

| Computing                             | Autumn Term   |  | Spring term   |  | Summer term   |  |
|---------------------------------------|---|--|---|--|---|--|
| Year 7                                | 1   | 2  | 3   | 4  | 5   | 6  |
| Topic Summary                         | Digital literacy/online citizenship   | Programming essential 1  | Networks - from semaphore to the Internet   | Programming II   | Spreadsheets  | Developing for the web   |
| <b>Thinking Hard</b>                  | <b>Acquiring knowledge / curiosity:</b> Using new systems responsibly. What is a network? How do you keep them secure? Where is the cloud?<br>How can you make a computer do what you want it to do?<br>How can a computer or different computers be made to carry out a range of different tasks?<br><b>Mindfulness:</b> Do you change who you are depending on who you are talking to?<br><i>Creating independence:</i> If you find it on the Internet, can you use it? (copyright)   | <b>Acquiring knowledge/numeracy:</b> What is a variable? Can you solve problems with code?<br><b>Self-assurance:</b> Is there any problem a computer can't solve?<br><b>Creativity:</b> Is Coding the most creative skill there is?  | <b>Acquiring knowledge/curiosity:</b> What is hardware? Who owns the Internet? What are the advantages of networking? Are there any disadvantages?<br>How do you build a network? Wired or wireless?<br><b>Mastery</b> How does data move through a network? What is a protocol?<br><b>World citizen:</b> What's the difference between the Internet and the WWW? | <b>Grit/curiosity:</b> Why do we need subprograms? Does efficiency matter?<br><b>Acquiring knowledge:</b> How do computers make decisions?   | <b>Acquiring knowledge/curiosity:</b> What is a spreadsheet?<br>What applications do they have in real life?<br>What's a cell/table/column/field/record?<br>Using basic formulas, using conditional formatting. Filtering & sorting data.<br><b>Mastery:</b> Complex functions and formulas | <b>Acquiring knowledge:</b> How do you make a web page?<br><b>Mastery:</b> Can you write a website in html/ use CSS for s        |
| <b>Developing Character</b>           | <b>Mindfulness and self-control:</b> Online responsibility & staying secure online<br><b>Grit/optimism:</b> <i>Collaborating online in lessons.</i><br><b>Gratitude, Kindness, Problem solving - coping with complexity, persevere, handle ambiguity, adapt, communicate, investigate.<br/><b>Self control - Using the school LAN</b><br/><b>Self assurance:</b> <i>Presenting to an audience - can you adjust how you speak depending on who you are talking to?</i></b>   | <b>Mindfulness/grit</b> - Developing stuckability, debugging your code.<br><b>Self-assurance/curiosity</b> - PRIMM - predicting outcomes -<br><b>Independence/SA/Creativity</b> - creating your own program<br>Helping others, paired programming<br><b>Grit:</b> How easily do you give up? Can you predict outcomes? | <b>Mindfulness</b> - how much time do you spend on the Internet? Do you/could you go tech free? How long for?<br><b>Curiosity/gratitude:</b> Imagine a world without computer networks, how different would your life would be.<br><b>Self assurance/awareness/optimism:</b> Do you rely on the Internet? What would lockdown be like without the Internet?       | Mindfulness & Grit/not fearing failure - developing stuckability, debugging your code.<br><b>Self-assurance</b> - PRIMM - predicting outcomes<br><b>Independence/SA/Creativity/Curiosity</b> - creating your own program<br><b>Self-assurance/awareness/kindness</b> - helping others, paired programming.<br><b>Grit:</b> Do you have great stuckability?<br>Not fearing failure/ kindness: Does it matter if someone else can or can't read your code? (maintainability) | <b>Grit</b> - writing your own formulas<br><b>Self awareness/self assurance/curiosity:</b> How could you use a spreadsheet in your life?<br><b>Mastery:</b> Why use a spreadsheet instead of a calculator?  | <b>Grit</b> - coding in HTML   |
| <b>Understanding Diversity</b>        | <b>Understanding environmental diversity/respecting human rights:</b> What is the Digital divide? Local v global level of DD - empathy<br>Computers everywhere - from simple to complex. Does it have to be complex to solve global problems?<br><b>Understanding mental and physical diversity/Optimism:</b> <b>How can computers assist with health and disabilities?</b><br><b>Anti-bullying and understanding people with disabilities-</b> video with Prince William<br>Computers - the people - hidden figures. How has life changed? Exerpt from the book as well as a video clip. | <b>Mastery/world citizen:</b> Are all computers the same? Are they all expensive? Can you solve a problem with a microbit? Can anyone write code? HLLs<br>Should you understand how your computer works?   | <b>Mastery</b> - Not all computers are expensive. How can everyone access technology. How could you use a Microbit to solve a problem?<br><b>Kindness/Gratitude:</b> Digital divide - local v global. Should you pay for the Internet   | <b>Mastery:</b> Humans v computers - why do we count in 10s? Are there other number bases? How would these work?<br><b>Kindness:</b> Debug someone else's code/paired programming  | <b>Self assurance/kindness:</b> Different programming languages for different purposes, ages, experiences. Open Source v proprietary software   | What adaptations are there on webpages for people with different needs -eg visual impairments, physical disabilities             |
| <b>Literacy Reading, Oracy</b>        | Computer literacy and fluency - logging on to school's network, Google Cloud, Insight understanding the uses of each<br>Do Now task - Read article on "Cyberbullying: Being Bullied Online and Advice on What to Do".   | Literacy: Syntax in code= grammar<br>Key programming vocabulary<br>Oracy - COP 26 - descibe a climate change issue we are facing   | Literacy - keyword vocabulary, The importance of syntax   | Sequential ordering of algorithms and programming - why is order important?  | Literacy - keyword vocabulary pertaining to spreadsheet (functions, conditional formatting, data validation. use of spell check, find and replace)  | Choosing appropriate text for a specific audience, use of good grammar, spelling and punctuation and its importance on websites. |
