Computing	Autumn Term		Spring term		Summer term	
Year 7	1	2	3	4	5	6
Topic Summary	**Digital literacy/ Clear Messaging in Digital Media**	**Networks - from semaphore to the Internet**	**Programming essential 1**	**Introduction to Spreadsheets**	**Using Media**	**Programming II**
Thinking Hard	**Acquiring knowledge / curiosity**: This unit has been devised as a transitional unit to allow learners to confidently move from Year 6 to Year 7. By the end of the unit, they should also be able to use the school network safely and respectfully. Using new systems responsibly, What is a network? How do you keep them secure? Where is the cloud? **Mindfulness**: Do you change who you are depending on who you are talking to? *Creating independence:* If you find it on the Internet, can you use it? (copyright)	**Acquiring knowledge/curiosity**: This unit progresses students' knowledge and understanding of networks and associated hardware. The unit will establish a foundation understanding of how data is transmitted across networks, as well as exploring the factors that can affect performance. **Mastery** How does data move through a network? What is a protocol? **World citizen**: Explain the term 'connectivity' as the capacity for connected devices ('internet of things') to collect and share information about me with or without my knowledge (including microphones, cameras and geolocation). Describe how internet-connected devices can affect me.	**Acquiring knowledge/numerac**y: This unit focuses on the development of the following key techniques: Sequencing Variables Selection Operators Count-controlled iteration **Mastery:** Independent problem-solving skills demonstrated **Self-assurance**: Is there any problem a computer can't solve? **Creativity**: Is Coding the most creative skill there is?	**Acquiring knowledge/curiosit**y: This unit progresses learners' knowledge and understanding of modelling data using a spreadsheet.? What applications do they have in real life? What's a cell/table/column/field/record? Using basic formulas, using conditional formatting. Filtering & sorting data. **Mastery**: Complex functions and formulas	**Acquiring knowledge/ curiosity:** During this unit, learners develop their understanding of information technology and digital literacy skills. They will use the skills learnt across the unit to create a blog post about a real-world cause that they would like to gain support for. Learners will develop software formatting skills and explore concerns surrounding the use of other people's work, including licensing and legal issues **Mastery:** Fully publish accurate blogpost	**Grit/curiosity**: Why do we need subprograms? Does efficiency matter? **Acquiring knowledge: t**he unit requires learners to complete a set of tasks using a Scratch program and focuses on the development of the following key techniques: Sequencing Variables Selection Operators Count-controlled iteration
Developing Character	**Mindfulness and self-control:** Online responsibility & staying secure online ***Grit/optimism:*** *Collaborating online in lessons.* ***Self control*** *- Using the school LAN* ***Self assurance:*** *Presenting to an audience - can you adjust how you speak depending on who you are talking to?*	**Mindfulness** - how much time do you spend on the Internet? How long could you go tech free? **Curiosity/gratitude**: Imagine a world without computer networks, how different would your life would be. (Lesson 1) **Self assurance/awareness/optimism**: Do you rely on the Internet? What would lockdown be like without the Internet?	**Mindfulness/grit** - Developing stickability, debugging your code. **Self-assurance/curiosity** - PRIMM - predicting outcomes - **Independence/SA/Creativity** - creating your own program Helping others, paired programming **Grit**: How easily do you give up? Can you predict outcomes?	**Grit** - writing your own formulas **Self awareness/self assurance/curiosity**: How could you use a spreadsheet in your life? **Mastery**: Why use a spreadsheet instead of a calculator?	Mindfulness & Grit/not fearing failure - developing stickability, debugging your code. **Self-assurance** - PRIMM - predicting outcomes **Independence/SA/Creativity/Curiosity** - creating your own program **Self-assurance/awareness/kindness** - helping others, paired programming. **Grit**: Do you have great stickability? Not fearing failure/ **kindness**: Does it matter if someone else can or can't read your code? (maintainability)	Mindfulness & Grit/not fearing failure - developing stickability, debugging your code. **Self-assurance** - PRIMM - predicting outcomes **Independence/SA/Creativity/Curiosity** - creating your own program **Self-assurance/awareness/kindness** - helping others, paired programming. **Grit**: Do you have great stickability? Not fearing failure/ **kindness**: Does it matter if someone else can or can't read your code? (maintainability)
Understanding Diversity	**Understanding environmental diversity/respecting human rights:** What is the Digital divide? Local v global level of DD - empathy **Understanding mental and physical diversity/Optimism**: Anti-bullying and understanding people with disabilities-video with Prince William Hidden figures. How has life changed? Excerpt from the book as well as a video clip.	**Mastery** - Not all computers are expensive. How can everyone access technology. How could you use a Microbit to solve a problem? **Kindness/Gratitude**: Digital divide - local v global. Should you pay for the Internet	**Mastery/world citizen**: Are all computers the same? Are they all expensive? Can you solve a problem with a microbit? Can anyone write code? HLLs Should you understand how your computer works?	**Self assurance/kindness**: Different programming languages for different purposes, ages, experiences. Open Source v proprietary software	**Understanding environmental diversity/respecting human rights:** Copyright and ownership I know that commercial online content can be viewed, accessed, or downloaded illegally. (11–14) I can accurately define the concept of plagiarism. (11–14) I can use this definition to evaluate my own use of online sources. (11–14) I understand the concept of software and content licensing. (11–14) I understand Creative Commons Licensing protocols. (11–14) I can identify the potential consequences of illegal access or downloading and how it may impact me and my immediate peers. (11–14) **Understanding mental and physical diversity/Optimism**: Managing online information on self for a positive digital world	**Mastery**: Humans v computers - why do we count in 10s? Are there other number bases? How would these work? **Kindness**: Debug someone else's code/paired programming
Literacy Reading, Oracy	Computer **literacy** and fluency - logging on to school's network, Google Cloud, Insight understanding the uses of each Do Now task - Read article on "Cyberbullying: Being Bullied Online and Advice on What to Do". Do Now task - "Hidden Figures" read (Charles Babbage - Inventor of first computer)	Literacy - keyword vocabulary, The importance of syntax; protocol	Literacy: Syntax in code= grammar Key programming vocabulary Oracy - COP 26 - describe a climate change issue we are facing Do now task - "What is a Digital Content Producer?"	Literacy - keyword vocabulary pertaining to spreadsheet (functions, conditional formatting, data validation. use of spell check, find and replace)	Literacy - keyword vocabulary, The importance of spelling punctuation and grammar when publishing your own writing	Sequential ordering of algorithms and programming - why is order important?
