary storage. • the various secondary storage technologies. • the characteristics of various secondary storage devices. • the difference between LANs and WANs • the difference between a peer-to-peer and client server network • how different data transfer mediums carry data. • how data is transmitted across a network. • The differences between a hub and a switch • The hardware needed to connect to a LAN Students know how to: • explain the roles of RAM, ROM, Cache and Virtual Memory and describe the contexts where each are required. • discuss the benefits and drawbacks of Virtual Memory. • discuss the effects of varying amounts of cache on the performance of a computer system. • explain why computer systems require secondary storage and be able to state the characteristics of various secondary storage devices. • explain how the various secondary storage technologies work • discuss the suitability of various secondary storage devices in a variety of contexts. • Explain the differences between LANs and WAN and how they communicate with devices • Explain the differences between peer-to-peer and client server networks • explain how packet switching works. • discuss the advantages and disadvantages of different data transfer mediums • explain the role of different hardware needed to connect to a network and its role in transferring data • identify the most appropriate hardware needed for a specific scenario, and justify their choice	**Subject-specific skills:** computational thinking skills Exam technique **Employability skills:** Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall **Career links:** Network Manager	The summer term learning business on students understanding of the CPU and its role, to move on and explore how information is stored and the need for primary and secondary storage devices. This will then result in learners having an understanding of individual computers and their role in storing and processing data. This leads into learning how computers communicate with each other across networks. Learning in this half term will then allow learners to move into their final half term to then look at the rules governing the transfer of data across networks

Summer 2 6 lessons	1.3 – Computer networks, connections and protocols 1.3.2 – Wired and wireless networks, protocols and layers Practical programming skills – (approx. 3 hours)	Students know: • what is meant by the term ‘protocol’. • the purpose of a variety of common network protocols • what the term ‘layer’ means • the difference between a protocol and a standard • the benefits of using layers on a network Students know how to: • describe some common protocols. • explain the purpose of common protocols • explain the layers of a network and their purpose • Use a range of programming techniques to plan and develop programs that solve real world problems	**Subject-specific skills:** computational thinking skills Exam technique **Employability skills:** Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall **Career links:** Network and data manager	This final unit of work builds on concepts introduced in the network unit in year 9 by exploring the protocols and layers governing the transfer of data across a network. Some of these concepts are quite abstract and require learners to have an understanding of how computers communicate across a network before exploring the protocols. This final unit in year 10 will fed directly into learners exploring threats to networks at the start of their year 11 learning.