| <b>Gatsby, Careers</b>                | Do now task - Identify skills that can be developed/ learnt throughout computing and how they can be used and developed in future careers.<br>Skills- Communication, creativity, presentation skills, IT software skills, research skills, data analysis  | Research job description, salary for careers where programming is used.<br>-Computer Programmer<br>- App/game developer<br>-Software developer   | Work with school IT dept, students to come up with questions to ask them to get knowledge on how they run the school network. Industry experience   | Review skills they have previously learnt in other topics including prgramming one. What skills could they continue to develop in this topic. How will that help them in future careers  | Careers - Spreadsheets in the world of work/Transferable skills. Interview Finance team. Data analysis, link to crosscurrucular - Sport performance analysis, research analysis   | Web designers - duties, salary, education  |
| <b>Mental and Physical Well-being</b> | Online safety & responsibility<br>Cyber bullying presentation   | Using fitbits to track activity/ microbits   | Screen breaks - physical well-being   |  | Physical well-being - create a spreadsheet which tracks healthy eating.   | <b>Mindfulness</b> - online jigsaw in silence  |
| <b>Cross-Curricular Links</b>         | PD (online safety & responsibility)<br>History - key moments in the development of computers, to include Enigma and Collosus in WW2   | Maths - logic, variables, constancts, problem-solving  | Science/physics - connecting your Microbit  | Numeracy - binary conversions, adding binary numbers, different bases<br>MFL - writing a translation program   | Maths - writing Mathematical formulae, BIDMAS, variables, constant, problem solving<br>PE- analysis performance analysis data on a performer  | Art, Photography - selecting and editing images that are English/literacy - using text which is appropriate to a spe             |
| <b>Extra-Curricular Links</b>         | <b>Networks in your home.</b><br><b>26th Sept</b> European day of languages - google translate the do now task. Respecting human rights/understanding democracy:<br>Black History month - clips from Hidden Figures movie:<br>Katherine Johnson, Dorothy Vaughan, Mary Jackson  | Bebras - external competition running as afterschool club. First Lego League running as after school club.<br>Cyber Discovery Phase 1 (Assess)<br>15-19 Nov Anti-Bullying Week<br>3 Dec International day of the disabled person<br>10 Human Rights Day  | 6 January - National Technology Day<br><br>Safer Internet Day - Feb - assemblies and Tutor time activities.   | International Women's Day - Focus on Ada Lovelace, Mary Coombs<br><a href="https://www.youtube.com/watch?v=X0yYDxjBnTM&amp;t=4s">https://www.youtube.com/watch?v=X0yYDxjBnTM&amp;t=4s</a> ]  | World Day for Cultural Diversity - Digital Divide - countries where there is a lack of technology and access to the internet.   | pride month - Alan Turing: who is he, why is he important?   |

|                                       |   |   |  |  |   |  |
|---------------------------------------|---|---|--|--|---|--|
| <p>Precise Learning Endpoints</p>     | <ul style="list-style-type: none"> <li>Acquire skills to use computers around the school responsibly. Recognise they are accessing a network/Google Drive, use Google Apply knowledge to access Classroom and Insight to track Home Learning.</li> <li>Articulate the fundamentals of a computers system and define (with examples): computer, network, the cloud, input/output device, HW and SW.</li> <li>Demonstrate an awareness of the historical development of computers</li> <li>Appraise effective presentations for a given audience</li> <li>Recognise cyberbullying and analyse its effects</li> <li>Differentiate between safe and unsafe online behaviour.</li> </ul> <p>REVISION</p> | <ul style="list-style-type: none"> <li>Compare how humans and computers understand instructions (understand and carry out) <ul style="list-style-type: none"> <li>Recognise that computers follow the control flow of input/process/output</li> <li>Define a sequence as instructions performed in order, with each executed in turn</li> <li>Predict the outcome of a simple sequence</li> <li>Modify a sequence</li> <li>Define a variable as a name that refers to data being stored by the computer</li> <li>Predict the outcome of a simple sequence that includes variables</li> <li>Trace the values of variables within a sequence</li> <li>Make a sequence that includes a variable</li> <li>Define a condition as an expression that will be evaluated as either true or false</li> <li>Identify that selection uses conditions to control the flow of a sequence</li> <li>Identify where selection statements can be used in a program</li> <li>Modify a program to include selection</li> <li>Create conditions that use comparison operators (&gt;, &lt;, =)</li> <li>Create conditions that use logic operators (and/or/not)</li> <li>Identify where selection statements can be used in a program that include comparison and logical operators</li> <li>Define iteration as a group of instructions that are repeatedly executed</li> <li>Describe the need for iteration</li> <li>Identify where count-controlled iteration can be used in a program</li> <li>Implement count-controlled iteration in a program</li> <li>Detect and correct errors in a program (debugging)</li> <li>Independently design and apply programming constructs to solve a problem (subroutine, selection, count-controlled iteration, operators, and variables)</li> </ul> </li> </ul> <p>REVISION</p> | <ul style="list-style-type: none"> <li>Define what a computer network is and explain how data is transmitted between computers across networks <ul style="list-style-type: none"> <li>Define 'protocol' and provide examples of non-networking protocols</li> <li>List examples of the hardware necessary for connecting devices to networks</li> <li>Compare wired to wireless connections and list examples of specific technologies currently used to implement such connections</li> <li>Define 'bandwidth', using the appropriate units for measuring the rate at which data is transmitted, and discuss familiar examples where bandwidth is important</li> <li>Define what the internet is</li> <li>Explain how data travels between computers across the internet</li> <li>Describe key words such as 'protocols', 'packets', and 'addressing'</li> <li>Explain the difference between the internet, its services, and the World Wide Web</li> <li>Describe how services are provided over the