Key Stage Curriculum Map:

The curriculum in Computer Science and ICT will ensure that students will be able to use two or more programming languages, use logical reasoning, model the state and behaviour of real-world problems and understand a range of ways to use technology safely, respectfully, responsibly and securely.

THE YEAR 7 CURRICULUM MAP

	HALF TERM 1	HALF TERM 2&3	HALF TERM 4	HALF TERM 5	HALF TERM 6	EVERY HALF TERM
Topic and Learning Focus						
	Understanding Computers	**Algorithms with Flowol**	**Modelling Data**	**Scratch**	**Using Media**	**E-Safety**
Foundational Knowledge Prior learning needed	- The key components of a computer. - The difference between hardware and software.	- Understand that computers need precise instructions. - Identify common types of sensors used in control systems	- Understand the importance of collecting, organising and presenting data for different purposes.	- Predict the outcome of a simple sequence. - Define a sequence. - Compare how humans and computers understand instructions.	- Selecting the most appropriate software to use to complete a task.	- Understand that using the Internet can be unsafe.
Core Knowledge and Skills	- Identifying input, output and storage devices. - Explain how a computer input, processes and outputs	- Identify control flowchart symbols and understand how they are used to describe systems. - Understand why a	- Identifying columns, cells, rows and cell references. - Use formatting techniques in a spreadsheet.	- Recognise that computers follow the control flow input, process, output. - Define what a variable is.	- Identify the key features of a word processor. - Apply key features of a word processor to format a document.	- Describe how to communicate with peers online. - Be able to check who you are talking to online.

	information. ● Who was Alan Turing and why was his contribution to modern computing so significant.	control system might fail and explain the impact this can have on safety. ● Use decision systems in a flowchart.	● Use basic formulas with cell references for calculations in a spreadsheet. ● Use the autofill tool to replicate the cell data.	● Trace the values of variables within a sequence. ● Define a condition as an expression that will be evaluated as either ‘true’ or ‘false’. ● Define logical operators AND, OR & NOT.	● Select appropriate images for a given context. ● Demonstrate the ability to credit the original source of an image.	● Explain what sources online can be trusted and why. ● Understand how cyberbullying can happen. ● Know who you can report cyberbullying to.
Developmental Knowledge and Skills	● How the fetch - decode - execute cycle works in the CPU. ● How the speed of the processor is measured. ● What RAM and ROM are used for. ● Binary conversion. ● Future of technology like driverless cars, artificial intelligence, home help robots and robot pets.	● Develop a control flowchart solution for a simple problem. ● Develop a control solution for a system that uses two flowcharts operating in sequence. ● Understand how the use of the subroutines can make programs more efficient.	● Explain the difference between data and information. ● Explain the difference between primary and secondary sources of data. ● Collect data. ● Analyse data. ● Use a spreadsheet to sort and filter data.	● Modify a sequence. ● Make a sequence that includes a variable. ● Identify that selection uses conditions to control the flow of a sequence. ● Define iteration as the process of repeatedly executing instructions. ● Use logical operators AND, OR & NOT.	● Construct a blog using appropriate software. ● Organise the content of the blog based on credible sources. ● Check digital content for credibility / trustworthiness.	● Understand how to be respectful when communicating online. ● Learn why it is important to know who you are talking to online.
Complex Knowledge	● Define Hz, MHz and GHz and how they relate to the speed of the processor. ● Binary addition and why it’s necessary. ● Describe how binary digits are represented on a CD.	● Develop a control solution for a system that uses multiple sensors. ● Develop a control solution for a system that includes a subroutine. ● Understand what a variable is and explain how variables can be	● Use the functions SUM, MAX, MIN and COUNTA in a spreadsheet. ● Create appropriate charts in a spreadsheet. ● Use the functions COUNTIF, AVERAGE and IF in a spreadsheet.	● Identify where count-controlled interaction can be used in a program. ● Implement count-controlled iteration in a program. ● Identify where selection statements can be used in a program.	● Evaluate formatting techniques to understand why we format documents. ● Use citations and recognise the concept of plagiarism. ● Apply referencing techniques that credit authors appropriately. ● Design the layout of	● Understand the right way to search to find what you are looking for safely online. ● Be able to make sense of what online content is based on facts, half-truths and lies.

🟥		used in control systems. ● Develop a control solution for a system that uses variables.		● Detect and correct errors in a program (debugging).	the content to make it suitable for the audience.	
Links with the National Curriculum	● Understand how numbers can be represented in binary and be able to carry out simple operations on binary numbers. ● Understand the difference between hardware and software. ● Describe how computer components communicate (inputs, processes and output information).	● Understand several key algorithms that reflect computational thinking and use logical reasoning to compare the utility of alternative algorithms for the same problem. ● Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems.	● Undertake creative projects that involve selecting, using and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users.	● Design, use and evaluate computational abstractions that model the state and behaviour of real world problems and physical systems. ● Use two or more programming languages to solve a variety of computational problems. Design and develop modular programs that use procedures or functions.	● Create, re-use, revise and re-purpose digital artefacts for a given audience with attention to trustworthiness, design and usability. ● Undertake creative projects that involve selecting, using and combining multiple applications.	● 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.
Literacy (including reading)	● Key vocabulary worked through in the knowledge organiser. ● Use of the textbooks in lessons to encourage reading and comprehension. ● Research tasks to be completed by searching for and reading different sources on the	● Key vocabulary worked through in the knowledge organiser. ● Learning a variety of new words and applying them to the real-world problem solving scenarios presented in the unit.	● Key vocabulary worked through in the knowledge organiser. ● Use of the textbooks in lessons to encourage reading and comprehension.	● Key vocabulary worked through in the knowledge organiser. ● The visual coding structure for Scratch (the use of blocks) builds students' understanding about language and the key terminology used in programming.	● Key vocabulary worked through in the knowledge organiser. ● Research tasks to be completed by searching for and reading different sources on the Internet.	● Key vocabulary worked through in the knowledge organiser. ● Safe searching on the Internet tasks using keywords.