Gatsby, Careers	Do now task - Identify skills that can be developed/ learnt throughout computing and how they can be used and developed in future careers. Skills- Communication, creativity, presentation skills, IT software skills, research skills, data analysis	Work with network technicians. Students to come up with questions to ask them to get knowledge on how they run the school network.	Research job description, salary for careers where programming is used. -Computer Programmer -Digital Content Producer Do now task - read article on "What is a Digital Content Producer?", video interview with a DCP from Chichester University.	Careers - Spreadsheets in the world of work/Transferable skills. Interview Finance team. Data analysis, link to cross curricular - Sport performance analysis, research analysis	Careers - Influencers and digital marketing	Review skills they have previously learnt in other topics including programming one. What skills could they continue to develop in this topic. How will that help them in future careers
Mental and Physical Well-being	Privacy and security I can explain how my internet use is often monitored (e.g. by my school or internet service provider) (Y7)	Screen breaks - physical well-being	Mindful mountain	Mindfulness - online jigsaw in silence Physical well-being - create a spreadsheet which tracks healthy eating.	Mindfulness - Physical wellbeing blog post - About you	A mindful stroll Digital wellbeing and your Digital footprint - video
Cross-Curricular Links	PD (online safety & responsibility) Literacy- Comprehension, SPAG, Persuasive writing History - Hidden Figures **National curriculum links** - Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems	Physical Education link - Belaying Protocol (L1) **National Curriculum Links** - Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems	Maths - logic, variables, constants, problem-solving; Science/physics - connecting your Microbit **National curriculum links** 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 (e.g. lists, tables, or arrays); design and develop modular programs that use procedures or functions Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem	Maths - writing Mathematical formulae, BIDMAS, variables, constant, problem solving PE- analysis performance analysis data on a performer **National curriculum links** 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	Maths - logic, variables, constants, problem-solving; **National curriculum links** 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 Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability	Numeracy - binary conversions, adding binary numbers, different bases MFL - writing a translation program **National curriculum links** 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 (e.g. lists, tables, or arrays); design and develop modular programs that use procedures or functions. Understand several key algorithms that reflect

Extra-Curricular Links	Networks in your home. European day of languages - google translate the do now task. Respecting human rights/understanding democracy: Black History month - clips from Hidden Figures movie: Katherine Johnson, Dorothy Vaughan, Mary Jackson Careers in computing	Scratch - External Club running after School Nov Anti-Bullying Week Dec International day of the disabled person Human Rights Day - Do Now reading task and presentation to raise awareness of AIDS and HIV. STEM day - TBC	Scratch - External club running after school	World Day for Cultural Diversity - Digital Divide - countries where there is a lack of technology and access to the internet.	Scratch - External club running after school	Scratch - External club running after school
Specific Learning Endpoints	**What we want students to learn/be able to:** SLE 1.1 -Acquire skills to use computers around the school responsibly. SLE 1.2 - Recognise they are accessing a network and Cloud storage (Google Drive) Apply knowledge to access Classroom and Insight to track Home Learning. SLE 1.3 - Articulate the fundamentals of a computers system and define (with examples): computer, network, the cloud, input/output device, HW and SW. SLE 1.4 - Appraise effective presentations for a given audience SLE 1.5 - Recognise cyberbullying and analyse its effects SLE 1.6 - Differentiate between safe and unsafe online behaviour. SLE 1.7 - Identify key features of a good poster SLE 1.8 - Plan a poster to clearly convey a message SLE 1.9 - Choose and download a suitable image SLE 1 10 - Create a poster using a desktop publishing application SLE 1. 11Modify a logo using a graphic editing program SLE 1.12 - Choose how to combine text and graphics in a slide SLE 1.13 - Use digital tools to provide feedback on design choices SLE 1.14 - Plan a consistent layout for a set of slides SLE 1.15 - Modify a logo so that it fits in with the	**What we want students to learn/be able to:** SLE 2.1 - Define what a computer network is and explain how data is transmitted between computers across networks. SLE 2.2 - Define 'protocol' and provide examples of non-networking protocols SLE 2.3 - List examples of the hardware necessary for connecting devices to networks SLE 2.4 - Compare wired to wireless connections and list examples of specific technologies currently used to implement such SLE 2.5 connections SLE 2.5 - Define 'bandwidth', using the appropriate units for measuring the rate at which data is transmitted, and discuss familiar examples where bandwidth is important SLE 2.6 - Define what the internet is SLE 2.7 - Explain how data travels between computers across the internet SLE 2.8 - Describe key words such as 'protocols', 'packets', and 'addressing' SLE 2.9 - Explain the difference between the internet, its services, and the World Wide Web SLE 2.10 - Describe how services are provided over the internet SLE 2.11 - List some of these services and the context in which they are used SLE 2.12 - Explain the term 'connectivity' as the capacity for connected devices ('Internet of Things') to collect and share information about me with or without my knowledge (including microphones, cameras, and geolocation)	**What we want students to learn/be able to:** SLE 3.1 - Compare how humans and computers understand instructions (understand and carry out) SLE 3.2 - Define a sequence as instructions performed in order, with each executed in turn SLE 3.4 - Predict the outcome of a simple sequence SLE 3.5 - Modify a sequence SLE 3.6 - Define a variable as a name that refers to data being stored by the computer SLE 3.7 - Recognise that computers follow the control flow of input/process/output SLE 3.8 - Predict the outcome of a simple sequence that includes variables SLE 3.9 - Trace the values of variables within a sequence SLE 3.10 - Make a sequence that includes a variable SLE 3.11 - Define a condition as an expression that will be evaluated as either true or false SLE 3.12 - Identify that selection uses conditions to control the flow of a sequence SLE 3.13 - Identify where selection statements can be used in a program SLE 3.14 - Modify a program to include selection SLE 3.15 - Create conditions that use comparison operators (>,<,=) SLE 3.16 - Create conditions that use logic operators (and/or/not) SLE 3.17 - Identify where selection statements can be used in a program that include comparison and logical operators SLE 3.18 - Define iteration as a group of instructions that are repeatedly executed SLE 3.19 - Describe the need for iteration SLE 3.20 - Identify where count-controlled iteration can be	**What we want students to learn/be able to:** SLE - 4.1 - Identify columns, rows, cells, and cell references in spreadsheet software SLE 4.2 - Use formatting techniques in a spreadsheet SLE 4.3 - Use basic formulas with cell references to perform calculations in a spreadsheet (+, -, *, /) SLE 4.5 - Use the autofill tool to replicate cell data SLE 4.6 - Explain the difference between data and information SLE 4.7 - Explain the difference between primary and secondary sources of data Collect data SLE 4.8 - Analyse data SLE 4.9 - Create appropriate charts in a spreadsheet SLE 4.10 - Use the functions SUM, COUNTA, MAX, and MIN in a spreadsheet to Analyse data SLE 4.11 - Use a spreadsheet to sort and filter data SLE 4.12 - Use the functions AVERAGE, COUNTIF, and IF in a spreadsheet SLE 4.13 - Use conditional formatting