internet</li> <li>List some of these services and the context in which they are used</li> <li>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)</li> <li>Describe how internet-connected devices can affect me</li> <li>Describe components (servers, browsers, pages, HTTP and HTTPS protocols, etc.) and how they work together</li> </ul> </li> </ul> <p>REVISION &amp; ASSESSMENT 2</p> | <ul style="list-style-type: none"> <li>Define a subroutine as a group of instructions that will run when called by the main program or other subroutines <ul style="list-style-type: none"> <li>Define decomposition as breaking a problem down into smaller, more manageable subproblems</li> <li>Identify how subroutines can be used for decomposition</li> <li>Identify where condition-controlled iteration can be used in a program</li> <li>Implement condition-controlled iteration in a program</li> <li>Evaluate which type of iteration is required in a program</li> <li>Define a list as a collection of related elements that are referred to by a single name</li> <li>Describe the need for lists</li> <li>Identify when lists can be used in a program</li> <li>Use a list</li> <li>Decompose a larger problem into smaller subproblems</li> <li>Apply appropriate constructs to solve a problem</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>Identify columns, rows, cells, and cell references in spreadsheet software <ul style="list-style-type: none"> <li>Use formatting techniques in a spreadsheet</li> <li>Use basic formulas with cell references to perform calculations in a spreadsheet (+, -, *, /)</li> <li>Use the autofill tool to replicate cell data</li> <li>Explain the difference between data and information</li> <li>Explain the difference between primary and secondary sources of data</li> <li>Collect &amp; Analyse data</li> <li>Create appropriate charts in a spreadsheet</li> <li>Use the functions SUM, COUNTA, MAX, and MIN in a spreadsheet</li> <li>Use a spreadsheet to sort and filter data</li> <li>Use the functions AVERAGE, COUNTIF, and IF in a spreadsheet</li> <li>Use conditional formatting in a spreadsheet</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>Describe what HTML is <ul style="list-style-type: none"> <li>Use HTML to structure static web pages</li> <li>Modify HTML tags using inline styling to improve t</li> <li>Display images within a web page</li> <li>Apply HTML tags to construct a web page structur</li> </ul> </li> <li>Describe what CSS is <ul style="list-style-type: none"> <li>Use CSS to style static web pages</li> <li>Assess the benefits of using CSS to style pages instea</li> <li>Describe what a search engine is</li> <li>Explain how search engines 'crawl' through the World V</li> <li>Analyse how search engines select and rank results wt</li> <li>Use search technologies effectively</li> <li>Discuss the impact of search technologies and the issu</li> <li>Create hyperlinks to allow users to navigate between n</li> <li>Implement navigation to complete a functioning website</li> <li>Complete summative assessment</li> </ul> </li></ul> |
| <p>Computing</p>                      | <p>Autumn Term</p>  |   | <p>Spring term</p>   |  | <p>Summer term</p>  |  |
| <p>Year 8</p>                         | <p>1</p>  | <p>2</p>  | <p>3</p>   | <p>4</p>   | <p>5</p>  | <p>6</p>   |
| <p>Topic Summary</p>                  | <p>Computing Systems</p>  | <p>Python - Intro to text-based languages</p>   | <p>Representations - from clay to silicon</p>  | <p>Mobile App Development</p>  | <p>Developing for the web / Green Computing</p>   | <p>Computational thinking &amp; Key algorithms</p>   |
| <p>Thinking Hard</p>                  | <p><b>Acquiring knowledge/curiosity:</b><br/>How long have computers existed?<br/>What's the difference between a general purpose and an embedded computer system?<br/>What's the difference between hardware and software?<br/><b>Mastery:</b> What is a computer system? What's with the 1s and 0s? What's the future of AI?<br/>NC links: 3.4, 3.5, 3.6</p>  | <p><b>Acquiring knowledge/curiosity:</b><br/>What is an algorithm? Why use text-based coding languages?<br/><b>Mastery:</b> What is logic? What is syntax? Why iterate? What are the different ways to iterate?<br/>NC Links: 3.1, 3.3, 3.6</p>   | <p><b>Acquiring knowledge/curiosity:</b><br/>How can you turn 1s and 0s into text/images/video/sound?<br/>What is analogue? What is digital?<br/>What is compression? How do you compress a file?<br/>What's the difference between lossy and lossless compression? Which is better in which situation?<br/><b>Mastery:</b> Why do we need bits &amp; bytes?<br/>NC links: 3.6</p>   | <p><b>SA:</b> Which app is the best and why?<br/><b>Mastery:</b> What makes a good GUI? Which apps are missing from the market?<br/><b>Creativity/ Curiosity</b> - Can you make an app?<br/>NC links: 3.1, 3.3, 3.8</p>  | <p><b>Curiosity:</b> What's your phone made from?<br/><b>Mastery:</b> Is data bad for the environment? Where is your data?<br/><b>Acquiring knowledge:</b> What is html? Can you make a website from scratch? Do you search efficiently? How do search engines choose what you see? Do you search safely?<br/>How do you generate traffic on a website?</p>   | <p><b>Acquiring knowledge/curiosity</b><br/>How do computers sort data? How do computers search for data?<br/>How do you sort your socks? How do you search for your socks?<br/>What does the code for a search/sort look like?<br/><b>Mastery:</b> Are some ways of sorting better than others?<br/><b>Self assurance:</b> Can CT skills help us solve real world problems<br/><b>Mastery:</b> What algorithms rule your life? Can you improve them?</p>  |