	Internet.					
Cultural Capital	● Students will study current technologies such as: the CPU, RAM, ROM, motherboard, NIC etc. As well as future technology such as: driverless cars and robotic pets.	● Students learn to solve real-world problems using algorithms such as: the monitoring systems in greenhouses to optimise food production and the optimization of central heating systems to improve the climate crisis.	● Students learn skills suitable for the business world such as event planning and data management as well as everyday life skills such as budgeting.	● So much of our world is automated, students will learn to control the interaction between people and machines, through the use of a visual programming language.	● Students learn about creating online content for a given audience, taking care to recognise where copyright laws may exist and how they should be respected.	● Students learn how to stay safe online, how to search for the exact content they need, how to communicate with others online and what dangers there are on the Internet.
Social, Moral, Spiritual and Cultural Development	● Cultural, learning about Alan Turing and his contribution to modern computing and the end of WWII. ● How the building of computers and other devices contributes negatively to other countries and why it is important to recycle e-waste.	● Linking their understanding of how automated systems can be an efficient solution to issues such as climate change, healthcare systems and care for the elderly.	● Building students self-knowledge of how budgets work. ● Learning how sources of data can be a valuable resource if used correctly and how data can be easily misused.	● Ethical issues with regards to programming, such as: addictive design, giving proper credit for intellectual property and contributing to society and human well being.	● Building students self-knowledge, self-esteem and self confidence; when creating content that they will put out into society. ● Learning to take responsibility for the content they put out into the public domain.	● Students learn about the different forms of bullying. ● Students learn about inappropriate contact from strangers on the Internet including how to spot if someone is a groomer.
Fundamental British Values	● Using computers positively, for the good of society. ● The history of the modern computer, including the influence of key historical figures from the UK in the development of	● Teamwork, mutual respect and tolerance built through solving real-world problems with the use of algorithms.	● Respect for others.	● Being able to be creative, expressing themselves with their code whilst respecting each other's ideas and opinions.	● Respect for the laws around creating original content and learning what to do if you want to use other peoples online content.	● Respect for the law and respect for others.

	modern day technology.					
Assessment	● Google form, multiple choice assessment, auto mark, 40 marks. Assesses student knowledge of binary conversion, input and output devices and RAM vs ROM. ● Exam style questions, provided on paper, based on for example the fetch - decode - execute cycle, how processor speeds are measured or binary addition showing working, 10 marks.	● Google form, multiple choice assessment, auto mark, 30 marks. Assesses student knowledge of flowchart symbols, sensors and how to sequence instructions to solve a problem. ● Practical task where students solve three problems with varying degrees of difficulty with the use of a flow chart to map out the solution, 20 marks.	● Google form, multiple choice assessment, auto mark, 20 marks. Assesses student knowledge of the functions and formulas used in spreadsheets, the difference between data and information and the methods of data collection. ● Creation of a spreadsheet, from data collected from the other students in the classroom. Marks will be awarded for the type of data collected, how it is organised, the functions, sorting and filtering used in the spreadsheet and how the data is presented by charts and graphs, 20 marks.	● Google form, multiple choice assessment, auto mark, 20 marks. Assesses student knowledge of the block based programming language Scratch and the different programming techniques students have learnt. ● Practical task where students solve three problems with varying degrees of difficulty with the use of Scratch, 20 marks.	● Google form, multiple choice assessment, auto mark, 20 marks. Assesses students knowledge of different softwares for creating online content, crediting image sources and using citations. ● Students will create a blog using the appropriate software, consolidating knowledge they have learned so far on the course and which will be useful going into Year 8, 20 marks.	● Assessed in class with the use of the knowledge organisers and mini plenaries.

Key Stage Curriculum Map:

The curriculum in Computer Science and ICT will ensure that students will be able to use two or more programming languages, use logical reasoning, model the state and behaviour of real-world problems and understand a range of ways to use technology safely, respectfully, responsibly and securely.

THE YEAR 8 CURRICULUM MAP

	HALF TERM 1	HALF TERM 2&3	HALF TERM 4	HALF TERM 5	HALF TERM 6	EVERY HALF TERM
Topic and Learning Focus						
	Networks	**Video Editing**	**Introduction to Python**	**Mobile App Development**	**Physical Computing with Micro:bit**	**E-Safety**
Foundational Knowledge Prior learning needed	● Describe where a network would be located.	● Know the features of video as a visual media format. ● Recognise which devices can and can't record video.	● Know that Python is a programming language. ● Know the types of programs that can be made in Python.	● Describe what an App is and what you might use one for.	● Describe what the Micro:bit is. ● List the built in components for input and output.	● Understand that using the Internet can be unsafe and some of the ways we can protect ourselves. ● Understand the possible dangers of social networking sites.
Core Knowledge and Skills	● Define what a computer network is. ● List network hardware. ● Explain how data is	● Be able to use different camera angles. ● Explain the purpose of a storyboard and	● Define algorithms and programs and describe how they are executed. ● Write and execute	● Use a block based programming language to create a sequence. ● Pass the value of a	● Use variables and data structures to keep track of information. ● Combine features to	● The ability to use information and communication technologies to find,

