Computer Science Curriculum Overview Plan

Whole school curriculum intent

Develop a broad and balanced curriculum that enables students to learn, recall and apply knowledge and skills across different contexts, supported by a robust and consistent approach to assessment. This will lead to successful and resilient lifelong learners who can cope in a range of changing contexts.

Computer Science curriculum intent

In Computing we equip pupils to use computational thinking and creativity, to view and understand new methods of breaking down complex problems and change the way they look at the world. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge, students are equipped to use a range of tools to create a number of projects related to real world scenarios and problems.

Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the **future workplace** and as active participants in a growing **digital world**.

Key stage 3 and 4 learning is focused around three main themes:

1) How computers and digital systems work and the principles behind their use
2) How logical thinking and programming can be used to solve problems.
3) What impact technology has on our lives and how can it be applied in the workplace.

These themes are revisited in every year in a spiral curriculum, so that core knowledge is revisited and understanding is deepened, so that students are well prepared for exam success.
Student learning is all hosted on Microsoft teams and regularly assessed electronically. This provides granular data to teachers and students, making it possible to focus teaching on learning gaps. This also supports regular review as knowledge organisers and revision activities are available online.

Computer Science Curriculum Overview Plan

Term	Year 7	Year 8	Year 9	Year 10	Year 11
Autumn 1	**Introduction to computing and e safety** **Can we ever be safe online?** **(USCER unit)**	**Introduction to spreadsheets** **Can we accurately model the world using computer software?** **(Unit 4, spreadsheets)**	**Ethical, Legal and environmental concerns** **Unit 9**	**Unit 7 Programming fundamentals** **What types of programs are quicker to solve with a program, and which are slower?** **Unit 1 Computer Architecture** **How can we design the fastest computer in the world?**	**Unit 5 Legal and ethical impact of computer.** **How computers made the world a better place?**
Autumn 2	**Understanding computers** **How can we design the fastest computer in the world?** **(Unit 2)**		**Networks: Will the internet slow down as it gets bigger and grows older.** **(networks unit)**	**Unit 2 Data representation** **Is there anything that cannot be represented by 0s and 1s?** **Unit 3 Networks** **Why can data travel securely across a network?**	Review and practice questions **Units 7**, **Unit 1** and **Unit 2**
Spring 1		**Cyber security** **Why is our data so valuable to hackers?** **(Unit 6, cyber crime)**		**Unit 4 Network security and system software** **How can we guarantee that data will not be hacked when it travels across the internet?** **Unit 6 Algorithms** **How can we think more like a computer?**	Review and practice questions **Units 3**, **Unit 4** and **Unit 6**
Spring 2	**Introduction to programming** **Can a computer be more intelligent than the human who programmed it?** **(Intro to Python unit)**		**Further programming** **How can we solve problems with efficient programs** **(Further python unit)**	**Unit 8 Producing robust programs** **How can we solve problems with efficient programs**	Review and practice questions **Units 8** and **Unit 5** Full practice papers, final revision
Summer 1		**Efficient programming** **How can we think more like a computer?** **(Computational thinking Unit)**			Full practice papers, final revision
Summer 2	Theory revision How does theoretical knowledge help us become better Computer Scientists?	Theory revision How does theoretical knowledge help us become better Computer Scientists?	Theory revision How does theoretical knowledge help us become better Computer Scientists?	Theory revision How does theoretical knowledge help us become better Computer Scientists?	