| <p>Developing Character</p>           | <p><b>Selfassurance/optimism:</b> AI - what are the applications of that make/would make the world a better place?<br/><b>SA/curiosity:</b> Turing Test - could a computer convince you it's human? Can a computer provide friendship?<br/><b>Being a world citizen/kindness:</b> Can you do what you want with software? Who owns it?</p>  | <p><b>Creativity/Curiosity</b> - making a quiz in Python<br/><b>Grit</b> - debugging code<br/><b>Self-assurance/ kindness-</b> paired programming</p>   | <p><b>Mastery/creating independence:</b> Moore's law: How much storage do you need? Can we keep expanding storage capacity and processing power? What are the consequences of this?</p>  | <p><b>Grit/not fearing failure</b> - debugging code<br/><b>Gratitude &amp; kindness-</b> paired programming<br/><b>Self assurance:</b> Can you live without your apps? What apps do we need to improve the world?<br/>What apps appeared during lockdown?</p>  | <p><b>Mindfulness/being a world citizen/awareness of where you live:</b> Students consider the effects of our consumption of technology on the environment. Where does your ewaste go?<br/>Who is responsible for ewaste?<br/><b>Respecting human rights/understanding democracy:</b> poor working conditions..<br/><b>Self assurance:</b> Do you consider who's collecting your data? Why do you need to upgrade your phone?<br/><b>SA/self control/Acquiring cultural capital:</b> What makes you use a website? Do you think before you click?</p>   | <p><b>Curiosity/mastery:</b> Do computers think?<br/><b>Grit</b> - problem-solving skills<br/><b>Mastery/Self-assurance</b> - making the abstract concrete. Explaining how computers sort and search to another student.</p>   |
| <p>Understanding Diversity</p>        | <p><b>Respecting human rights/literacy:</b><br/>Black History Month: Do Now Task - reading comprehension task: "7 Famous Black Computing Pioneers"</p>  | <p><b>Self assurance / kindness:</b> Different strengths. Which challenges can you complete? Peer teaching opportunity.</p>   | <p><b>Self assurance/being a world citizen:</b> Ascii v Unicode - the need for character sets that represent all languages</p>   | <p>Apps to help society - eg Mental Health, People with disabilities...</p>  | <p><b>Being a world citizen/PD:</b> Digital divide, ewaste in other countries<br/><b>Curiosity/respecting human rights:</b> Who made your phone?<br/><b>Understanding environmental diversity:</b> What happens when resources run out? Does tech create more problems than it solves?</p>  | <p><b>Kindness/optimism:</b> Real life situations - locked in syndrome - can you use Computational Thinking skills to solve problems that aren't computer related?</p>   |
| <p>Literacy Reading, Oracy</p>        | <p>Students read passage about anitkythera mechanism and decide if it is a computer.<br/><b>Black History Month: Do Now Task</b> - reading comprehension task: "7 Famous Black Computing Pioneers"<br/>Students read unfamiliar text from Hidden Figures and respond.<br/>Do Now task - read an article on "FOMO" and relate it to social media and online life.<br/>Synonyms worksheet</p>   | <p>Syntax - debugging your own programs.<br/>Talking your program aloud to find your bugs.</p>  |  | <p>Understanding target audience - developing an app for a particular target audience</p>  | <p><b>Understanding Search Engines - keyword searches</b></p>   | <p>Sorting - alphabetical order - ascending/descending</p>   |
| <p>Gatsby, Careers</p>                | <p><b>Students write Qs for the network technicians. 1 per class.</b></p>   | <p>Careers/Options - Is CS for you?</p>   | <p><b>Creating ambition/changing the world</b> - App Inventor, businesses built on apps, eg fast fashion</p>   | <p>App Inventor, businesses built on apps, eg fast fashion. Who can make an app - research activity.</p>   |   | <p><b>No limits to your destination</b> / Careers week - what careers are out there if you like STEM?</p>  |
| <p>Mental and Physical Well-being</p> | <p>FOMO - Fear of Missing Out - how does FOMO affect your screen-time?</p>  | <p>The data self</p>  | <p>Keeping up appearances - the selfie v yourself</p>  | <p>Gamification</p>  |   |  |

|                                   |   |   |   |  |  |   |
|-----------------------------------|---|---|---|--|--|---|
| <b>Cross-Curricular Links</b>     | E & P - Ethical/moral issues - software licensing, AI, do we want computers to be unidentifiable?<br>Maths - binary and place values, logical (Boolean) operators, arithmetic operators<br>Science - circuits/logic, variables, switches, transistors<br>Careers - IT Technician - students talk to IT Technicians about their roles  | Maths (variables, constants), problem solving).<br>Numeracy - calculations in programs/arithmetic expressions   | History - the development of character sets from ASCII to Unicode.<br>Art - the advancement of images in video games from 8 bit and up<br>Numeracy - adding binary  | Science - logic gates<br>Maths - arithmetic operators and logic expressions  | Maths - data analysis<br>FE - spreadsheets<br>PD - the environment   | Maths - sorting, sequencing   |
| <b>Extra-Curricular Links</b>     | 26th Sept European day of languages - google translate the do now task.<br>Black History month Y8 - Investigate 7 Black computing pioneers, who was the most influential?<br>Self control- what's your moral code - software piracy<br>Self assurance - Bebras - external competition running as afterschool club<br>Lego First league running as club<br>Maths - number bases - binary   | Bebras - external competition running as afterschool club. First Lego League running as after school club.<br>Cyber Discovery Phase 1 (Assess)<br>15-19 Nov Anti-Bullying Week<br>3 Dec International day of the disabled person<br>10 Human Rights Day   | Understanding democracy: Cyber First for Girls<br>Creating ambition/not fearing failure: The Turing Cryptology Competition<br>6 January - National Technology Day   | Careers week 1-6 March<br>No limits/understanding democracy/creating ambition/changing the world: Cyber First for Girls<br>26 Feb - Internet Safety Day <b>Self assurance/ self control:</b> Assemblies and R Time activities<br>3 March World book Day  | World citizen - World Earth Day - 22 April<br>Mental Health Awareness week - 10-16 May 2021  | Self assurance: Cyber Discovery Phase 1 (Assess) begins in June (until Sept)  |