	transmitted between computers across networks. ● Define what the Internet is. ● Compare wired to wireless connection. ● Explain the difference between the Internet, its services and the World Wide Web. ● Explain the term ‘connectivity’. ● Describe how services are provided over the internet. ● Describe components of a network and how they work together.	create one. ● Identify the features of a video recording device or application. ● Determine what scenes will convey an idea. ● Combine filming techniques for a given purpose. ● Recognise that projects need to be exported to be shared. ● Collect assets for a project in line with the Copyright, Designs and Patents Act. ● Import images/videos into video editing software.	basic Python programs. ● Use arithmetic expressions to calculate values. ● Use variables to reference and store values. ● Use variables as counters. ● Use the three control structures (sequence, selection and iteration) to build a more complex program.	variable in an object. ● Use user input in an event-driven programming environment. ● Identify when a problem needs to be broken down. ● Establish user needs when completing a creative project. ● Use a block-based programming language to include selection. ● Reflect and react to user feedback.	develop solutions to meaningful problems. ● Use IDE to write and execute Python programs. ● Use the IDE to write Python programs for the Micro:bit. ● Write programs that use the Micro:bit’s built-in output devices. ● Write programs that use the micro:bits built-in input devices. ● Select hardware components that are fit for purpose.	evaluate, create and communicate information requiring both cognitive and technical skills. ● Learn how to respond to threats on the internet. ● Know what a groomer is. ● Know why students should consider age ratings. ● Know how to report and block unkind messages when you are gaming.
Developmental Knowledge and Skills	● Define ‘protocol’ and provide examples of non-networking protocols. ● Describe keywords such as protocols, packets and addressing. ● Explain how data travels between computers across the Internet. ● Describe how internet-connected devices can affect me.	● Recognise that filming techniques can be used to create different effects. ● Identify that videos can be improved through reshooting and editing ● Use split, trim and crop to edit videos. ● Export a finished video to a given file type. ● Add appropriate effects and transitions. ● Add text and titles to a project where relevant.	● Debug syntax errors. ● Use selection to control the follow of a program between branches. ● Explore how selection can handle more than two possible branches. ● Use iteration to allow the flow of program execution to include loops. ● Use boolean operators, variables and expressions.	● Recognise that events can control the flow of a program. ● Use variables in an event-driven programming environment. ● Evaluate the success of a programming project.	● Use selection and iteration to control program execution flow. ● Call functions and use results they return in expressions. ● Locate and correct common syntax errors. ● Write programs that use the GPIO pins for input and output.	● Learn how to keep your identity secure on the Internet. ● Know what to do if a student is concerned about who they have met online. ● Knowing what is and is not appropriate behaviour for viewers whilst you are live streaming.

Complex Knowledge	● Define ‘bandwidth’ using the appropriate units for measuring the rate at which data is transmitted. ● Explain how the Internet of Everything can collect and share information about me with or without my knowledge.	● Recognise the need to regularly review and reflect on a video project. ● Decide what changes need to be made to improve a video through editing and reshooting. ● Export a finished video to a fit for purpose file type.	● Follow walkthroughs of code and keep track of variable values. ● Write programs that receive numerical input from the keyboard. ● Introduce elements of randomness into programs.	● Implement and customise GUI elements to meet the needs of the user. ● Identify and fix common coding errors in a block-based environment. ● Apply decomposition to break down a larger problem into more manageable steps.	● Use IDE to write and execute Python programs. ● Combine components of a physical computing system to solve meaningful problems. ● Design a physical computing artefact purposefully.	● Understand the warning signs of being groomed. ● Knowing how to stay safe online whilst broadcasting to an audience.
Links with the National Curriculum	● Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems.	● Undertake creative projects that involve selecting, using and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users.	● Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems. ● Design and develop modular programs that use procedures or functions.	● Use two or more programming languages, at least one of them which is textual, to solve a variety of computational problems; make appropriate use of data structures; design and develop modular programs that use procedures or functions.	● Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems.	● 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.
Literacy (including reading)	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● Extended writing and exam style questions given in the knowledge organiser.	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● Use of labels and descriptors when storyboarding. ● Script writing for the	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● New vocabulary to learn to do with new programming constructs students	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● Use of block based or textual programming style - new vocabulary for students to learn.	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● Use of block based or textual programming style - new vocabulary for students to learn.	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● Written questions to be answered in full sentences.