Year Group		Autumn	Spring	Summer
Year 7	Topic	**Aut 1: Introduction to computing and e-safety** **Can we ever be safe online?** **Aut 2: Understanding computers** **How can we design the fastest computer in the world?**	**Spr 1: Understanding computers** **How can we design the fastest computer in the world?** **Spr 2: Introduction to programming** **Can a computer be more intelligent than the human who programmed it?**	**Sum 1: Introduction to programming** **Can a computer be more intelligent than the human who programmed it?** **Sum 2: Theory revision** **How does theoretical knowledge help us become better Computer Scientists?**
	Core knowledge for this topic	**Aut 1:** • File management • Social networking risks • Password use • Using email • Critically searching the web • ASSESSMENT **Aut 2** • Elements of a computer (inputs and outputs) • How the CPU works • 8 Bit binary conversion	**Spr 1** • Binary addition • Storage devices, how they work and have developed • New technologies and their impact on the world • ASSESSMENT **Spr 2** • Strings and variables • Data types and arithmetic • Selection • Algorithms	**Sum 1** • While loops • Searching • Programming challenges • ASSESSMENT **Sum 2** • Review “Can we ever be safe online” • Review “How can we design the fastest computer in the world” • Review “Can a computer be more intelligent than a human” • FINAL ASSESSMENT
	Links to national curriculum	• Understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns • Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems	• Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems • Understand how instructions are stored and executed within a computer system • Understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]	• Understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem • Use Python to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
	Previous content that this topic builds upon	This is new content that is meant to bridge the gap of different schools, teaching different content at KS2. Some students will have had E-safety assemblies or some ICT-based lessons that would be more PSHE focused. This unit will create a strong foundation for students to build onto in KS3.	This is new content for students who will not have looked at this at KS2.	Students may have completed some programming elements at Primary such as Kodu game lab or Scratch programming that is very visually based with pre-defined options for students to plug and play with. Python will allow students to take some of the key ideas of sequence, selection and iteration and put them into practice with independent lead tasks using a textual language.
	Key vocabulary	**New Vocabulary** • File Explorer • Teams • Cloud Storage • Social Network • Email • Reply • CC • BCC • Attachment • Search engine • Web browser • Data	**New Vocabulary** • CPU • RAM • Main memory • Secondary storage • ALU • Register • Binary • Hex • Magnetic • Solid state • Emerging technology	**New Vocabulary** • Strings • Integer • Float • Data types • Arithmetic • Casting • Selection • Iteration • Binary numbers • Indentation • Concatenation

Computer Science Curriculum Overview Plan

		• Advanced search		
	Development of cultural capital	• How modern communications developed from the telegraph and morse code. • The origins of the Von Neumann programmable computer. • The origins of binary from mechanical computers and electrical circuits	• Development of hard disk technology, from tape to magnetic and then solid state. • What future developments in technology could do to change the world. • The ethical implications of computing making jobs obsolete and controlling more aspects of our lives. • Beginnings of programming and logical thinking established by Ada Lovelace. •	• How early programming languages e.g. Basic, established conventions used today. • Logical thinking principles behind programming
	Development of reading	Read texts on: • Online safety • Effective passwords • Searching the web • Fetch – Decode – Execute cycle • History of binary	Read texts on • Developments in data storage • Future technological developments • Data types and why they are important • How algorithms work.	Read texts on • Creating a while loop • Types of searches • Programming and problem solving
	What students can do at end of topic	Students will be able to be proactive and safe users of the internet and **advice and support others in its use**.	Students will be able to explain how a computer uses binary signals to express and compute information white also comparing different components and its uses and helping match them to the criteria required by an end user.	Students will be able to create and edit simple Python programs to solve a range of problems.

Year Group		Autumn	Spring	Summer
Year 8	**Topic**	**Introduction to spreadsheets** **Can we accurately model the world using computer software?**	**Cyber security** **Why is our data so valuable to hackers?**	**Sum 1: Efficient programming** **How can we think more like a computer?** **Sum 2:**
	Core knowledge from this topic	• Computer models • Financial models • What if scenarios • Conditional formatting and validation • Macros and charts • Spreadsheet modelling • ASSESSMENT	• Email scams • Computer misuse • Protecting personal data • Copyright • Health and safety • ASSESSMENT	• Logical thinking • Logic gates • Algorithmic thinking • Abstraction • Decomposition • ASSESSMENT
	Links to the national curriculum	• design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems • use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions	• Understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns	• understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal] • understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
	Previous content that this topic builds upon	• Logical thinking and programming languages from the programming unit in year 7 •	• Links to e safety and use of the internet from year 7. Gets students to think more about the security of data and the importance of passwords and encryption.	• Links to binary in Year 7 • Links to Python programming in Year 7 • Links to modelling at start of year 8 •
	Key vocabulary	• Row • Column • Value • Worksheet • Cell • Active cell • Label • Function • Conditional formatting • Validation • Macro • Chart	• Exams Scams • Phishing • Trojan Horse • Ransomware • Malware • Virus • Bots • Hacking • Identify theft • Passwords • Copyright • Plagiarism • Laws • Regulations • Safety • The environment	• Algorithms • Logic gates • AND OR NOT • Abstraction • Decomposition • Binary • Data representation • Loop and nested loop
	Development of cultural capital	• File based systems from the 1960s • Relational models of data • Online processing of data • Use of big data today	• Development of encryption and how technology is changing crime • Historical origins of encryption from Caesar cipher, the enigma code and modern 256 bit encryption. • Potential impact of quantum computing on future web security.	• Development of early programming by Alan Perlis and Donald Knuth (the father of algorithms). • Importance of algorithms in shaping the modern world
	Development of reading	Read texts on: • Anatomy of a spreadsheet • What is a relational database?	Read texts on: • The talk talk data breach • History of encryption and ciphers	Read texts on: • Early work of Knuth and Perlis in developing algorithms • The characteristics of an algorithm