in a spreadsheet SLE 4.14 - Apply all of the spreadsheet skills covered in this unit	**What we want students to learn/be able to:** SLE 5.1 - Select the most appropriate software to use to complete a task SLE 5.2 - Identify the key features of a word processor SLE 5.3 - Apply the key features of a word processor to format a document SLE 5.4 - Evaluate formatting techniques to understand why we format documents SLE 5.5 - Select appropriate images for a given context SLE 5.6 - Apply appropriate formatting techniques SLE 5.7 - Demonstrate an understanding of licensing issues involving online content by applying appropriate Creative Commons licences SLE 5.8 - Demonstrate the ability to credit the original source of an image SLE 5.9 - Critique digital content for credibility SLE 5.10 - Apply techniques to identify whether or not a source is credible SLE 5.11 - Apply referencing techniques and recognise the concept of plagiarism SLE 5.12 - Evaluate online sources for use in own work SLE 5.13 - Construct a blog using appropriate software SLE 5.14 - Create content for a blog based on credible sources SLE 5.15 - Apply referencing techniques that credit authors appropriately SLE 5.16 - Design the layout of the content to make it suitable for the audience SLE 5.17 - Construct a blog using appropriate software SLE 5.18 - Create content for a blog based on credible sources SLE 5.19 - Apply referencing techniques	**What we want students to learn/be able to:** SLE 6.1 - Compare how humans and computers understand instructions (understand and carry out) SLE 6.2 - Define a sequence as instructions performed in order, with each executed in turn SLE 6.3 - Predict the outcome of a simple sequence SLE 6.4 - Modify a sequence SLE 6.5 - Define a variable as a name that refers to data being stored by the computer SLE 6.6 - Recognise that computers follow the control flow of input/process/output SLE 6.7 - Predict the outcome of a simple sequence that includes variables SLE 6.8 - Trace the values of variables within a sequence SLE 6.9 - Make a sequence that includes a variable SLE 6.10 - Define a condition as an expression that will be evaluated as either true or false SLE 6.11 Identify that selection uses conditions to control the flow of a sequence SLE 6.12 - Identify where selection statements can be used in a program SLE 6.13 - Modify a program to include selection SLE 6.14 - Create conditions that use comparison operators (>,<,=) SLE 6.15 - Create conditions that use logic operators (and/or/not) SLE 6.16 Identify where selection statements can be used in a program that include comparison
Computing	**Autumn Term**		**Spring term**		**Summer term**	
Year 8	**1**	**2**	**3**	4	5	6
Topic Summary	**Computing Systems**	**Python - Intro to text-based languages**	**Representations - from clay to silicon**	**Mobile App Development**	**Vector Graphics**	**Developing for the web**
Thinking Hard	**Acquiring knowledge/curiosity:** This unit takes learners on a tour through the different layers of computing systems: from programs and the operating system, to the physical components that store and execute these programs, to the fundamental binary building blocks that these components consist of. **Mastery**: What is a computer system? What's with the 1s and 0s? What's the future of AI? NC links: 3.4, 3.5, 3.6	**Acquiring knowledge/curiosity**: This unit introduces students to text-based programming with Python. The lessons form a journey that starts with simple programs involving input and output, and gradually moves on through arithmetic operations, randomness, selection, and iteration. **Mastery**: What is logic? What is syntax? Why iterate? What are the different ways to iterate?	**Acquiring knowledge/curiosity**: This unit conveys essential knowledge relating to binary representations. The activities gradually introduce students to binary digits and how they can be used to represent text and numbers. **Mastery**: Why do we need bits & bytes?	**Acquiring knowledge/curiosity**: this unit aims to take the students from designer to project manager to developer in order to create their own mobile app. Using App Lab from code.org, learners will familiarise themselves with the coding environment and have an opportunity to build on the programming concepts they used in previous units before undertaking their project. Students develop the following key techniques: Event handling Sequencing Variables Selection Operators **Mastery**: Why do we need bits & bytes?	**Acquiring knowledge/curiosity:** This unit offers students the opportunity to design graphics using vector graphic editing software. By the end of the unit learners will have produced an illustration, a logo, or some icons using vector graphics.	**Acquiring knowledge/Curiosity**: In this unit, students will explore the technologies that make up the internet and World Wide Web. Starting with an exploration of the building blocks of the World Wide Web, HTML, and CSS, learners will investigate how websites are catalogued and organised for effective retrieval using search engines. key areas of networks: Searching Threats HTML and CSS **Mastery**: Students will have a fully functioning website
Developing Character	**Self Assurance/optimism:** AI - what are the applications of that make/would make the world a better place? **SA/curiosity**: Turing Test - could a computer convince you it's human? Can a computer provide friendship? **Being a world citizen/kindness**: Can you do what you want with software? Who owns it?	**Creativity/Curiosity** - making a quiz in Python **Grit** - debugging code **Self-assurance/ kindness**- paired programming	**Mastery/creating independence**: Moore's law: How much storage do you need? Can we keep expanding storage capacity and processing power? What are the consequences of this?	**Mastery/creating independence**: Independently problem solve to develop project	**Mastery/creating independence**: Independently problem solve to design and manipulate vector graphic.	**Mindfulness/being a world citizen/awareness of where you live:** Students consider the effects of our consumption of technology on the environment. Where does your e-waste go? Who is responsible for e-waste? **Respecting human rights/understanding democracy**: poor working conditions..
Understanding Diversity	**Respecting human rights/literacy**: Black History Month: Do Now Task - reading comprehension task: "7 Famous Black Computing Pioneers"	**Self assurance / kindness**: Different strengths. Which challenges can you complete? Peer teaching opportunity. Research Alan Turing and the Turing Test - talk about Turing being persecuted for being gay.	**Self assurance/being a world citizen**: Ascii v Unicode - the need for character sets that represent all languages	**Self assurance/being a world citizen**: Use of universal programming language	**Self assurance/being a world citizen**: Use of universal keywords for designing vector graphics	**Being a world citizen/PD**: Digital divide lack of internet in countries and poor connectivity **Curiosity/respecting human rights:** Access to knowledge and public services **Understanding environmental diversity**: What happens when resources run out? Does tech create more problems than it solves?
Literacy Reading, Oracy	Students read passage about Antikythera Mechanism and decide if it is a computer. Black History Month: Do Now Task - reading comprehension task: "7 Famous Black Computing Pioneers" Do Now task - read an article on "FOMO" and relate it to social media and online life.	Syntax - debugging your own programs. Talking your program aloud to find your bugs. Using PRIMM to debug your own programs	Oracy - explain the need for Unicode	Literacy - in programming language in a block based programming language	Literacy and keywords for learning vector graphics	Literacy - Building blocks for the work wide web. Keywords and terminology

Gatsby, Careers	Careers BBC Roadshow	Careers/Options - Is CS for you? Research Career of Computer Programmer as a Do Now to go alongside our intro to Python programming.	How will technology play a role in your career? Discussion.	Development of careers in digital marketing and mobile app development	How will these skills help you in the future for employment discussion	Research website design companies. Could this be a career to aspire to?