		video project.	would not have come across before. ● Students use and gain understanding of new vocabulary when programming in Python.	●		
Cultural Capital	● Students learn how large organisations network their computers and devices to run as efficiently as possible.	● Students learn the art of content creation for a range of audiences.	● Students learn about coding in Python which is used by important organisations from across the world such as: Intel, NASA, IBM, Pixar, Netflix, Facebook and Spotify to name just a few.	● Students learn how to create apps for commonly used devices like mobile phones and tablets.	● Students learn how robotics and computer science are intrinsically linked and how programming is used to control everyday objects such as street lights and traffic lights.	● Students learn how to stay safe online, how to search for the exact content they need, how to communicate with others online and what dangers there are on the Internet.
Social, Moral, Spiritual and Cultural Development	● Students learn about ethical issues surrounding networks, such as hacking and the way that these types of offences can be prevented.	● Students learn to work in teams on a project that needs to be managed by them from the planning phase, through development and to the delivery stage.	● Ethical issues with regards to programming, such as: addictive design, giving proper credit for intellectual property and contributing to society and human well being.	● Building students self-knowledge, self-esteem and self confidence; when creating content that they will put out into society.	● Students learn about the impact that small simple robotics can have on everyday life and what this can mean for bigger scale projects.	● Students learn what a groomer is and what the signs are of being groomed. ● Students learn more about current popular online trends like challenges and live streaming.
Fundamental British Values	● Using computers positively, for the good of society.	● Respect for the laws around creating original content and learning what to do if you want to use other peoples online content.	● Respect for others during paired programming tasks, an extreme programming methodology used in industry.	● Being able to be creative, expressing themselves with their code whilst respecting each other's ideas and opinions.	● Respect for others with regards to learning about how devices can aid people in their everyday lives.	● Respect for the law and respect for others.
Assessment	● Google form, multiple	● Google form, multiple	● Google form, multiple	● Google form, multiple	● Google form, multiple	● Assessed in class with

	choice assessment, auto mark, 40 marks. Assesses student knowledge of networks, protocols, the Internet and the World Wide Web. ● Exam style questions, provided on paper, 10 marks.	choice assessment, auto mark, 30 marks. Assesses student knowledge of the software used to create and edit videos. ● Practical project that runs through the unit, students must make a video using Serif Video. The videos will be handed in and assessed at the end of the unit. 10 marks.	choice assessment, auto mark, 20 marks. Assesses student knowledge of Python programming and the different programming techniques students have learnt. ● Practical task where students solve three problems with varying degrees of difficulty with the use of a Python, 20 marks.	choice assessment, auto mark, 30 marks. Assesses student knowledge of the code used to make the mobile app. ● Practical project that runs through the unit, students must make an app using Studio Code website. The apps will be handed in and assessed at the end of the unit, 10 marks.	choice assessment, auto mark, 20 marks. Assesses student knowledge of the Micro:bit and the code used to program it. ● Practical task where students solve three problems with varying degrees of difficulty with the use of a Micro:bit and the textual based code, 20 marks.	the use of the knowledge organisers and mini plenaries.

Key Stage Curriculum Map:

The curriculum in Computer Science and ICT will ensure that students will be able to use two or more programming languages, use logical reasoning, model the state and behaviour of real-world problems and understand a range of ways to use technology safely, respectfully, responsibly and securely.

THE YEAR CURRICULUM MAP

	HALF TERM 1	HALF TERM 2&3	HALF TERM 4	HALF TERM 5	HALF TERM 6	EVERY HALF TERM
Topic and Learning Focus						
	Cybersecurity	**Minecraft for Education**	**Computing Systems**	**Web Development**	**Vector Graphics**	**E-Safety**
Foundational Knowledge Prior learning needed	● Be able to explain the difference between data and information.	● Know the basics of how a computer game works and what the controls might be for a PC.	● Recall that a program is a sequence of instructions. ● Provide a broad definition of artificial intelligence.	● Be able to describe what Web Design is.	● List uses for designing different graphics.	● Understand that using the Internet can be unsafe and there are ways of protecting ourselves.
Core Knowledge and Skills	● Be able to explain the difference between data and information. ● Explain the need for the data protection act. ● Define hacking in the context of cybersecurity.	● Use the coding concept of sequencing to complete tasks. ● Test and code the Agent to move around in game play. ● Become familiar with Code Builder and	● Recall that program instructions specify operations that are to be performed on data. ● Recall that a general-purpose computer system is a device for executing	● Describe what HTML is. ● Describe what CSS is. ● Assess the benefits of using CSS to style pages instead of in-line formatting. ● Describe what a search engine is.	● Draw basic shapes with different properties. ● Manipulate individual objects. ● Manipulate groups of objects. ● Provide examples	● Learn about having healthy relationships and having boundaries. ● Learn about seeking help from a trusted adult to stop harmful communication. ● Learn how to create a