Computer Science Curriculum Overview Plan

		• What if scenarios • How charts make data accessible	• Cyber crime and its impact on computing • Environmental impact of technology	• Representing data with binary • Using abstraction and decomposition to solve problems
	Concepts –what will students be able to do at the end of the topic	Use a data model to predict what will happen in different situations and present data in a visual format	• Students will be able to identify malware and give advice on treatment. • Students will advise on legal practice in computing and respective health and safety concerns including recycling.	• Solve problems using logic gates and abstraction • Create simple algorithms • Make links between algorithms, spreadsheets and Python

Year Group		Autumn	Spring	Summer
Year 9	**Topic**	Ethical, Legal and environmental concerns Does technology make the world a better place?	Networks: Will the internet slow down as it gets bigger and grows older.	Further programming How can we solve problems with efficient programs
	Core knowledge from this topic	• Cultural and ethical concerns • Computers in today's world • The environmental impact of computers • Legislation and privacy • Application of legislation • ASSESSMENT	• The internet • Connectivity • Topologies • Client-server networks • Encryption • ASSESSMENT	• The basics • Loops • Lists • Introducing functions • Functions returning values • Arrays and lists • ASSESSMENT
	Links to the national curriculum	• create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability • understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns	• understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems	• use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions • understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
	Previous content that this topic builds upon	• E safety unit in Year 7, building on personal protection to wider concerns. • Cybercrime unit in Year 8, building from crime to other implications of technology	• Computer architecture in year 7, looking at how computers work as part of a network • Cybercrime in year 8, how networks increase the risks of cyber crime and methods used in network design to combat this.	• Introduction to Python in year 7, building on the basics. • Spreadsheets logical thinking from Year 8, how these can be developed in Python. • Algorithmic thinking unit in year 8, how can these principles be applied in practice using Python
	Key vocabulary	• Phishing • Trojan Horse • Ransomware • Malware • Virus • Bots • Hacking • Identify theft • Passwords • Copyright • Plagiarism • Laws • Regulations • Safety • The environment • Recycling	• The internet • The world wide web • Networks • Connections • Latency • Bandwidth • Topology • Bus • Star • Mesh • Client • Server • Node • Switch • Hub • Router • Server • Encryption	• Strings • Integer • Float • Data types • Arithmetic • Casting • Selection • Iteration • Binary numbers • Indentation • Concatenation
	Development of cultural capital	The morality of computing and developing technology. Are they making the world a better or worse place?	Bob Kahn and Vint Cerf, the creation of TCP and IP, making the internet possible. Tim Berners Lee and the creation of the world wide web	• Development of early programming by Alan Perlis and Donald Knuth (the father of algorithms). • Importance of algorithms in shaping the modern world

Computer Science Curriculum Overview Plan

	Development of reading	Reading on technologies impact on • The environment • The law • Employment • Social relationships	Reading texts on: • Packet switching, how the internet works? • Internet protocols, how do they manage the internet? • Different network topologies	Reading texts on • The importance of arrays in allowing program to access data • How functions can improve the effectiveness of programs. • Quantum computing and its implications for programming
	Concepts –what will students be able to do at the end of the topic	• Articulate the positive and negative impacts of technology • Evaluate the size of the impact and have view on if this is overall positive or negative	• Explain the basics of how networks allow computer to communicate • Know the advantages and disadvantages of networks • Explain different topologies and why they are used	• Students will be able to confidently build a series of complex Python programs to solve a range of scenarios using a range of skills and techniques.

