Year	Unit of work	Rational	Key skills
7	Introduction to computer science – E-safety, cyber security and file management.	Pupils in year 7 are initially taught in forms and moved into sets after the first half term. Pupils arrive with a wide range of experience in computer science, from none to extensive. This initial unit of work gives us an overview of their understanding and tackles important preparatory elements such as, e-safety, cyber security and file management.	• use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.
	Skills in computer science – Problem Solving (careers in computer science)	It is important to discuss careers within the curriculum and one of the main tasks that underpins computer science is problem solving. In school: Computers need specific and detailed instructions to follow to successfully complete a task. That's essentially what a computer program is. Pupils will learn how to write programs, taking those instructions and putting them into a language a computer can understand. Not everything will work first time, so they'll have to think around the problem to solve it, often using trial and error techniques. This can be taught at any point within the curriculum once pupils have been placed in sets.	• analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
	Computational thinking – Creating and testing algorithms	Selection, sequencing and iteration are key aspects of computer science as well as resilience and problem solving. This unit is an excellent way	• understand what algorithms are; how they are implemented as programs on digital devices;

		to introduce pupils to those concept and creates a sound base for further units.	and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	System architecture – Primary hardware components and their uses.	This unit of work deals with the hardware aspect of computer science, were pupils will learn about the main components that computer systems contain. It is important that pupils have an understanding of the components before	• recognise common uses of hardware components. • Understand how computers process data and produce information.

		learning about software and many of them are referred to in the next unit.	• Understand what data is and how analogue data if transferred to digital data.
	System software – OS and application software	This unit of work flows on logically from system architecture. Hardware and software are the main components that make computer system work.	• use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school. • Understand the fundamental need for software and its different uses.
	Wireless and wired networks - Topologies	This unit of work cannot be taught without a sound understanding of the two previous units.	• understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration.
	System security – Threats including malware	Once pupils have an understanding of how computer network function, we believe that it is important for them to study the security issues that arise from digital content and how to prevent issues. This is the logical unit to follow computer networks.	• understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration. • recognise common uses of information technology beyond school. • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a

			range of ways to report concerns about content and contact.
	Ethical, legal and cultural issues – Environmental issues and the digital divide	This unit could be taught at any point of the curriculum. However, we feel a sound understanding of four previous units is needed to fully understand this unit.	• use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school. • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact. • can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.
	Skills in computer science – Mathematical equations (careers in computer science)	It is important to discuss careers within the curriculum. When working with algorithms, mathematical equations are very important. In school: Mathematical principles are essential to computer programming. Pupils will practice binary and hexadecimal conversions and calculations to simulate how a computer processes instructions. Computer science also	• use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.

		requires practical skills including evaluating Boolean algebra, drawing logic gates and analysing mathematical functions used to represent algorithmic efficiency. This can be taught at any point within the curriculum once pupils have been placed in sets.	
	Algorithms	This unit could potentially be studied after computational thinking as the concept are very closely linked. However, we believe that too much of the same thing might over face pupils. Therefore, we study this in the summer term.	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output.

			• use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Programming techniques – Visual programming	See above	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.

	Data representation –binary	This is a discreet unit of study that could potentially be taught at any point of the curriculum. It actually appears and is taught in part throughout the other units of work. This can be taught at any point within the curriculum once placed in sets.	• understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation.

If there is a class/s that are shared between 2 teachers, the curriculum will follow a slightly different style. If there is a split with the teachers, the units of work will be shared and taught individually. The units of work which can be taught on their own (E-safety, ethics, careers in computer science, data representation) will be taught by one teacher whilst the other follows the SOW missing out these units. Once these units have been completed pupils will have covered systems architecture, software and wired and wireless networks so will have some knowledge on how computers work. Here, algorithms and programming techniques will be taught. There will always be overlap with topics as concepts in computer science do.