	● Explain the need for the Computer Misuse Act. ● List the common malware threats.	MadeCode blocks. ● Learn basic navigation in Minecraft: Education Edition. ● Embrace the coding mindset. ● Learn about boolean operators and using those within conditionals.	programs. ● Describe the function of the hardware components used in computing systems. ● Provide a broad definition of ‘operating systems’. ● Describe the AND, OR and NOT logical operators. .	● Discuss the impact of search technologies. ● Create hyperlinks to allow users to	where using vector graphics would be appropriate.	secure memorable password. ● Learn about the ways images can be altered online to improve people’s appearance. ● Encourage students to use common sense with regards to dangerous online challenges.
Developmental Knowledge and Skills	● Identify what happens to data entered online. ● Recognise how human errors pose security risks to data. ● Explain how a DDoS attack can impact users of online services. ● Identify strategies to reduce the chance of a brute force attack being successful.	● Create programs that include sequences, events, loops and conditionals. ● Develop plans that describe a sequence of events, goals and expected outcomes. ● Use the coding concepts of loops and nested loops to complete tasks. ● Debug errors in an algorithm or program that includes sequences and simple loops.	● Explain the difference between a general-purpose computing system and a purpose-built device. ● Describe how hardware components work together in order to execute programs. ● Recall the role of an operating system in controlling program execution. ● Describe how machine learning differs from traditional programming.	● Use HTML to structure web pages. ● Be able to display images within a web page. ● Use CSS to style static web pages. ● Explain how search engines ‘crawl’ through the World Wide Web and how they select and rank results.	● Combine paths by applying operations. ● Convert paths to objects. ● Draw paths. ● Edit path nodes. ● Explain what vector graphics are. ● Learn the difference between vector and bitmap images. ● Understand that logos communicate a message.	● Learn how to identify a phishing scam. ● Learn how to avoid being the victim of an email scam. ● Learn about the different messages portrayed in pornography and where support can be found if needed. ● Learn how to use critical thinking to question how real or trustworthy online challenges are.
Complex Knowledge	● Critique online services in relation to data privacy. ● Implement strategies to minimise the risk of data being	● Decompose problems into smaller manageable subproblems to facilitate the development process.	● Recall that all computing systems, regardless of form, have a similar architecture. ● Analyse how hardware	● Modify HTML tags to improve the appearance of web pages. ● Apply HTML tags to construct a web page	● Combine multiple tools and techniques to create a vector graphic design. ● Evaluate vector graphics.	● Explain that consent is fluid and cannot be assumed. ● Learn how to create online identities that make students feel good

	compromised through human error. ● Question how malicious bots can have an impact on societal issues. ● Examine how different types of malware cause problems for computer systems.	● Test and debug your own code. ● Use of paired programming to complete the challenges that test coding skills. ● Model the way programs store and manipulate data by using numbers or other symbols to represent information.	components work together in order to execute programs. ● Use logic gates to construct simple logic circuits. ● Describe the steps involved in training machines to perform tasks.	structure from a provided design. ● Analyse how search engines select and rank results when searches are made. ● Implement navigation to complete a functioning website.	● Know different design techniques used by companies.	about themselves and fosters good self esteem. ● Students learn to be proactive in coming forward
Links with the National Curriculum	● Understand the hardware and software components that make up computer systems, and how they communicate with one another and other systems.	● Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures; design and develop modular programs that use procedures or functions.	● Understand simple boolean logic and some of its uses in circuits and programming. ● Understand the hardware and software components that make up computer systems, and how they communicate with one another and other systems.	● Create, re-use, revise and re-purpose digital artefacts for a given audience with attention to trustworthiness.	● Undertake creative projects that involve selecting, using and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users.	● 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.
Literacy (including reading)	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● Extended writing and exam style questions given in the knowledge organiser.	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● New vocabulary to learn to do with new programming constructs students	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● Use of the textbooks in lessons to encourage reading and comprehension.	● Key vocabulary and written lesson tasks worked through in the knowledge organiser.	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● Written peer reviews and reviews of their own work.	● Key vocabulary and written lesson tasks worked through in the knowledge organiser. ● Reading news articles relating to E-Safety.

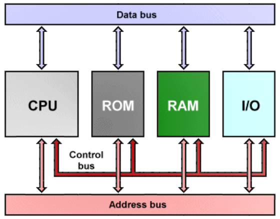
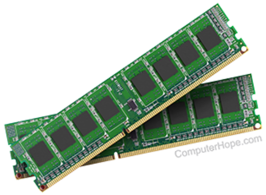
		would not have come across before.				
Cultural Capital	● Students learn about the role of cybersecurity in wider society.	● Students learn how computer games are coded and the planning behind making one.	● Students learn about the history of computers and how they have been used in different forms for centuries.	● Students learn about designing for the web and the rules for doing it effectively.	● Students learn about designing graphical content such as logos for organisations.	● Students learn how to stay safe online, how to prevent themselves from being scammed, learn about pornography and about online challenges.
Social, Moral, Spiritual and Cultural Development	● Students learn about hacking, including ethical hacking and social engineering.	● Students learn about how computer games are constructed and how to do that collaboratively like in industry.	● Learn about the different types of artificial intelligence, how advanced it is so far and its different uses.	● Students learn about creating websites responsibly, using only content that they have created or content free from Copyright.	● Students learn about the power that good representation has for an organisation in the form of logos.	● Learn about the ethical issues surrounding pornography. ● Learn about the possible life threatening implications of taking part in online challenges.
Fundamental British Values	● Respect for the law - students learn about the Computer Misuse Act 1990.	● Respect for others during paired programming tasks, an extreme programming methodology used in industry.	● Using computers positively, for the good of society. ● The history of the modern computer, going back centuries.	● Respect for the laws around creating original content and learning what to do if you want to use other peoples online content.	● Being able to be creative, expressing themselves with their code whilst respecting each other's ideas and opinions.	● Respect for the law and respect for others.
Assessment	● Google form, multiple choice assessment, auto mark, 40 marks. Assesses student knowledge of cybersecurity concepts such as: hacking, phishing, DoS attacks, brute force attacks and	● Google form, multiple choice assessment, auto mark, 40 marks. Assesses student knowledge of Minecraft coding basics. ● Practical task, students must choose a problem	● Google form, multiple choice assessment, auto mark, 40 marks. Assesses student knowledge of Computing Systems concepts such as: machine learning, hardware, circuitry and	● Google form, multiple choice assessment, auto mark, 40 marks. Assesses student knowledge of HTML code and web design principles. ● Practical project running through the	● Google form, multiple choice assessment, auto mark, 40 marks. Assesses student knowledge of vector graphics and graphic design principles. ● Practical project running through the	● Assessed in class with the use of the knowledge organisers and mini plenaries.

	malicious bots. ● Exam style questions, provided on paper, 10 marks.	to solve of varying difficulties using the Minecraft for Education interface. 10 marks.	logic gates. ● Exam style questions, provided on paper, 10 marks.	unit, students must design a website using either HTML or Google Sites, the quality of which will be assessed at the end of the unit. 10 marks.	unit, students must design a new logo for Ashlawn School, the quality of which will be assessed at the end of the unit. 10 marks.	