Computer Science Curriculum Overview Plan

Year Group		Autumn Term 1a	Autumn Term 1b+2a	Autumn Term 2a	Autumn term 2b	Spring term 1	Spring Term 2
Year 10	**Topic**	**Unit 7: Programming skills** What types of programs are quicker to solve with a program, and which are slower?	**Unit 1: Systems Architecture** How can we design the fastest computer in the world?	**Unit 2: Data Representation** Is there anything that cannot be represented by 0s and 1s?	**Unit 3: Network, connections and protocols** Why can data travel securely across a network?	**Unit 4: Network security and systems software** How can we guarantee that data will not be hacked when it travels across the internet?	**Unit 6 Algorithms** How can we think more like a computer?
	Core knowledge from this topic	• Recap programming fundamentals • Selection and sequence statements. • iterative statements • Procedures and functions • Calling text files • Query statements in SQL	• Parts of a processor. • Processor speed variables. • main and secondary storage, benefits and uses. • Magnetic and solid state storage. • How data is stored	• Binary to deanery conversion. • Binary addition, hexadecimal conversion. • Character sets, ASCII and Unicode. • Sound storage • Picture storage • Compression types (lossy and lossless)	• Internet protocols and world wide web. • LAN and WAN, characteristics and differences. • Wireless networks, wifi and Bluetooth, including potential issues. • Client server networks and peer to peer networks. • Network topologies. • Network protocols and their TCP/IP network layers.	• Common network threats. • Common vulnerabilities • Preventing network threats • Operating systems • Utility software	• What is computational thinking? • Searching algorithms using Binary, linear and random search. • Sorting algorithms including Bubble, merge and insertion sort. • Flowcharts • Pseudocode • Searching algorithms.
	Links to the national curriculum (if applicable)	N/A	N/A	N/A	N/A	• N/A •	N/A
	Previous content that this topic builds upon	• Y7, intro to Python • Y8, spreadsheets • Y9, further python	• Y7, understanding computers	• Y7, understanding computers • Y8, computational thinking	• Y9 networking	• Y8, cybercrime	• Y7, intro to Python • Y8, computational thinking • Y9, further python
	Key vocabulary	• Strings • Integer • Float • Data types • Arithmetic • Casting • Selection • Iteration • Binary numbers • Indentation • Concatenation • Array • List • Text • CSV	• Binary • Storage • CPU • ALU • CU • BUS • Control • Memory • Data • Cache • Cores • Speed • RAM	• Binary • Storage • Addition • Hexadecimal • Conversion • Twos compliment • Characters • ASCII • Unicode • Extended ASCII • Resolution • Pixel • RGB	• Internet • World wide web • Network • Bandwidth • Latency • Wireless • WAP • WPA • LAN • WAN • PAN • Client • Server	• Malware • Virus • Worm • Bot • Botnet • Trojan-horse • Ransomware • Phishing emails • Vulnerabilities • Operating system • Firewall • Anti-virus • Utility software	• Abstraction • Decomposition • Linear search • Random search • Binary search • Flowcharts • Subprograms • Merge sort • Bubble sort • Insertion sort

		• IDE • SQL	• ROM • SSD • HDD • CD • DVD • Flash • Lazer • Electro-magnet • Electricity • Charge • Pit • Wavelength • Architecture	• File Size • Gigabyte • Megabyte • Kilobyte • Byte • Nibble • Bit • DAC • ADC • Microphone • Frequency • Pitch • Bit Depth • Colour Depth • Lossy • Lossless Compression	• Node • Switch • Hub • Topology • Star • Bus • Mesh • Ring • Protocols • FTP • POP • IMAP • SMTP • Layers • Link • Network • Application • Transport	• Spyware •	
	Development of cultural capital	Logical thinking principles behind programming. Boolean logic origins and more recent application	Moore's law and the pace of change. Potential of Quantum computing	Ada Lovelace and data representation. Development of image and sound formats and their impact on the growth of data today.	How networks and protocols have lead to the internet and its impact on the world today	Students will see the impact of cybercrime and how it impacts business and society.	Rosser and Kleene, and the "Church-Turing thesis". The principles of algorithmic thinking
	Development of reading	1) Programming principles 2) The power of loops 3) Application of Python 4) George Boole and the logic revolution	5) Fetch Decode Execute cycle 6) Chip design 7) Moore's law and exponential growth in storage 8) Quantum computing	9) Character set development (Ascii and Unicode) 10) How images use lossy compression. 11) Sound representation in a streamed world	12) The origins of the internet 13) Creation of the world wide web 14) How protocols make the internet go around	15) How the next war will be won (cyber espionage) 16) Is encryption ever truly secure?	17) The principles of Algorithms 18) Can algorithms replicate thought?
	Concepts –what will students be able to do at the end of the topic	Students will be able to confidently build a series of complex Python programs to solve a range of scenarios using a range of skills and techniques.	Students will be able to explain how the computer utilises the main components and how they affect its performance. Students will review and compare the effects of different components on computer performance.	Students will be able to convert and add binary numbers from denary and convert them to Hexadecimal. Students can work out file size of images and sound files from the appropriate data given. Students will compare compression methods and the difference of impact on the file type.	Students will be able to see how computers are connected and share information in a variety of different scenarios and how we can measure performance of these networks against one another whilst looking for best practice.	Students will be able to identify common forms of malware and how they can be tackled and prevented. Students will be able to see the impact on society of this malware and why we should encourage good practice to all people.	Students will be able to decompose complex algorithms to their component parts and build their problem-solving skills to develop algorithms