Year	Unit of work	Rational	Key skills
8	E-safety – Staying safe online	This unit could be studied at any point of the curriculum. However, we believe that the importance of the subject matter needs to be projected as early as possible in the year as the pupils may have had experiences of online issues over the summer holidays. This allows us to deal with any potential issues. The theme of this unit runs throughout the curriculum and crops up regularly in other units such as cyber security. This can be taught at any point within the curriculum.	• use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.
	Computational thinking – Using algorithms to represent control systems.	In year 8 we reinforce the key aspects of selection, sequencing and iteration as well as resilience and problem solving as early as possible.	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling

			or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	System architecture – How components work and integrate	This unit of work enhances the pupils' knowledge of the hardware aspect of computer science and pupil learn about the main components that computer systems contain in more depth. It is important that pupils have an understanding of the components before learning about software and many of them are referred to in the next unit.	• recognise common uses of hardware components. • Understand how computers process data and produce information. • Understand what data is and how analogue data if transferred to digital data.
	System software	This unit of work flows on logically from system architecture. Hardware and software are the main components that make computer system work. Pupils learn about the functions of OS and the difference between application software and system utilities.	• use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school. • Understand the fundamental need for software and its different uses.
	Wireless and wired networks – Star/mesh topologies and LAN/ WAN	This unit of work cannot be taught without a sound understanding of the two previous units.	• understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the

		In this unit we introduce different types of topologies and dive deeper into LAN and WAN.	opportunities they offer for communication and collaboration.
	System security – Network attacks	Once pupils have an understanding of how computer network function and the features of the different topologies we believe that it is important for them to study the security issues that arise from digital content and how to prevent issues in greater depth. This is the logical unit to follow computer networks.	• understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration. • recognise common uses of information technology beyond school. • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.
	Ethical, legal and cultural issues -The digital divide	This unit could be taught at any point of the curriculum. However, we feel a sound understanding systems architecture and software systems is needed to full understand this unit as technical terms are used.	• use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school. • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

			• can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.
	Skills in computer science – Data analysis (careers in computer science)	It is important to discuss careers within the curriculum. When working with algorithms understanding how the data is stored is important as pupils need to know that computers do not "think" and have to be told the exact information and how data will be stored. In school: We create a vast amount of digital data. Capturing and storing this data is an essential part of computer science. Pupils will explore the abstract concepts of attributes, records and files, as well as learn about specific data types and structures, which are used to represent information, and how to apply them in different scenarios. This can be taught at any point within the curriculum.	• Understand how computers process data and produce information. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation.
	Algorithms – Flow diagrams and pseudo code.	This unit could potentially be studied after computational thinking as the concept are very closely linked. However, there are more practical elements to this unit, such as text based programming, which we find more appealing to teach in the spring term as much of the previous content is theory.	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science,

			including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Programming techniques – Text based programming languages.	See above	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing

			computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Data representation –characters, binary and logic.	This is a discreet unit of study that could potentially be taught at any point of the curriculum. It actually appears and is taught in part throughout the other units of work. This can be taught at any point within the curriculum.	• understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • recognise common uses of information technology beyond school.
	Skills in computer science – Creativity (careers in computer science)	It is important to discuss careers within the curriculum. Creativity is an important skill to have within computer science as it is needed to produce new programs and to create new ways to solve problems. In school: Creativity and problem solving go hand in hand. Sometimes pupils will need to think creatively when they're writing a computer program because it may not be obvious how to	• are responsible, competent, confident and creative users of information and communication technology; • select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting,

		solve a particular problem. Pupils might also have creative ideas for the development of an app or computer game. This can be taught at any point within the curriculum.	analysing, evaluating and presenting data and information.

If there is a class/s that are shared between 2 teachers, the curriculum will follow a slightly different style. If there is a split with the teachers, the units of work will be shared and taught individually. The units of work which can be taught on their own (E-safety, ethics, careers in computer science, data representation) will be taught by one teacher whilst the other follows the SOW missing out these units. Once these units have been completed pupils will have covered systems architecture, software and wired and wireless networks so will have some knowledge on how computers work. Here, algorithms and programming techniques will be taught. There will always be overlap with topics as concepts in computer science do.