Key Stage Curriculum Map: Year 10

The curriculum in Computer Science and ICT will ensure that students will be able to use two or more programming languages, use logical reasoning, model the state and behaviour of real-world problems and understand a range of ways to use technology safely, respectfully, responsibly and securely.

THE YEAR 10 CURRICULUM MAP

	HALF TERM 1	HALF TERM 2	HALF TERM 3	HALF TERM 4	HALF TERM 5	HALF TERM 6
Topic and Learning Focus	 Data bus CPU ROM RAM I/O Control bus Address bus					
	System Architecture	**Memory & Storage Part 1**	**Memory & Storage Part 2**	**Computer Networks & Network Security**	**System Software**	**Ethical, Legal, Cultural & Environmental Issues**
Foundational Knowledge Prior learning needed	• Understand what the CPU of a computer does. • Be aware of the fetch, decode, execute cycle.	• Be able to describe what RAM and ROM are. • The units of data storage.	• Converting denary numbers into binary numbers and vice versa. • The need for compression.	• Know the types of networks: LAN and WAN. • Be aware of types of illegal activity online such as phishing and social engineering.	• What each function of the operating system does.	• Understand that technology introduces ethical, legal, cultural, environmental and privacy issues.
Core Knowledge and Skills	• Know the stages of the fetch, decode, execute cycle. • Describe the Von Neumann architecture.	• The need for primary storage. • The difference between RAM and ROM.	• How to add two binary digits together and explain overflow errors that occur. • The use of binary codes	• Factors that affect the performance of networks. • The internet as a worldwide collection of	• Features of a user interface. • The purpose and functionality of utility software.	• Knowledge of a variety of examples of digital technology and how this impacts on society. • An ability to discuss the

	● Know what is meant by the term ‘embedded system’. ● Examples of embedded systems.	● The purpose of ROM in a computer system ● The purpose of RAM in a computer system. ● Virtual memory ● Common types of storage: optical, magnetic, solid state.	to represent characters. ● The term ‘character set’. ● How an image is represented as a series of pixels, represented in binary. ● Types of compression: lossy vs lossless.	computer networks: DNS, hosting, the cloud, web servers and clients. ● Star and mesh network topologies. ● Modes of connection: wired (ethernet) and wireless (Wi-Fi and Bluetooth). ● Understand threats to computer systems and networks such as: malware, social engineering and brute-force attacks.	● User management functions e.g. allocation of an account, access rights and security. ● File management and the key features e.g. naming, allocating to folders, moving files, saving etc.	impact of technology on wider society including: ethical, legal, cultural, environmental and privacy issues. ● The need to licence software and the purpose of a software licence.
Developmental Knowledge and Skills	● Know what the registers in the CPU are. ● Know the components of the Von Neumann architecture e.g. ALU, CU, Cache, Registers. ● Know the other components of the CPU e.g. MAR, MDR, program counter, accumulator. ● Know several examples of embedded systems.	● The need for secondary storage. ● Suitable storage devices and storage media for a given application. ● How data needs to be converted into a binary format to be processed by a computer.	● How to convert denary numbers into 2-digit hexadecimal numbers and vice versa. ● The relationship between the number of bits per character in a character set, and the number of characters that can be represented e.g. ASCII, Unicode. ● The effect of colour depth and resolution on: the quality of the image and the size of an image file.	● The different roles of computers in a client-server and peer-to-peer network. ● The hardware needed to connect stand-alone computers into a Local Area Network: wireless access points, routers, switches, NIC, transmission media. ● Identifying and preventing vulnerabilities using methods such as: anti-malware software, firewalls, passwords, physical security.	● Memory management, e.g. the transfer of data between memory, and how this allows for multitasking. ● Understand that data is passed between devices and the processor and this process needs to be managed.	● The purpose of each piece of legislation (Data Protection Act 2018, Computer Misuse Act 1990, Copyright Designs and Patents Act 1988 and software licences (i.e. open source proprietary) and the specific actions it allows or prohibits.