Mental and Physical Well-being	FOMO - Fear of Missing Out - how does FOMO affect your screen-time?	The data self World Aids Day Research Alan Turing and the Turing Test - talk about Turing being persecuted for being gay.	Keeping up appearances - the selfie v yourself	Take rest periods every 20 minutes to avoid eye strain	Mindfulness - online design in silence	Mindfulness - Respect individuals and be responsible what is posted online via websites
Cross-Curricular Links	Maths - binary and place values, logical (Boolean) operators, arithmetic operators Science - circuits/logic, variables, switches, transistors **National curriculum links -** Can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems. Subject content understand simple Boolean logic [for example, AND, OR and NOT] 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 with other systems Understand how instructions are stored and executed within a computer system.	Maths (variables, constants), problem solving, logical operators <,>). Numeracy - calculations in programs/arithmetic expressions Curriculum links **National curriculum links** Can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation Can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem Understand how instructions are stored and executed within a computer system Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems	History - the development of character sets from ASCII to Unicode. Art - the advancement of images in video games from 8 bit and up Maths - different number bases, place values, comparing with denary. **National curriculum links** Understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits	Art - the advancement of images in mobile apps Maths - different number bases, variables and sequencing. **National curriculum links** National curriculum links 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, 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 several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability	Art - 3D Design graphics and shapes Maths - shape and space. **National curriculum links** 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 create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability	Maths - logical operators PD - the environment **National curriculum links** Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability.
Extra-Curricular Links	Sept European day of languages - google translate the do now task. Black History month Y8 - Investigate 7 Black computing pioneers, who was the most influential? Self control- what's your moral code - software piracy	Scratch - external competition running as after school club. Nov Anti-Bullying Week Dec International day of the disabled person Human Rights Day	Creating ambition/not fearing failure: The Turing Cryptology Competition Scatch - External after School club January - National Technology Day Feb - Safer Internet Day - R Time activities, all year groups."	Scratch - External after school club	Scratch - External after school club	Scratch - External after school club
Specific Learning Endpoints	**What we want students to learn/be able to:** SLE 7.1 - Recall that a general-purpose computing system is a device for executing programs SLE 7.2 - Recall that a program is a sequence of instructions that specify operations that are to be performed on data SLE 7.3 - Explain the difference between a general-purpose computing system and a purpose-built device SLE 7.4 - Describe the function of the hardware components used in computing systems SLE 7.5 - Describe how the hardware components used in computing systems work together in order to execute programs SLE 7.6 - Recall that all computing systems, regardless of form, have a similar structure ('architecture') SLE 7.7 - Analyse how the hardware components used in computing systems work together in order to execute programs SLE 7.8 - Define what an operating system is, and recall its role in controlling program execution SLE 7.9 - Describe the NOT, AND, and OR logical operators, and how they are used to form logical expressions SLE 7.10 - Use logic gates to construct logic circuits, and associate these with logical operators and expressions SLE 7.11 - Describe how hardware is built out of increasingly complex logic circuits SLE 7.12 - Recall that, since hardware is built out	**What we want students to learn/be able to:** SLE 8.1 - Describe what algorithms and programs are and how they differ SLE 8.2 - Recall that a program written in a programming language needs to be translated in order to be executed by a machine SLE 8.3 - Write simple Python programs that display messages, assign values to variables, and receive keyboard input SLE 8.4 - Locate and correct common syntax errors SLE 8.5 - Describe the semantics of assignment statements SLE 8.6 - Use simple arithmetic expressions in assignment statements to calculate values SLE 8.7 - Receive input from the keyboard and convert it to a numerical value SLE 8.8 - Use relational operators to form logical expressions SLE 8.9 - Use binary selection (if, else statements) to control the flow of program execution SLE 8.10 - Generate and use random integers SLE 8.11 - Use multi-branch selection (if, elif, else statements) to control the flow of program execution SLE 8.12 - Describe how iteration (while statements) controls the flow of program execution SLE 8.13 - Use iteration (while loops) to control the flow of program execution SLE 8.14 - Use variables as counters in iterative	**What we want students to learn/be able to:** SLE 9.1 - List examples of representations SLE 9.2 - Recall that representations are used to store, communicate, and process information SLE 9.4 - Provide examples of how different representations are appropriate for different tasks SLE 9.5 - Recall that characters can be represented as sequences of symbols and list examples of character coding schemes SLE 9.6 - Measure the length of a representation as the number of symbols that it contains SLE 9.7 - Provide examples of how symbols are carried on physical media SLE 9.8 - Explain what binary digits (bits) are, in terms of familiar symbols such as digits or letters SLE 9.9 - Measure the size or length of a sequence of bits as the number of binary digits that it contains SLE 9.10 - Describe how natural numbers are represented as sequences of binary digits SLE 9.11 - Convert a decimal number to binary and vice versa SLE 9.12 - Convert between different units and multiples of representation size SLE 9.13 - Provide examples of the different ways that binary digits are physically represented in digital devices	**What we want students to learn/be able to :** SLE 10.1 - Identify when a problem needs to be broken down SLE 10.2 - Implement and customise GUI elements to meet the needs of the user SLE 10.3 - Recognise that events can control the flow of a program SLE 10.4 - Use user input in an event-driven programming environment SLE 10.5 - Use variables in an event-driven programming environment SLE 10.4 - Develop a partially complete application to include additional functionality SLE 10.5 - Identify and fix common coding errors SLE 106 - Pass the value of a variable into an object SLE 10.7 - stablish user needs when completing a creative project SLE 10.8 - Apply decomposition to break down a large problem into more manageable steps SLE 10.9 - Use user input in a block-based programming language SLE 10.10 - Use a block-based programming language to create a sequence SLE 10.11 - Use variables in a block-based programming language SLE 10.12 - Use a block-based programming language to include sequencing and selection SLE 10.13 - Use user input in a block-based programming language	**What we want students to learn/be able to:** SLE 11.1 - Use tools to draw and modify shapes SLE 11.2 - Change the position and rotation shapes SLE 11.3 - Explain how z-order