Year	Unit of work	Rational	Key skills
9	E-safety – Digital copyright and legislation	This unit could be studied at any point of the curriculum. However, we believe that year 9 pupils should become more aware of the issues surrounding copyright and the legislation that governs it including illegal streaming and plagiarism.	• use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.
	Skills in computer science – Logical thinking (careers in computer science)	It is important to discuss careers within the curriculum. Logical thinking is a key skill that is needed within computer science. In school: Computers rely on logic to run and you need to learn how to think 'algorithmically'. Pupils have to make sure their programs are logically sequenced when they write them and know how to avoid common errors through data validation and verification techniques. Pupil also learn to approach any programming problems from a logical perspective to develop a working solution.	• understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	System architecture – memory, secondary storage, Von Neumann architecture.	This unit of work looks at systems architecture in more depth than previously, including GCSE topics such a primary/secondary storage and Von Neumann architecture. Many of these components are referred to in the subsequent units, therefore it is important that pupils have a sound knowledge on how they work.	• recognise common uses of hardware components. • Understand how computers process data and produce information. • Understand what data is and how analogue data if transferred to digital data.

	System software – operating systems, utility/application software and its uses.	This unit of work flows on logically from system architecture. Hardware and software are the main components that make computer system work. Pupils in greater depth about the functions of OS and how application software and system utilities are used.	• use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school. • Understand the fundamental need for software and its different uses.
	Wireless and wired networks – Hardware components	This unit of work cannot be taught without a sound understanding of the two previous units. In this unit pupils study network topologies and LAN/WAN in greater depth.	• understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration.
	System security – Network attacks and malware	This is the logical unit to follow computer networks. Once pupils have a greater understanding of how computer network function and the features of the different topologies we believe that it is important for them to study in more detail about the security issues that arise from digital content and how to prevent issues.	• understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration. • recognise common uses of information technology beyond school. • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

	Ethical, legal and cultural issues - Impact of digital technology	This unit could be taught at any point of the curriculum. However, we feel a sound understanding systems architecture and software systems is needed to full understand this unit as technical terms are used.	• use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school. • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact. • can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.
	Algorithms – Searching and sorting algorithms	This unit could potentially be studied after computational thinking as the concept are very closely linked. However, it is important to complete this unit before the programming unit as algorithms are used as a basis for robust programs.	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems.

			• design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Programming techniques – Text based programming languages.	See above	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

			• use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Data representation – Images, characters, audio, binary and logic.	This is a discreet unit of study that could potentially be taught at any point of the curriculum. It actually appears and is taught in part throughout the other units of work. This can be taught at any point within the curriculum.	• understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • recognise common uses of information technology beyond school.
	Knowledge in Computer Science - Programming (careers in computer science)	This unit of work can be done any point after algorithms and programming. Knowledge of the software development process, including iterative design principles. Understand how to complete the initial concept and analysis stages, as well as design, implementation, testing routines and evaluation of the completed solution. Understand how to establish SMART objectives using thorough research and user/client engagement to develop an effective software solution or project investigation.	• Understand how programming skills are used within different careers

If there is a class/s that are shared between 2 teachers, the curriculum will follow a slightly different style. If there is a split with the teachers, the units of work will be shared and taught individually. The units of work which can be taught on their own (E-safety, ethics, careers in computer science, data representation) will be taught by one teacher whilst the other follows the SOW missing out these units. Once these units have been completed pupils will have covered systems architecture, software and wired and wireless networks so will have some knowledge on how computers work. Here, algorithms and programming techniques will be taught. There will always be overlap with topics as concepts in computer science do.

This however rarely happens in year 9.

J276

Year	Unit of work	Rational	Key skills
KS4	Components of a computer system – Hardware and software.	This unit of work looks at systems architecture in more depth than previously. Many of these components are referred to in the subsequent units, therefore it is important that pupils have a sound knowledge on how they work.	• recognise common uses of hardware components. • Understand how computers process data and produce information. • Understand what data is and how analogue data if transferred to digital data. • use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school. • Understand the fundamental need for software and its different uses.
	Wireless and wired networks – Topologies, LAN/WAN, system security and Hardware components	This unit of work cannot be taught without a sound understanding of the previous unit.	• understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration. • recognise common uses of information technology beyond school.