Complex Knowledge	● How common characteristics of the CPUs affect their performance: clock speed, cache size, number of cores. ● Understand how to program. ● Know how GPU's and CPU's help process images.	● The advantages and disadvantages of different storage devices and storage media relating to these characteristics: capacity, speed, portability, durability, reliability and cost. ● Data capacity and calculation of data capacity requirements.	● Binary shifts ● Metadata ● How sound can be sampled and stored in digital form. ● The effect of sample rate, duration and bit depth on: playback quality and the size of a sound file.	● Encryption ● IP addressing and MAC addressing. ● Standards ● Common protocols including: TCP/IP, HTTP, HTTPS, FTP, POP, IMAP, SMTP. ● The concept of layers. ● Identify and prevent vulnerabilities using more complex methods such as: penetration testing, user access levels and encryption.	● Understand that computers often comes with utility software and how this performs housekeeping tasks. ● Purpose of the identified utility software and why it is required.	● Features of open source (providing access to the source code and the ability to change the software). ● Features of proprietary (no access to the source code, purchased commonly as off-the-shelf). ● Recommend a type of licence for a given scenario including benefits and drawbacks.
Literacy (including reading)	● Use of CGP revision guides including the exam style question book for 'do now' tasks. ● Use of the textbooks to encourage reading and comprehension. ● Key vocab given and written in exercise books every lesson.	● Use of CGP revision guides including the exam style question book for 'do now' tasks. ● Use of the textbooks to encourage reading and comprehension.	● Use of CGP revision guides including the exam style question book for 'do now' tasks. ● Use of the textbooks to encourage reading and comprehension. ● Creating revision resources in the form of presentations to be given to the rest of the class.	● Research tasks to be completed by searching for and reading different sources on the Internet. ● Reading news articles and case studies about issues to do with network security.	● Use of CGP revision guides including the exam style question book for 'do now' tasks. ● Use of the textbooks to encourage reading and comprehension.	● Research tasks to be completed by searching for and reading different sources on the Internet. ● Reading news articles about ethical, legal, cultural and environmental issues to do with Computer Science.
Cultural Capital	● Knowledge of the Von Neumann architecture for the CPU.	● Knowledge of data storage techniques.	● Understanding how digital imagery can be constructed out of simple data.	● Knowledge that the world wide web is a collection of interconnected networks.	● Understanding the need for file management systems.	● Ethical, legal, cultural and environmental issues with technology across the UK.

Social, Moral, Spiritual and Cultural Development	● Knowledge of the use of embedded systems across every area of society from inside the home to work and leisure activities.	● Knowledge of the need for certain types of storage particularly password protected or encrypted storage and what sectors require this level of data protection.	● Knowledge of the need for certain types of storage particularly password protected or encrypted storage and what sectors require this level of data protection.	● Knowledge of the reasons why networks are accessed by unauthorised users.	● Issues with inclusivity in programming, making sure that operating systems developed by companies are accessible for every type of user.	● Learning about collaborating with others, sharing and improving open-source code for software. ● Learning about the impact of technology on wider society.
Fundamental British Values	● The history of modern computing, learning about Von Neumann vs. Harvard architecture.	● Teamwork, mutual respect and tolerance built through creating revision resources in the form of presentations.	● Teamwork, mutual respect and tolerance built through creating revision resources in the form of presentations.	● Rule of law, students will gain knowledge of the ways networks can be accessed by unauthorised users and what protections we can put in place to safeguard this.	● Respect for others.	● Rule of law, students will gain knowledge of the legislation that monitors computer use in the UK.
Assessment	● Assessments are made up of real past-paper hand written exam questions as this is key practice for their real exam at the end of Year 11 (45 marks).	● Assessments are made up of real past-paper hand written exam questions as this is key practice for their real exam at the end of Year 11 (45 marks).	● Assessments are made up of real past-paper hand written exam questions as this is key practice for their real exam at the end of Year 11 (45 marks).	● Assessments are made up of real past-paper hand written exam questions as this is key practice for their real exam at the end of Year 11 (45 marks).	● Assessments are made up of real past-paper hand written exam questions as this is key practice for their real exam at the end of Year 11 (45 marks).	● Assessments are made up of real past-paper hand written exam questions as this is key practice for their real exam at the end of Year 11 (45 marks).

Key Stage Curriculum Map:

The curriculum in Computer Science and ICT will ensure that students will be able to use two or more programming languages, use logical reasoning, model the state and behaviour of real-world problems and understand a range of ways to use technology safely, respectfully, responsibly and securely.

THE YEAR 11 CURRICULUM MAP

	HALF TERM 1	HALF TERM 2	HALF TERM 3	HALF TERM 4-6
Topic and Learning Focus	**Algorithms & Boolean Logic**	**Programming Fundamentals**	**Producing Robust Programs**	**Revision & Exams**
Foundational Knowledge Prior learning needed	● To be able to identify the inputs, processes and outputs for a problem. ● Recognition of each logic gate symbol.	● Knowledge that different types of programming languages exist.	● Understanding there are issues facing programmers in terms of creating robust designs.	● The fetch decode execute cycle. ● Understand what the CPU of a computer does. ● Be able to describe what RAM and ROM are. ● The units of data storage. ● Converting denary numbers into binary numbers and vice versa. ● The need for compression. ● Know the types of networks: LAN and WAN. ● Types of illegal activity online such as phishing and social engineering. ● Recognition of each logic gate symbol.
Core Knowledge and Skills	● Principles of computational thinking: abstraction, decomposition, algorithmic thinking. ● Structure diagrams. ● Searching algorithms: binary and linear searches. ● Sorting algorithms: bubble, merge and insertion sorts. ● Simple logic diagrams	● The use of variables, constants, operators, inputs, outputs and assignments. ● The use of three basic programming constructs used to control the flow of a program: sequence, selection, iteration. ● The use of common arithmetic operators. ● The use of records to	● Defensive design considerations: anticipating misuse and authentication. ● Input validation. ● The purpose of testing. ● Types of testing: final and iterative. ● Understanding the issues a programmer should consider to ensure that a program caters for all likely input	● The difference between RAM, ROM & virtual memory. ● Common types of storage: optical, magnetic, solid state. ● Binary addition, image and sound composition and character sets. ● Types of compression: lossy vs lossless. ● The domain name system, hosting, the cloud, web servers and clients. ● Star and mesh network topologies and network performance. ● Understand threats to networks such as: malware, social engineering and brute-force attacks. ● User management functions e.g. allocation of an account, access rights and security. ● An ability to discuss the impact of technology on wider society including: ethical, legal, cultural, environmental and privacy issues. ● Searching algorithms: binary and linear searches.