determines what is visible SLE 11.4 - Use tools to align and distribute objects to create uniformity SLE 11.5 - Explain how grouping can be used to work with several objects at once SLE 11.6 - Combine two shapes using union, intersection, and difference SLE 11.7 - Explain that vector graphics are made up of paths SLE 11.8 - Create and modify straight and curved paths SLE 11.9 - Change shapes to paths and edit them SLE 11.10 - Choose a project and plan a design SLE 11.11 - Combine tools and techniques to create a vector image SLE 11.12 -Evaluate the project against its given purpose SLE 11.13 - Explain how mark-up defines what a vector graphic looks like SLE 11.14 - Change an object by modifying its mark-up SLE 11.15 - Plan improvements and implement them to develop a project SLE 11.15 - Explain key differences between vector and bitmap images SLE 11.16 - Outline which image type best suits which uses SLE 11.17Evaluate their image against a rubric	**What we want students to learn/be able to:** SLE 12.1 - Describe what HTML is SLE 12.2 - Use HTML to structure static web pages SLE 12.3 - Modify HTML tags using inline styling to improve the appearance of web pages SLE 12.4 - Display images within a web page SLE 12.5 - Apply HTML tags to construct a web page structure from a provided design SLE 12.6 - Describe what CSS is SLE 12.7 - Use CSS to style static web pages SLE 12.8 - Assess the benefits of using CSS to style pages instead of in-line formatting SLE 12.9 - Describe what a search engine is SLE 12.10 - Explain how search engines 'crawl' through the World Wide Web and how they select and rank results SLE 12.11 - Analyse how search engines select and rank results when searches are made SLE 12.12 - Use search technologies effectively SLE 12.13 - Discuss the impact of search technologies and the issues that arise by the way they function and the way they are used SLE 12.14 - Create hyperlinks to allow users to navigate between multiple web pages SLE 12.15 - Implement navigation to complete a functioning website Complete summative assessment
Computer Science	Autumn Term		Spring term		Summer term	
Year 9	1	2	3	4	5	6
Topic Summary	Cyber Security / Python Revisited	Data Representation - Going audio visual	**The CPU - Von Neumann & Beyond / HTML ***	**Cybersecurity**	**Data Science**	**Networks / Python - functions & procedures ***

Thinking Hard	**Acquiring knowledge/curiosity**: This unit introduces learners to how data can be represented and processed in sequences, such as lists and strings. Why use subprograms? What is program flow? **Mastery:** What is the difference between data and information?	**Acquiring knowledge/creating independence**: In this unit, students will focus on digital media such as images and sounds, and discover the binary digits that lie beneath these types of media. What are different number bases? Why are they needed? How can you represent text/images/sound/video with just 1s and 0s? How can you convert analogue to digital? How can you represent colours? How many colours in a photograph? Why compress? What's the difference between lossy and lossless compression? Being creative: Can you create you own images with code?	**Acquiring knowledge / curiosity**: What is going on in the CPU? How does the CPU communicate with other hardware & software? What's behind a website? How would you attack a website?	**Acquiring knowledge / curiosity**: This unit takes the students on an eye-opening journey of discovery about techniques used by cybercriminals to steal data, disrupt systems, and infiltrate networks.	**Acquiring knowledge / curiosity**: In this unit, learners will be introduced to data science, and by the end of the unit they will be empowered by knowing how to use data to investigate problems and make changes to the world around them.	**Acquiring knowledge**/How do you build a network? How many networks do you use? Why do we need protocols? **Changing the world:** What happens if the Internet goes down. Permanently. Not fearing failure/creating independence/mastery: developing code independently
Developing Character	**Self assurance** - Taking part in international computational thinking competition. (Bebras) **Grit**: Students encounter realistic problems : solar system planets, book texts, capital cities, leaked passwords, word dictionaries, ECG data. **Self assurance / teamwork**: Cyber Discovery competition **Self control:** Why hack? Is it ever OK to hack?? **Optimism** - debug your code	**Self assurance** - Taking part in international computational thinking competition. (Bebras) **Kindness** - can you explain number bases to someone else?	**Self control/Being a world citizen**: Websites - revisit digital artefacts/copyright. Can you use it? Should you? **Mindfulness**: How can you fit a billion transistors into a single chip?	**Self control/Being a world citizen**: Computer Misuse Act and Fraud Act **Mindfulness**: Be a good digital citizen	**Grit:** Demonstrate determination by analysing complex data and drawing conclusions	**Grit/self-assurance** - developing code independently Considering secondary storage - cost v capacity & performance. Protocols - what protocols are there in society? What new protocols appeared during lockdown Which have we adopted? **Curiosity**: What's in a data packet? **Mindfulness**: Can you imagine a world without computers?
Understanding Diversity	**Acquiring cultural capital:** Does privacy still exist? **Respecting human rights:** Who has your data? Do you mind? What about sensitive data? Is it OK to collect data on race, gender, sexuality, appearance, disability **Awareness of where you live**: How can code solve real-life problems? **Not fearing failure:** Will you code work first time?	**Understanding mental and physical diversity:** Which methods are you using to solve the conversions? Why can't your computer count?	**Understanding democracy**: Are all computers big & expensive? Making the abstract concrete: Demystifying the CPU Being a world citizen/awareness of where you live: E-waste - where have all the hard drives gone?	**Understanding democracy**: Data protection act and the right to keep information secure? Consequences of breaking the law **Being a world citizen/awareness** of where you live: How to protect yourself and your data on the internet. Manage security software (e.g. anti-virus, security patches, adware blockers) on my devices and understand why regular updates are important	**Understanding democracy**: Freedom of information Act **Being a world citizen: -** Analyse data and global trends that effect citizens such as the environment	**Respecting human rights** - How do you keep a network it secure?
Literacy Reading, Oracy	**Oracy** - Talking the bugs out of your programs Development of literacy: Syntax, debugging (compare to proofreading), importance of accuracy in code. **Reading** - the dark web,. Cyber Intelligence Officer	Write a leaflet to explain clearly how to convert into and out of binary.	Trip to Tbc	Literacy - Compromising and presentation skills	Literacy - Comprehension and report writing	Client-server v peer-to-peer- silent reading comprehension.
Gatsby, Careers	Cyber Intelligence Officer - do you love to hack? P20 of CS careers booklet, reading.	Careers - Graphic Designer, Video Editor, Music Producer	Careers - Key figures in Computing - what was their background? How did they get there.	Careers in cyber security and GCHQ	Recognise Careers that will have analysts as job roles and the importance of these roles to the business	What do our network technicians do?
Mental and Physical Well-being	**Phishing quiz**	Digital identities	Digital well-being: Log your screen time	Digital well-being: Using computers ethically and legally	Digital well-being: Select data that does not overwhelm you.	Always being on - does it affect our friendships?