			• use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour;
	Ethical, legal and cultural issues - Impact of digital technology	This unit could be taught at any point of the curriculum. However, we feel a sound understanding systems architecture and software systems is needed to full understand this unit as technical terms are used.	• use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school. • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact. • can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.
	Algorithms – Flow diagrams, pseudocode, sorting and searching algorithms.	We introduce this creation of flow diagrams as a design for robust programs at the beginning of the year. We do this by setting aside one lesson a week to learn how to design and code a robust program for a given problem. This is a large and complex unit so it is important that pupils understand the key concept early on in year 10. The searching and sporting algorithms are less imperative and are taught in this order of the	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation.

		units. As of the moment this also prepares pupils for the NEA section of the course.	• analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Programming techniques – Text based programming languages.	See above	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing

			computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Design, testing and IDEs	The design aspect and the use of an IDE are explained and addressed in the above two units. Testing happens when pupils develop an algorithm, code a solution and then test the program. However, formal testing is taught at this point of the curriculum and is a natural antecedent of programming. This again prepares pupils for the NEA aspect of the course currently.	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling

			or simulating physical systems; solve problems by decomposing them into smaller parts. • use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Data representation – Images, audio, binary and logic.	This is a discreet unit of study that could potentially be taught at any point of the curriculum. It actually appears and is taught in part throughout the other units of work.	• understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • recognise common uses of information technology beyond school.

J277 (2022 exams onwards)

Year	Unit of work	Rational	Key skills
KS4	Exploration and inspiration career skills and how they relate to the workplace.	It is important to discuss careers within the curriculum. Pupils will identify and learn how the computer science curriculum can be applied in a career. This lesson will be used to identify five key skill areas that are developed in computer science and pupils should appreciate that the skills used in the classroom are directly relevant to the workplace.	• develop their capability, creativity and knowledge in computer science, digital media and information technology • develop and apply their analytic, problem-solving, design, and computational thinking skills • understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to identify and report a range of concerns.
	Components of a computer system – Hardware and software.	This unit of work looks at systems architecture in more depth than previously. Many of these components are referred to in the subsequent units, therefore it is important that pupils have a sound knowledge on how they work.	• recognise common uses of hardware components. • Understand how computers process data and produce information. • Understand what data is and how analogue data if transferred to digital data. • use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school.

			• Understand the fundamental need for software and its different uses.
	Data representation – Images, audio, binary and logic.	This is a discreet unit of study that could potentially be taught at any point of the curriculum. It actually appears and is taught in part throughout the other units of work.	• understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • recognise common uses of information technology beyond school.
	Wireless and wired networks – Topologies, LAN/WAN, system security and Hardware components	This unit of work cannot be taught without a sound understanding of the previous unit.	• understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration. • recognise common uses of information technology beyond school. • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour;
	Ethical, legal and cultural issues - Impact of digital technology	This unit could be taught at any point of the curriculum. However, we feel a sound understanding systems architecture and software systems is needed to full understand this unit as technical terms are used.	• use technology purposefully to create, organise, store, manipulate and retrieve digital content. • recognise common uses of information technology beyond school.

			• use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact. • can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.
	Algorithms – Flow diagrams, pseudocode, sorting and searching algorithms.	We introduce this creation of flow diagrams as a design for robust programs at the beginning of the year. We do this by setting aside one lesson a week to learn how to design and code a robust program for a given problem. This is a large and complex unit so it is important that pupils understand the key concept early on in year 10. The searching and sporting algorithms are less imperative and are taught in this order of the units.	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

			• use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Programming techniques – Text based programming languages.	See above	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

			• use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Design, testing and IDEs	The design aspect and the use of an IDE are explained and addressed in the above two units. Testing happens when pupils develop an algorithm, code a solution and then test the program. However, formal testing is taught at this point of the curriculum and is a natural antecedent of programming. This again prepares pupils for paper 2 aspect of the course.	• understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems. • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. • use sequence, selection, and repetition in programs; work with variables and various forms of input and output.

			• use logical reasoning to explain how some algorithms work and to detect and correct errors in algorithms and programs.
	Careers in computer science	Visitor from outside of school to visit to discuss careers in the curriculum. This can be taught at any point within the curriculum when speaker is confirmed.	• Understand how skills in computer science are used in the real world.