	using the operators AND, OR and NOT.	store data.	values.	● Sorting algorithms: bubble, merge and insertion sorts.
Developmental Knowledge and Skills	● Create, interpret, correct, complete and refine algorithms using flowcharts. ● Identify common errors. ● Knowledge of the truth tables for each logic gate. ● Combining boolean operators using AND, OR and NOT. ● Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios.	● The use of common boolean operators AND, OR and NOT. ● The use of data types: integer, real, boolean, character and string, casting. ● The use of basic string manipulation. ● The use of SQL to search for data. ● How to use subprograms to produce structured code.	● Maintainability: use of sub programs,naming conventions, indentation and commenting. ● Understand why commenting is useful and apply this appropriately. ● Ability to identify suitable test data for a given scenario.	● Know the components of the Von Neumann architecture e.g. ALU, CU, Cache, Registers. ● Know the other components of the CPU e.g. MAR, MDR, program counter, accumulator. ● Know several examples of embedded systems and the need for secondary storage. ● Suitable storage devices and storage media for a given application. ● How data needs to be converted into a binary format to be processed by a computer. ● How to convert denary numbers into 2-digit hexadecimal numbers and vice versa. ● Character sets, colour depth and resolution. ● The different roles of computers in a client-server and peer-to-peer network. ● The hardware needed to connect stand-alone computers into a Local Area Network: wireless access points, routers, switches, NIC, transmission media. ● Identifying and preventing vulnerabilities using methods such as: anti-malware software, firewalls, passwords, physical security. ● Memory management, e.g. the transfer of data between memory, and how this allows for multitasking. ● Legislation and ethical issues.
Complex Knowledge	● Create, interpret, correct, complete and refine algorithms using pseudocode, reference language and high-level programming languages. ● Trace tables. ● Applying logical operators in truth tables to solve problems.	● The use of basic file handling operations: open, read, write, close. ● The use of arrays when solving problems, including both one-dimensional and two-dimensional arrays. ● Random number generation.	● Selecting and using suitable test data: norma, boundary, invalid/erroneous. ● Refining algorithms. ● Ability to create/complete a test plan.	● How common characteristics of the CPUs affect their performance: clock speed, cache size, number of cores. ● The advantages and disadvantages of different storage devices and storage media relating to these characteristics: capacity, speed, portability, durability, reliability and cost. ● Data capacity and calculation of data capacity requirements. ● Binary shifts, metadata, sound files, encryption. ● IP addressing and MAC addressing. ● Standards and common protocols including: TCP/IP, HTTP, HTTPS, FTP, POP, IMAP, SMTP. ● Identifying and preventing vulnerabilities. ● Purpose of utility software and why it is required.

🟥				● Features of open source and proprietary software and software licensing.
Literacy (including reading)	● Use of CGP revision guides including the exam style question book for ‘do now’ tasks. ● Use of the textbooks to encourage reading and comprehension.	● Key terminology learnt with the Python workbooks. ● New key terminology learnt relating to programming, Python in particular.	● Key terminology learnt with the Python workbooks. ● New key terminology learnt relating to programming, Python in particular.	● Flash card making. ● Regular use of the revision guides and online revision resources. ● Regular mock exam question practice. ● Wider reading into current news to keep to to date on ethical, legal, cultural and privacy issues to do with Computer Science.
Cultural Capital	● Knowledge of specific circuitry inside computers and how alterations to it alter the function of the computer.	● Knowledge of common programming techniques.	● Knowledge of robust programming constructs and techniques.	● An overall appreciation of computing as a global technology that has the power to change the world and has already changed it significantly.
Social, Moral, Spiritual and Cultural Development	● Knowledge of algorithmic bias, for example: the writer of an algorithm being biassed against a certain part of society.	● Knowledge of ethical issues to do with programming such as questionable personal data ownership and addictive design.	● Knowledge of ethical issues such as defensive design considerations e.g. ensuring that misuse of a system can be prevented.	● Appreciation of the impact of technology on wider society including ethical, legal, cultural, environmental and privacy issues.
Fundamental British Values	● Teamwork, mutual respect and tolerance built through solving real-world problems with the use of algorithms.	● Teamwork, mutual respect and tolerance built through paired programming.	● Teamwork, mutual respect and tolerance built through paired programming.	● Rule of law, students will gain knowledge of the legislation that monitors computer use in the UK. ● Teamwork, mutual respect and tolerance built through creating revision resources in the form of presentations.
Assessment	● End of topic test, a 45 mark past questions paper. Students will need to write algorithms to solve	● Mock Exams ● End of topic test, a 20 mark past paper with questions about programming.	● End of topic test, a 20 mark past paper with questions about programming. ● Practical programming	● Assessments created using past paper questions. ● Regular short assessments in class with feedback given in the lesson. Use of the CGP exam question revision guides.

	problems and also trace algorithms.	● Practical programming task, 25 marks, students must choose one of three tasks to program using Python.	task, 25 marks, students must choose one of three tasks to program using Python.	