Cross-Curricular Links	Maths - Numerical Data **National curriculum links** Can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms, and data representation Can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems Subject content Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems Understand how instructions are stored and executed within a computer system Understand several key algorithms that reflect computational thinking; 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	Art - digital images/Photoshop Music - Audacity, creating and saving digital music Media - video editing/compression Numeracy - different units and bases **National curriculum links** Understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits	Science - circuits, transistors, logic gates Maths - logic, binary	Business - Legislation impacting business **National curriculum links** 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	Maths - Data and statistics, graphs Business - Data and reports **National curriculum links** 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	PD - protocols in society
Extra-Curricular Links	Sept European day of languages - google translate the do now task. Python - create a multiple choice languages quiz Black History Month - Y9 - Choose one influential black programmers to research and present (oracy),	Scratch - External afterschool club Nov STEM day tbc Nov Anti-Bullying Week Dec International day of the disabled person Human Rights Day	Jan - National Technology Day Feb - Safer Internet Day - R Time activities, all year groups. Scratch = External after school club	Scratch - External after school club	Scratch - External after school club	Scratch - External after school club

Specific Learning Endpoints	**What we want the students to learn/be able to:** SLE - 13.1 - Write programs that display messages, receive keyboard input, and use simple arithmetic expressions in assignment statements SLE 13.2 - Use selection (if-elif-else statements) to control the flow of program execution SLE 13.3 - Locate and correct common syntax errors SLE 13.4 - Create lists and access individual list items SLE 13.5 - Perform common operations on lists or individual items SLE 13.6 - Use iteration (while statements) to control the flow of program execution SLE 13.7 - Perform common operations on lists or individual items SLE 13.8 - Perform common operations on strings or individual characters SLE 13.9 - Use iteration (for statements) to iterate over list items SLE 13.6 - Perform common operations on lists or strings SLE .13.7 - Use iteration (for loops) to iterate over lists and strings SLE 13.8 - Use variables to keep track of counts and sums SLE 13.9 - Combine key programming language features to develop solutions to meaningful problems	**What we want the students to learn/be able to:** SLE - 14.1 - Describe how digital images are composed of individual elements SLE 14.2 - Recall that the colour of each picture element is represented using a sequence of binary digits SLE 14.3 - Define key terms such as 'pixels', 'resolution', and 'colour depth' SLE 14.4- Describe how an image can be represented as a sequence of bits SLE 14.5 - Describe how colour can be represented as a mixture of red, green, and blue, with a sequence of bits representing each colour's intensity SLE 14.6 - Compute the representation size of a digital image, by multiplying resolution (number of pixels) with colour depth (number of bits used to represent the colour of individual pixels) SLE 14.7 - Describe the trade-off between representation size and perceived quality for digital images SLE 14.8 - Perform basic image editing tasks using appropriate software and combine them in order to solve more complex problems requiring image manipulation SLE 14.9 - Explain how the manipulation of digital images amounts to arithmetic operations on their digital representation SLE 14.10 - Describe and assess the creative benefits and ethical drawbacks of digital manipulation (Education for a Connected World)	**What we want the students to learn/be able to:** SLE 15.1- Identify components of the CPU. Label a diagram. SLE 15.2 - Understand and define key terms: CPU, register, bus, RAM, address, location, CU, ALU, Acc, MAR, MDR, hw, sw, data, instruction, memory, FDE cycle, secondary storage SLE 15.3 - Describe VN architecture and how data moves around the CPU. SLE15.4 Identify factors affecting performance of the CPU - clock speed, RAM, Cache, Cores SLE 15.6Describe the difference between GP and embedded systems **Algorithms/coding** SLE 15.7Students understand the need for different data types and use them in programs.	**What we want the students to learn/be able to:** SLE 16.1 - Explain the difference between data and information SLE 16.2 - Critique online services in relation to data privacy SLE 16.3 - Identify what happens to data entered online SLE 16.4 - Explain the need for the Data Protection Act SLE 16.5 - Recognise how human errors pose security risks to data SLE 16.6 - Implement strategies to minimise the risk of data being compromised through human error SLE 16.7 - Define hacking in the context of cyber security SLE 16.8 - Explain how a DDoS attack can impact users of online services SLE 16.9 - Identify strategies to reduce the chance of a brute force attack being successful SLE 16.10 - Explain the need for the Computer Misuse Act SLE 16.11 - List the common malware threats SLE 16.12 - Examine how different types of malware causes problems for computer systems SLE 16.13 - Question how malicious bots can have an impact on societal issues SLE 16.14 - Compare security threats against probability and the potential impact to organisations SLE 16.17 - Explain how networks can be	**What we want students to learn/be able to:** SLE - 17.1 - Define data science SLE 17.2 - Explain how visualising data can help identify patterns and trends in order to help us gain insights SLE 17.3 - Use an appropriate software tool to visualise data sets and look for patterns or trends SLE 17.4 - Recognise examples of where large data sets are used in daily life SLE 17.5 - Select criteria and use data set to investigate predictions SLE 17.6 - Evaluate findings to support arguments for or against a prediction SLE 17.7 - Define the terms 'correlation' and 'outliers' in relation to data trends SLE 17.8 - Identify the steps of the investigative cycle SLE 17.9 - Solve a problem by implementing steps of the investigative cycle on a data set SLE 17.10 - Use findings to support a recommendation SLE 17.11 - Identify the steps of the investigative cycle SLE 17.12 - Identify the data needed to answer a question defined by the learner SLE 17.13 - Create a data capture form SLE 17.14 - Describe the need for data cleansing SLE 17.15 - Apply data cleansing techniques to a data set SLE 17.16 - Visualise a data set SLE 17.17 - Visualise a data set SLE 17.18 - Analyse visualisations to identify patterns, trends, and outliers SLE 17.19 - Draw conclusions and report findings	**What we want the students to learn/be able to:** SLE 18.1 - Students identify key network hardware: NIC, router, switch (hub), ethernet cable, WAP SLE 18.2 - Students identify different transmission media and their advantages and disadvantages (fibre optic, ethernet, wireless) SLE 18.3 - Students compare wired and wireless networks and discuss the advantages and disadvantages of both including the threats and risks to wired and wireless networks SLE 18.4 - Students understand how encryption is used to send data securely in networks **Algorithms/coding** SLE 18.5 - Students are using subprograms to decompose and simplify larger programs SLE 18.6 - Students explain the difference between functions and procedures SLE 18.7 - Physical computing - Python on Microbits
Computer Science	Autumn Term		**Spring term**		Summer term	
Topics Summary	**1**	**2**	**3**	**4**	**5**	**6**
Year 10	**1.6 Ethical and Legal impacts on Digital Technology**	**1.1 Systems Architecture**	**1.2 Memory and storage**	**1.3 Computer Networks and protocols**	**1.4 Network Security**	**1.5 Systems Software**
Topic Summary	**Acquiring knowledge:** **Mastery**: **Creating independence**:	**Mastery**: How can you make an image or sound out of 0s and 1s? Why and how can you compress a file? Not fearing failure: Can you create a process to solve a complex problem? **Acquiring knowledge:** Can you use a trace table to track data in an algorithm? What's a string and why would you manipulate it?	**Acquiring knowledge:** How does data move around the CPU? What's the relationship between the CPU and RAM? **Mastery**: What's the difference between data and instructions? What's the difference between an address and data? Why do data types matter? **Creating independence**: How can you improve the performance of your PC?	**Acquiring knowledge**: How do computers make decisions? (Logic gates) How do you convert between bases 2, 10 and 16? How do you perform calculations (additions and shifts) in Base 2 How can 1s and 0s become numbers and text? How do you convert between units of storage? How is Computational Thinking used to solve complex problems? Can you design a solution to a problem using CT skills?	**Acquiring knowledge:** What is a threat to networks and computers? How can we identify potential threats? **Mastery**: identifying vulnerabilities and knowledge of prevention methods **Creating independence**: How can you act on vulnerabilities on your own home computers and devices?	**Changing the world:** Can you solve a (any) problem with code? **Not fearing failure:** Can you create your own program? Can you solve your own problems? **Acquiring knowledge**: How can you make your code more efficient? **Creating ambition:** Why should you future-proof your code? Is your code maintainable?
Thinking Hard	**Grit / SA** - Students to research real life case studies of Technology having a positive and negative impact on society	**Grit / SA** - Students to Read on how china restricts content on its network for its citizens	**Self-control/grit** - debugging your code **Kindness/Gratitude**: Help someone else debug their code. Team roles/peer teaching - paired programming Digital divide - performance v cost	**Grit** - learning new Mathematical operations using different bases. **Mindfulness**: How much does data weigh? Students understand that data has a 'mass'/size **Curiosity**: How much storage is required by a single character / a page of text / a book? Why do we need different bases? (Binary/hex)	**Self-control/grit** - Learning sometimes doesn't happen straight away. Read anything you do not understand or ask questions **Kindness/Gratitude**: Peer assessment and learning can help to solve difficult problems	**Grit / self assurance/optimism:** Students are working on a much more detailed coding solution and must take it from problem to solution using a range of CT techniques. They will design, develop, test and evaluate their project. **Gratitude/Kindness**/peer teaching: students will support each other and test each others' programs. Opportunity for paired programming.