| <b>Precise Learning Endpoints</b> | <ul style="list-style-type: none"> <li>Recall that a general-purpose computing system is a device for executing programs</li> <li>Recall that a program is a sequence of instructions that specify operations that are to be performed on data</li> <li>Explain the difference between a general-purpose computing system and a purpose-built device</li> <li>Describe the function of the hardware components used in computing systems</li> <li>Describe how the hardware components used in computing systems work together in order to execute programs</li> <li>Recall that all computing systems, regardless of form, have a similar structure ('architecture')</li> <li>Analyse how the hardware components used in computing systems work together in order to execute programs</li> <li>Define what an operating system is, and recall its role in controlling program execution</li> <li>Describe the NOT, AND, and OR logical operators, and how they are used to form logical expressions</li> <li>Use logic gates to construct logic circuits, and associate these with logical operators and expressions</li> <li>Describe how hardware is built out of increasingly complex logic circuits</li> <li>Recall that, since hardware is built out of logic circuits, data and instructions alike need to be represented using binary digits</li> <li>Provide broad definitions of 'artificial intelligence' and 'machine learning'</li> <li>Identify examples of artificial intelligence and machine learning in the real world</li> <li>Describe the steps involved in training machines to perform tasks (gathering data, training, testing)</li> <li>Describe how machine learning differs from traditional programming</li> <li>Associate the use of artificial intelligence with moral dilemmas</li> <li>Explain the implications of sharing program code</li> </ul> | <ul style="list-style-type: none"> <li>Describe what algorithms and programs are and how they differ</li> <li>Recall that a program written in a programming language needs to be translated in order to be executed by a machine</li> <li>Write simple Python programs that display messages, assign values to variables, and receive keyboard input</li> <li>Locate and correct common syntax errors</li> <li>Describe the semantics of assignment statements</li> <li>Apply simple arithmetic expressions in assignment statements to calculate values</li> <li>Receive input from the keyboard and convert it to a numerical value</li> <li>Apply relational operators to form logical expressions</li> <li>Apply binary selection (if, else statements) to control the flow of program execution</li> <li>Generate and use random integers</li> <li>Apply multi-branch selection (if, elif, else statements) to control the flow of program execution</li> <li>Describe how iteration (while statements) controls the flow of program execution</li> <li>Apply iteration (while loops) to control the flow of program execution</li> <li>Demonstrate use of variables as counters in iterative programs</li> <li>Combine iteration and selection to control the flow of program execution</li> <li>Demonstrate use of Boolean variables as flags</li> </ul> <p><b>ASSESSMENT 1</b></p> | <ul style="list-style-type: none"> <li>Identify examples of representations</li> <li>Recall that representations are used to store, communicate, and process information</li> <li>Provide examples of how different representations are appropriate for different tasks</li> <li>Recall that characters can be represented as sequences of symbols and list examples of character coding schemes</li> <li>Measure the length of a representation as the number of symbols that it contains</li> <li>Provide examples of how symbols are carried on physical media</li> <li>Explain what binary digits (bits) are, in terms of familiar symbols such as digits or letters</li> <li>Measure the size or length of a sequence of bits as the number of binary digits that it contains</li> <li>Describe how natural numbers are represented as sequences of binary digits</li> <li>Convert a decimal number to binary and vice versa</li> <li>Convert between different units and multiples of representation size</li> <li>Provide examples of the different ways that binary digits are physically represented in digital devices</li> </ul> | <ul style="list-style-type: none"> <li>Identify when a problem needs to be broken down</li> <li>Implement and customise GUI elements to meet the needs of the user</li> <li>Recognise that events can control the flow of a program</li> <li>Use user input in an event-driven programming environment</li> <li>Use variables in an event-driven programming environment</li> <li>Develop a partially complete application to include additional functionality</li> <li>Identify and fix common coding errors</li> <li>Pass the value of a variable into an object</li> <li>Establish user needs when completing a creative project</li> <li>Apply decomposition to break down a large problem into more manageable steps</li> <li>Use user input in a block-based programming language</li> <li>Use a block-based programming language to create a sequence</li> <li>Use variables in a block-based programming language</li> <li>Use a block-based programming language to include sequencing and selection</li> <li>Use user input in a block-based programming language</li> <li>Use variables in a block-based programming language</li> <li>Reflect and react to user feedback</li> <li>Use a block-based programming language to include sequencing and selection</li> <li>Use user input in a block-based programming language</li> <li>Use variables in a block-based programming language</li> <li>Evaluate the success of the programming project</li> </ul> <p>REVISION &amp; ASSESSMENT 2</p> | <ul style="list-style-type: none"> <li>Describe what HTML is</li> <li>Use HTML to structure static web pages</li> <li>Modify HTML tags using inline styling to improve the appearance of web pages</li> <li>Display images within a web page</li> <li>Apply HTML tags to construct a web page structure from a provided design</li> <li>Describe what CSS is</li> <li>Demonstrate use CSS to style static web pages</li> <li>Assess the benefits of using CSS to style pages instead of in-line formatting</li> <li>Describe what a search engine is</li> <li>Explain how search engines 'crawl' through the World Wide Web and how they select and rank results</li> <li>Analyse how search engines select and rank results when searches are made</li> <li>Apply search technologies effectively</li> <li>Discuss the impact of search technologies and the issues that arise by the way they function and the way they are used</li> <li>Create hyperlinks to allow users to navigate between multiple web pages</li> <li>Discuss issues of safety and security from a technological perspective</li> <li>Discuss the impact of networking technologies and services</li> </ul> | <ul style="list-style-type: none"> <li>Identify and describe 4 strands of CT: abstraction, decomposition, pattern recognition, algorithmic thinking.</li> <li>Apply CT to real life problems</li> <li>Know and describe key terms: algorithm, sort, search, linear, binary search, bubble sort, merge sort, insertion sort</li> <li>Demonstrate a sort</li> <li>Demonstrate a search</li> <li>Understand and explain efficiency in searching and sorting</li> </ul> |