Computer Science Curriculum Overview Plan

Year Group		Summer Term 1	Summer Term 2	Year Group		Autumn Term 1	Autumn Term 2	Spring term 1
Year 10 continued	**Topic**	**Unit 8: Logic and Languages** How can we solve problems with efficient programs	**Review and Year 10 Exam**	**Year 11**	**Topic**	**Unit 5: Impacts of digital technology**	**Application in programming and exam questions: Unit 7, Unit 1 and Unit 2**	**Application in programming and exam questions: Unit 3, Unit 4 and Unit 6**
	Core knowledge from this topic	• Logic diagrams and truth tables. • Defensive design. • Error checking and debugging. • Assemblers and compilers • IDE	• Unit 1 • Unit 2 • Unit 3 • Unit 4 • Unit 6 • Unit 7 • Unit 8		**Core knowledge from this topic**	• ethical and cultural issues • Environmental impact of computers • Responsible e citizens • impact of legislation • Impact on wider society	**Unit 7** • Recap programming fundamentals • Selection and sequence statements. • iterative statements • Procedures and functions • Calling text files Query statements in SQL **Unit 1** • Binary to deanery conversion. • Binary addition, hexadecimal conversion. • Character sets, ASCII and Unicode. • Sound storage • Picture storage • Compression types (lossy and lossless) **Unit 2** • Internet protocols and world wide web. • LAN and WAN, characteristics and differences. • Wireless networks, wifi and Bluetooth, including potential issues. • Client server networks and peer to peer networks. • Network topologies. • Network protocols and their TCP/IP network layers.	**Unit 3** • Internet protocols and world wide web. • LAN and WAN, characteristics and differences. • Wireless networks, wifi and Bluetooth, including potential issues. • Client server networks and peer to peer networks. • Network topologies. • Network protocols and their TCP/IP network layers. **Unit 4** • Common network threats. • Common vulnerabilities • Preventing network threats • Operating systems • Utility software **Unit 6** • What is computational thinking? • Searching algorithms using Binary, linear and random search. • Sorting algorithms including Bubble, merge and insertion sort. • Flowcharts • Pseudocode • Searching algorithms.
	Links to the national curriculum (if applicable)	N/A			**Links to the national curriculum (if applicable)**	N/A		
	Previous content that this topic builds upon	• Y7 intro to Python • Y8 Computational thinking • Y9 Further Python • Y10 Programming skills • Y10 Computational thinking			**Previous content that this topic builds upon**	• Y9 legal and environmental concerns		
	Key vocabulary	• Code • IDEs • Compilers • Defensive design • Debugging • Logic diagrams • Truth tables • Structure diagram • Module • Authentication • Syntax error • Logic error • Iterative Testing			**Key vocabulary**	• Ethical • Reduce • Reuse • Recycle • Law • Legislation		
	Development of cultural capital	Development of high level computer languages and how they have evolved.			**Development of cultural capital**	Students will review the practice of themselves and schools and businesses to see how we can be better digital citizens and use technology more efficiently.		

	Development of reading	19) The origins of programming 20) High level or low level? 21) Debug to perfection.			**Development of reading**	1) Sustainable technology 2) Technology and the law, can it keep up? 3) Workplace of the future.		
	Concepts –what will students be able to do at the end of the topic	Students will be able to review logic systems and explain if they work effectively and create their own using logic expressions. Students will be able to review different ways of creating programs depending on their requirements.			**Concepts –what will students be able to do at the end of the topic**	Students will look at the impact of technology and how it can be used responsibly more consistently and recycle their old computers rather than sending them to landfill. Students will look at the impact of E-Waste in developing nations and how we can support them.		

Year Group		**Spring Term 2**	**Summer Term 1**
Year 11 **Continued**	**Topic**	**Application in programming and exam questions: Unit 8 and Unit 5**	**Final practice papers, Papers 1 and 2**
	Core knowledge from this topic **Links to the national curriculum (if applicable)** **Previous content that this topic builds upon** **Key vocabulary** **Development of cultural capital**	**Unit 8** • Logic diagrams and truth tables. • Defensive design. • Error checking and debugging. • Assemblers and compilers • IDE **Unit 5** • ethical and cultural issues • Environmental impact of computers • Responsible e citizens • impact of legislation • Impact on wider society	**Paper 1:** Unit 1 Computer architecture Unit 2 Data representation Unit 3 Networks Unit 4 Network security and system software Unit 5 legal, ethical, environmental **Paper 2** Unit 6: Algorithms Unit 7: Programming Units 8 Logic and languages
	Development of reading		
	Concepts –what will students be able to do at the end of the topic		

Computer Science Curriculum Overview Plan