Developing Character	**Acquiring cultural capital:** GCHQ and the Police - What are their roles in digital technology? **Respecting human rights**: When technology is used for criminal activity? what is the impact ?	**Acquiring cultural capital:** developing an awareness of the data stored on your devices / phones (movies, songs) **Respecting human rights**: Who has your data and what can they do with it?	**Understanding democracy:** Being aware that you might not be the only person who uses your code. Being a part of the coding community. **No limits to your destination:** What problem would you like to see solved with code? **Self control:** Is it OK to take someone else's' code?	**Understanding environmental diversity:** If data takes up space, where is it all stored - considering the environmental aspect of data centres.	Be a good digital citizen and use equipment and networks safely	There is more than one way to solve a problem. Does it matter how long/short/complex/user friendly your solution is?
Understanding Diversity	**Understanding mental & physical diversity:** What age should children be allowed to gain access to the internet?	**Understanding mental & physical diversity:** Different strengths. Which challenges can you complete? Peer teaching opportunity.	Legal issues - copyright Careers - who builds computers? Who innovates - what careers are available?	Binary in real life (past, Hitler, future - AI, can we get AI to make non-binary decisions, what about face scanning and decisions/judgements made from this)	How many Threats to computer systems does Wimbledon tennis get each year?	Problem solving Attitudes & Behaviours
Literacy Reading, Oracy	Literacy and reading - Comprehension and presentation of writing	Development of literacy: Syntax, debugging (compare to proofreading), importance of accuracy in code." **Oracy**: Talking your code aloud to debug. **Oracy** - COP 26 - describe a climate change issue we are facing **Literacy** - researching/reading about careers in computer programming	**Oracy** - Describe how data moves around the CPU.	**Reading** -Green technology - which invention could have the most impact? **Literacy** - summarise the text.	Literacy - Analyse and evaluate information to make recommendations	Literacy - following / applying correct syntax in coding
Gatsby, Careers	BBC Careers Roadshow	Research careers as a computer programmer - Do Now task. Careers - Degree apprenticeships. Show slide with available apprenticeships p32 booklet. And discuss.	Top universities for Computing - p37 booklet Creating ambition: what jobs are available to coders?	IT trainer or teacher - you're the teacher role reversal - what's it like to be a teacher/trainer?	Careers in technology. Look at current vacancies for jobs in the local area	What's it like to be a SW developer or games designer? Invite one in for Q and A. R Time article - stem careers - studio engineer
Mental and Physical Well-being	Consider the implications on yourself and impact to your mental health in the event of a hacker	Does being anonymous make it easier to be mean?	Screen breaks - eye exercises	Ensure that you are getting enough sleep verses screen time	Digital well-being: Using computers ethically and legally	Focus your time with tech

Cross-Curricular Links	Business - Legal and ethical side of technology in business	Maths - image and sound file sizes and how to calculate them. Media - compressing sound and image files	Science - circuits Numeracy/Maths - data types, particularly integer & real	Science - logic gates and circuits Maths - different bases and performing calculations in binary. Geog / PD - the effect of data on the environment	Science - Experiments and testing Geog / PD - the effect of data on the environment	Maths - operators < > = != PD - supporting your peers / accepting support
Extra-Curricular Links	European Languages day - September Dyslexia Awareness Day	Oct COP26 - Climate Change External afterschool Scratch computer club Nov STEM day Dec International day of the disabled person Nov Anti-Bullying Week Dec International day of the disabled person Dec Human Rights Day	Jan - National Technology Day Feb - Safer Internet Day - R Time activities, all year groups. External afterschool Scratch computer club	External afterschool Scratch computer club Black History Month - Y10 - how can we make Computer Science more inclusive?	External afterschool Scratch computer club	Algorithms - the villains and heroes of the 'post-truth' era April - World Day - R time article, 5 ways AI is helping the climate crisis External afterschool Scratch computer club.
Specific Learning Endpoints	**What we want students to learn/be able to:** **1.6.1 Ethical, legal and cultural environmental impact -** Impact of digital technology on wider society. Legislation relevant to computer science.	What we want students to learn/be able to: **1.1.1 The Purpose of a CPU - State the actions that occur at each stage of the fetch-execute cycle.** The role/purpose of each component and what it manages, stores. or controls during the fetch-execute cycle. The purpose of each register, what it stores (Data or Address). The difference between storing data and address. Common CPU components and their function - ALU,CU,Cache,Registers. Von Neumann architecture - MAR (Memory Address Register), MDR (Memory Data Register) Program Counter, Accumulator **1.1.2 CPU Performance - How common chartists of CPUs affect performance:** Clock speed, cache size, number of cores. Understanding of each characteristic as listed The effects of changing any of the common characteristics on system performance, either individually or in combination **1.1.3 Embedded Systems -** The purpose and characteristic of embedded systems, examples of embedded systems. Understanding of each characteristic as listed The effects of changing any of the common characteristics on system performance, either individually or in	**What we want students to learn/be able to:** **1.2.1 Primary storage:** The need for primary storage. **1.2.**2 Secondary storage : The need for secondary storage ¨ Common types of storage: o Optical o Magnetic o Solid state **1.2.3 Units** Why computers have secondary storage **1.2.4 Data storage** The units of data storage: Numbers, Characters, Images and sound. **1.2.5 Compression**	**What we want students to learn/be able to:** **1.3.1 Networks and topologies** Types of networks **1.3.2 Wired and wireless network**s, **protocols and layers,** Modes of connection	**What we want a students to learn/ be able to:** 1.4.1 Threats to computer systems and networks - Forms of attack: 1.4.2 Identifying and preventing vulnerabilities - Common Prevention methods	**What we want students to learn/be able to:** **1.5.1 Operating systems -** The purpose and functionality of operating systems **1.5.2 Utility Software** The purpose and functionality of utility software
Computer Science	**Autumn Term**		**Spring term**			**Summer term**
Year 11	**1**	**2**	**3**	**4**	**5**	**6**
Topic Summary	**2.1 Algorithms**	**2.2 Programming fundamentals**	**2.3 Produce robots programs**	**2.4 Boolean Logic**	**2.5 Programming languages and integrated Development Environments**	Revision of all topics
Thinking Hard	**Changing the world:** Can any computer system be truly safe? Is there more to the Operating System than what you see? What is future-proofing? Why don't all tech companies do it? What is thorough testing? When do you write a test plan?	**Acquiring knowledge:** What is thorough testing? How do you write a test plan? How did programming languages develop? Why do we code in HLL if they need translating? **Changing the world**: Do you share music/movies/software? Is this right? Where does your waste Tech go? Does it matter? Do you care? Creating independence: Are you being manipulated by Tech companies?	Acquiring knowledge: How does a computer search for data? How does a computer sort data? How do you search and sort? What's the difference between searching and sorting? Mastery: How do you recognise searches and sorts in algorithms and code	**Acquiring knowledge:** Can you speak more then one digital language fluently? **Changing the world**: What other computer languages are emerging or developing? Do you care? Creating independence: Practicing Boolean Logic will help you master your programming skills	**Acquiring knowledge:** What are simple logic diagrams? Do truth tables always tell the truth? **Changing the world**: What is the future of IDEs? Do you care? Creating independence: Create a revision plan or what you need to practice	**Creating independence:** Algorithms/code - writing, completing, Exam technique and practice **Creating ambition:** Exam technique. How to respond to different command words. How to manage your time.