| <b>Computer Science</b>           | Autumn Term   |   | Spring term   |  | Summer term  |   |
| <b>Year 9</b>                     | 1   | 2   | 3   | 4  | 5  | 6   |
| <b>Topic Summary</b>              | Cyber Security / Python Revisited   | Data Representation - Going audio visual  | The CPU - Von Neumann & Beyond / HTML   | Secondary storage  | Networks / Python - functions&procedures   | Coding Project  |
| <b>Thinking Hard</b>              | <b>Acquiring knowledge/curiosity:</b> How can a computer represent and process sequences of data, such as lists and strings.<br>How can you access an element in a sequence?<br>How can you iterate through a list?<br>What is program flow?<br><b>Mastery:</b> What is the difference between data and information?  | <b>Acquiring knowledge/creating independence:</b> What are different number bases? Why are they needed?<br>How can you represent text/images/sound/video with just 1s and 0s?<br>How can you convert analogue to digital?<br>How can you represent colours?<br>How many colours in a photograph?<br>Why compress? What's the difference between lossy and lossless compression?<br>Being creative: Can you create you own images with code?   | Acquiring knowledge / curiosity:<br>What is going on in the CPU?<br>How does the CPU communicate with other hardware & software?<br>What's behind a website?<br>How would you attack a website?   | <b>Acquiring knowledge:</b><br>What's the difference between main memory and secondary storage?<br>Why do computer systems need Secondary storage?<br>Volatile or non-volatile? What is read/write speed?<br><b>Mastery:</b> What factors should you consider when choosing secondary storage?<br><b>Changing the world:</b> What's next for secondary storage? The development of memory technology - cost v capacity   | <b>Acquiring knowledge/How do you build a network?</b><br>How many networks do you use?<br>Why do we need protocols?<br><b>Changing the world:</b> What happens if the Internet goes down. Permanently.<br>Not fearing failure/creating independence/mastery: developing code independently  | <b>Being creative / creating ambition:</b> Can you take a coding project through from conception to evaluation?<br>Changing the world:Can we solve any problem with code?<br><b>Creating independence:</b> How do you decompose a complex problem?<br>Programming Project task tp be decided.   |
| <b>Developing Character</b>       | <b>Self assurance</b> - Taking part in international computational thinking competition. (Bebras)<br><b>Grit:</b> Students encounter realistic problems : solar system planets, book texts, capital cities, leaked passwords, word dictionaries, ECG data.<br><b>Self assurance / teamwork:</b> Cyber Discovery competition<br><b>Self control:</b> Why hack? Is it ever OK to hack??<br><b>Optimism</b> - debug your code  | <b>Self assurance</b> - Taking part in international computational thinking competition. (Bebras)<br><b>Kindness</b> - can you explain number bases to someone else?  | <b>Self control/Being a world citizen:</b> Websites - revisit digital artifacts/copyright. Can you use it? Should you?<br><b>Mindfulness:</b> How can you fit a billion transistors into a single chip?   | <b>Self assurance</b> - Taking part in international computational thinking competition. (Bebras)<br>Optimism: What's next for secondary storage?  | <b>Grit/self-assurance</b> - developing code independently<br>Considering secondary storage - cost v capacity & performance.<br>Protocols - what protocols are there in society?<br>What new protocols appeared during lockdown?x<br>Which have we adopted?<br><b>Curiosity:</b> What's in a data packet?<br><b>Mindfulness:</b> Can you imagine a world without computers?  | <b>Grit &amp; self-assurance</b> - developing code independently<br><b>Gratitude:</b> Peer assessment / reviewing each others' code   |
| <b>Understanding Diversity</b>    | <b>Acquiring cultural capital:</b> Does privacy still exist?<br><b>Respecting human rights:</b> Who has your data? Do you mind? What about sensitive data? Is it OK to collect data on race, gender,sexuality, appearance, disability<br><b>Awareness of where you live:</b> How can code solve real-life problems?<br><b>Not fearing failure:</b> Will you code work first time?   | <b>Understanding mental and physical diversity:</b> Which methods are you using to solve the conversions? Why can't your computer count?  | Understanding democracy: Are all computers big & expensive?<br>Careers - Key figures in Computing - what was their background? How did they get there.<br>Making the abstract concrete: Demystifying the CPU<br>Being a world citizen/awareness of where you live: Ewaste - where have all the hard drives gone?  | No limits: is cloud storage the future?<br>Respecting human rights: what are the human/privacy/legal implications of using cloud storage more and more?<br>Understanding environmental diversity: what are the implications on the environment of data storage?  | <b>Acquiring cultural capital:</b> How did the Internet change your experience of Lockdown?<br><b>Respecting human rights</b> - How do you keep a network it secure?   | <b>No limits:</b> Different solutions to the same problem   |
| <b>Literacy Reading, Oracy</b>    | <b>Oracy</b> - Talking the bugs out of your programs<br>Development of literacy: Syntax, debugging (compare to proof-reading), importance of accuracy in code.<br><b>Reading</b> - <a href="#">the dark web</a>   | Write a leaflet to explain clearly how to convert into and out of binary.   | Trip to Team GB / Inesco<br>Trip to TNMoC<br>Careers - Key figures in Computing - what was their background? How did they get there.  | First Legoleague club and competition (Dec)<br>Bebras Competition (Nov)<br>Maths - logical problems<br>Careers week 1-6 March  |  | <b>Literacy:</b> Syntax, debugging (compare to proof-reading), importance of accuracy in code.  |



|                                       |  |  |   |   |  |   |
|---------------------------------------|--|--|---|---|--|---|
| <b>Gatsby, Careers</b>                | Cyber Intelligence Officer - do you love to hack? P20 of CS careers booklet, reading.  | Careers - Graphic Designer, Video Editor, Music Producer   | Science - circuits, transistors, logic gates<br>Maths - logic, binary   | Numeracy/Maths - problem solving using CT techniques  | What do our network technicians do?  | Software engineer   |
| <b>Mental and Physical Well-being</b> | <b>Digital well-being:</b> Log your screen time  | Digital identities   |   |   |  |   |
| <b>Cross-Curricular Links</b>         | First Legoleague club and competition (Dec)<br>Preparation for Bebras Competition (Nov)<br>Cyber Discovery Stage 1 (Assess) continues  | Art - digital images/PhotoShop<br>Music - Audacity, creating and saving digital music<br>Media - video editing/compression<br>Numeracy - different units and bases   |   |   | PD - protocols in society  | Numeracy/Maths - problem solving, logical operators   |
| <b>Extra-Curricular Links</b>         | 26th Sept European day of languages - google translate the do now task.<br>Python - create a multiple choice languages quiz<br>Black History Month - Y9 - Choose one <a href="#">influential black programmers</a> to research and present (oracy),  | First Legoleague club and competition (Dec)<br>Preparation for Bebras Competition (Nov)<br>Cyber Discovery Stage 1 (Assess) continues<br>8 Nov STEM day<br>15-19 Nov Anti-Bullying Week<br>3 Dec International day of the disabled person<br>10 Human Rights Day   | 6 Jan - National Technology Day<br>8 Mar - International women's day - teachers select an inspirational woman to share with the class.  |   |  |   |
| <b>Precise Learning Endpoints</b>     | <ul style="list-style-type: none"> <li>Explain the difference between data and information</li> <li>Critique online services in relation to data privacy</li> <li>Identify what happens to data entered online</li> <li>Explain the need for the Data Protection Act</li> <li>Recognise how human errors pose security risks to data</li> <li>Implement strategies to minimise the risk of data being compromised through human error</li> <li>Define hacking in the context of cyber security</li> <li>Explain how a DDoS attack can impact users of online services</li> <li>Identify strategies to reduce the chance of a brute force attack being successful</li> <li>Explain the need for the Computer Misuse Act</li> <li>List the common malware threats</li> <li>Examine how different types of malware causes problems for computer systems</li> <li>Question how malicious bots can have an impact on societal issues</li> <li>Compare security threats against probability and the potential impact to organisations</li> <li>Explain how networks can be protected from common security threats</li> <li>Identify the most effective methods to prevent cyberattacks</li> </ul> <p>Python</p> <ul style="list-style-type: none"> <li>Write programs that display messages, receive keyboard input, and use simple arithmetic expressions in assignment statements</li> <li>Use selection (if-elif-else statements) to control the flow of program execution</li> <li>Locate and correct common syntax errors</li> <li>Create lists and access individual list items</li> <li>Perform common operations on lists or individual items</li> <li>Use iteration (while statements) to control the flow of program execution</li> <li>Perform common operations on lists, individual items, strings or individual characters</li> <li>Use iteration (for statements) to iterate over list items, lists &amp; strings</li> <li>Use variables to keep track of counts and sums</li> <li>Combine key programming language features to develop solutions to meaningful problems</li> </ul> | <ul style="list-style-type: none"> <li>Convert between different bases - denary, binary, hex</li> <li>Describe how digital images are composed of individual elements</li> <li>Recall that the colour of each picture element is represented using a sequence of binary digits</li> <li>Define key terms such as 'pixels', 'resolution', and 'colour depth'</li> <li>Describe how an image can be represented as a sequence of bits</li> <li>Describe how colour can be represented as a mixture of red, green, and blue, with a sequence of bits representing each colour's intensity</li> <li>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)</li> <li>Describe the trade-off between representation size and perceived quality for digital images</li> <li>Perform basic image editing tasks using appropriate software and combine them in order to solve more complex problems requiring image manipulation</li> <li>Explain how the manipulation of digital images amounts to arithmetic operations on their digital representation</li> <li>Describe and assess the creative benefits and ethical drawbacks of digital manipulation (Education for a Connected World)</li> <li>Recall that sound is a wave</li> <li>Explain the function of microphones and speakers as components that capture and generate sound</li> <li>Define key terms such as 'sample', 'sampling frequency/rate', 'sample size'</li> <li>Describe how sounds are represented as sequences of bits</li> <li>Calculate representation size for a given digital sound, given its attributes</li> <li>Explain how attributes such as sampling frequency and sample size affect characteristics such as representation size and perceived quality, and the trade-offs involved</li> <li>Perform basic sound editing tasks using appropriate software and combine them in order to solve more complex problems requiring sound manipulation</li> <li>Recall that bitmap images and pulse code sound are not the only binary representations of images and sound available</li> <li>Define 'compression', and describe why it is necessary</li> <li>Python - turtle</li> <li>Students can import turtle module</li> <li>Students can draw lines, shapes, fill shapes, change colour, change pen width</li> <li>Students use iteration to create repeating patterns.</li> </ul> | <ul style="list-style-type: none"> <li>Identify components of the CPU. Label a diagram.</li> <li>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</li> <li>Describe VN architecture</li> <li>Describe how data moves around the CPU. C24</li> <li>Identify factors affecting performance of the CPU - clock speed, RAM, Cache, Cores</li> <li>Describe the difference between GP and embedded systems</li> <li>Create a basic website using HTML</li> <li>Employ key terms: tag, body, head, content, domain name, url</li> </ul> <p>REVISION</p> | <ul style="list-style-type: none"> <li>Students identify the difference between main memory and secondary storage and explain the need for both in Computer Systems.</li> <li>Students identify the factors affecting choice of secondary storage device/media: portability, robustness, cost, reliability, durability, read/write speed</li> <li>Students discuss the advantages and disadvantages of different types of secondary storage</li> <li>Students consider the use of Cloud Storage and the advantages/disadvantages of Cloud storage.