Developing Character	**Self assurance** - life online - are you responsible? Who should pay if you lose money online? You install dodgy SW - who's to blame? **Mindfulness**: Whose fault is it if you get hacked? What if your password was weak?	**Curiosity**: Why test with bad data? **Self-control**: Copyright - how much of your data do you own? Think before you share **Self assurance**: Taking part in international computational thinking competition. **Optimism**: Are you happy with your IDE?	**Grit**: Determining between different searches and sorts. **Curiosity**: Why do we need more than one search or sort?	**Curiosity**: Why learn Boolean programming language? **Self-control**: How can you be sure to not make a mistake when programming? **Self assurance**: Is there any peers you can learn from in class? **Optimism**: What other subjects could you apply these skills to?	**Curiosity**: Why learn about IDE's? **Self assurance**: Is there any peers you can learn from in class? **Optimism**: What other subjects could you apply these skills to?	**Grit**: not giving up. **Self assurance** - be prepared, using your revision time and exam time to the maximum potential. **Kindness/Gratitude**: Working at different levels. Practising different tasks to your peers. Making online revision quizzes for use in class.
Understanding Diversity	Apple want to scan your photos to find child abusers - is this OK? Who makes your OS. Are you OK with this? Understanding mental & physical diversity - Different OS, good for different purposes, e.g. phone/tablet/computer. Diff features - Linux. How can an OS be adapted for people with different needs? **Understanding democracy** - Open source v proprietary - the developer community - freeware, shareware.	**Understanding environmental diversity**: Considering the impact our use of Tech has on the Earth. What is technology doing to the earth? Is it improving it? **Awareness of where you live**: What happens locally to our tech waste? **Being a world citizen**: What about mankind? Are we better off with or without tech? **Respecting human rights**: Digital divide: Technology in different parts of the world - . E-waste	**Acquiring cultural capital:** What are computers searching for? Why do we need more than one search or sort?	**Understanding environmental diversity**: Considering the impact of power needed for the digital world **Awareness of where you live**: What can you do to cut back on using electricity in relation to computing? **Being a world citizen**: Globally how can the world cut down on energy and CO2 generated by computers? **Respecting human rights**: Should everyone have access to technology in todays climate?	**Understanding environmental diversity**: Considering the impact our use of Tech has on society. What is technology doing to the economies? Is it improving it? **Awareness of where you live**: What happens locally to our tech waste? **Being a world citizen**: What about mankind? Are we better off with or without tech? **Respecting human rights**: Digital divide: Technology in different parts of the world - . E-waste	**No limit to your destination:** Fill in the knowledge gaps. Use Knowledge organisers to support your revision
Literacy Reading, Oracy	**Reading**: Read article about apple scanning photos for child abuse or the evolution of Operating Systems	**Literacy**: Answering long answer questions, Understanding the mark scheme. Planning your answer (can do this as a verbal exercise first) **Oracy** - COP 26 - describe a climate change issue we are facing	**Literacy**, planning - developing a program	Literacy - Comprehension and digital literacy for Boolean Logic language	Literacy - Key command words for answering exam paper 1 and 2	Exam skills - read the question, recognise and respond to exam command words. Development of literacy: Long answer exam Qs. Command words in exams.

Gatsby, Careers	SW developer. What OS are they using in Industry for what jobs? Cyber Security	SW Developer - what does the job involve?	What Computing courses are available to you?	Progression to FE or apprenticeships for computer Science	Jobs and roles that Computer Science will lead to	Exam skills, college, careers - next steps R Time article - stem careers - studio engineer
Mental and Physical Well-being	Positives and negatives of digital technologies	Changing circumstances - the effects of corona on disabled people	The importance of screen breaks	Paced Revision with breaks and appropriate rest and diet	Getting enough Sleep to aid learning and revision	Managing your revision time. Healthy habits for productive revision.
Cross-Curricular Links	Maths - encryption PD - Social Engineering, responsible use of Internet and systems	PD /Geog - Environmental effects of Tech MFL - translating languages	Maths - searching, sorting, comparisons, patterns	Maths - Logic and operators	Science - hypothesis and evaluation of programming	Exam skills apply across all subjects. Students will have similar command words in other subjects: identify, explain, discuss, advantages, disadvantages
Extra-Curricular Links	Sept European day of languages - google translate the do now task. External after school Scratch club Black History Month - exam Q - The software we use is primarily created by white men. How is this reflected in the software itself? How do we make software to reflect the experiences of all minority groups?	External after school Scratch club Nov STEM day Nov Anti-Bullying Week Dec International day of the disabled person Human Rights Day Oct COP26 - Climate Change	Jan - National Technology Day External after school Scratch club Feb - Safer Internet Day - R Time activities, all year groups.	External after school Scratch club	External after school Scratch club	April - World Day - R time article, 5 ways AI is helping the climate crisis. External after school Scratch club
Specific Learning Endpoints	**What we want students to learn/be able to:** **2.1.1 Computational thinking -** Principles of computational thinking. **2.1.2 Designing, creating and refining algorithms -** Identify the inputs, processes and outputs for a problem **2.1.3 Searching and sorting Algorithms -** Standard searching algorithms and standard sorting algorithms.	**2.2.1 Programming fundamentals -** The use of variables, constants, operators ad assignments. Three basic programming constructs used to control program flow. The common arithmetic operators. The common Boolean Operators AND, OR, NOT **2.2.2 Data Types. -** Integer,Real,Boolean,Character and string, Casting **2.2.3 Additional programming techniques -** The basic string manipulation. The use of basic file handling Record and store data. The use of arrays, Sub programs and random number generator.	**What we want students to learn/be able to:** **2.3.1 Defensive design -** Considerations, Input validation and maintainability. **2.3.2 Testing -** The purpose of testing, Types, Identify syntax and logic errors, Selecting and using suitable test data, Refining algorithms.	**What we want students to learn/ be able to :** **2.4.1 Boolean logic -** Simple logic diagrams using operators. Truth tables, combining Boolean operators, Applying logic operators	**What we want students to learn/be able to :** **2.5.1 Languages -** Characteristics and purpose of different levels of programming language The purpose of translators. The characteristics of compiler and interpreter. **2.5.2 The integrated Development Environments (IDE)** - Common tools for facilities available in an integrated Development Environment.	**What we want students to learn/be able to:** Possess the knowledge required to answer both papers. Create accurate Python code in Paper 2. Apply their knowledge and exam techniques well, giving them confidence in both exams. Perform well in the exams, do not leave gaps, check their answers, use the time available. Bring all unit knowledge together1.1,1.2,1.3,1.4,1.5,1.6,2.1,2.2,2.3,2.4,2.5