</li> </ul> <p>ASSESSMENT 1</p> | <ul style="list-style-type: none"> <li>Students identify key network hardware: NIC, router, switch (hub), ethernet cable, WAP</li> <li>Students identify different transmission media and their advantages and disadvantages</li> <li>Students identify key network protocols and their purpose</li> <li>Students compare wired and wireless networks and discuss the advantages and disadvantages of both.</li> <li>Students identify the threats and risks to wired and wireless networks</li> <li>Students understand how encryption is used to send data securely in networks</li> <li>Students are using subprograms to decompose and simplify larger programs</li> <li>Students explain the difference between functions and procedures</li> </ul> | <ul style="list-style-type: none"> <li>Students can take a project from initial concept through full life cycle:</li> <li>Analyse - decompose, abstract, recognise patterns. Create success criteria</li> <li>Design - create algorithms (flow charts and pseudocode)</li> <li>Develop - write code from their design in Python, debug</li> <li>Test - Use trace tables and relevant data to test</li> <li>Evaluate - compare final program with success criteria and evaluate the success of the project</li> <li>Students predict and verify outcomes by using trace tables.</li> <li>Students manage their programs using subprograms</li> <li>Students consider boundary, normal and erroneous data when selecting test data.</li> </ul> <p>REVISION &amp; ASSESSMENT 2</p> |
| <b>Computer Science</b>               | <b>Autumn Term</b>   |  | <b>Spring term</b>  |   | <b>Summer term</b>   |   |
| <b>Year 10</b>                        | <b>1</b>   | <b>2</b>   | <b>3</b>  | <b>4</b>  | <b>5</b>   | <b>6</b>  |
| <b>Topic Summary</b>                  | <p><b>Acquiring knowledge:</b> How do computers make decisions? (Logic gates)</p> <p>How do you convert between bases 2, 10 and 16?<br/>How do you perform calculations (additions and shifts) in Base 2<br/>How can 1s and 0s become numbers and text?<br/>How do you convert between units of storage?</p> <p>How is Computational Thinking used to solve complex problems?<br/>Can you design a solution to a problem using CT skills?</p>  | <p><b>Mastery:</b> How can you make an image or sound out of 0s and 1s?<br/>Why and how can you compress a file?<br/>Not fearing failure: Can you create a process to solve a complex problem?<br/><b>Acquiring knowledge:</b> Can you use a trace table to track data in an algorithm?<br/>What's a string and why would you manipulate it?</p>   | <p>Acquiring knowledge: How does data move around the CPU?<br/>What's the relationship between the CPU and RAM?<br/>Mastery: What's the difference between data and instructions? What's the difference between an address and data? Why do data types matter?<br/>Creating independence: How can you improve the performance of your PC?</p>   | <p>Acquiring knowledge: What's the difference between primary memory and secondary storage?<br/>Creating independence: What should you consider when choosing a storage device?<br/>Mastery: How can memory that deletes between uses have a use?</p>   | <p><b>Changing the world:</b> Can you solve a (any) problem with code?<br/><b>Not fearing failure:</b> Can you create your own program? Can you solve your own problems?<br/><b>Acquiring knowledge:</b> How can you make your code more efficient?<br/><b>Creating ambition:</b> Why should you future-proof your code?<br/>Is your code maintainable?</p>  | <p>Acquiring knowledge: Why network? How do you build a network?<br/>How is data transferred across a network?<br/>Mastery: What are layers? What are protocols?<br/>What's the difference between the Internet and the WWW?</p>  |
| <b>Thinking Hard</b>                  | <p><b>Grit</b> - learning new Mathematical operations using different bases.<br/><b>Mindfulness:</b> How much does data weigh? Students understand that data has a 'mass'/size<br/><b>Curiosity:</b> How much storage is required by a single character / a page of text / a book?<br/>Why do we need different bases? (Binary/hex)</p>  | <p><b>Grit / SA</b> - Students take part in international computational thinking competition.</p>  | <p>Self-control/grit - debugging your code<br/>Kindness/Gratitude: Help someone else debug their code.<br/>Self control: Is it OK to take someone else's code?<br/>Team roles/peer teaching - paired programming<br/>Digital divide - performance v cost<br/>Creating ambition: what jobs are available to coders?</p>  | <p>Mindfulness: How many computers do you come into contact with in your daily life?<br/>How would lockdown have been if you didn't have access to the Internet?</p>  | <p><b>Grit / self assurance/optimism:</b> 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.<br/><b>Gratitude/Kindness/peer teaching:</b> students will support each other and test each others' programs. Opportunity for paired programming.</p>   | <p><b>Self-assurance/ grit</b> - understanding abstract concepts ie layers and protocols.<br/><b>Mindfulness:</b> Who owns the Internet? Where is the Cloud?</p>  |

|                                       |   |  |   |  |  |  |
|---------------------------------------|---|--|---|--|--|--|
| <b>Developing Character</b>           | <b>Understanding environmental diversity:</b> If data takes up space, where is it all stored - considering the environmental aspect of data centres.  | <b>Understanding mental &amp; physical diversity:</b> Different strengths. Which challenges can you complete? Peer teaching opportunity.<br><b>Acquiring cultural capital:</b> developing an awareness of the data stored on your devices / phones (movies, songs)<br><b>Respecting human rights:</b> 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.<br>No limits to your destination: What problem would you like to see solved with code?   | Understanding democracy/Digital divide - Lockdown and the Internet. Did we all have the same access? Did you have days without the Internet?<br>Understanding mental and physical diversity: Computers in everyday life/accessibility - what embedded devices are there to help people with disabilities?  | There is more than one way to solve a problem. Does it matter how long/short/complex/user friendly your solution is?   | <b>Changing the world:</b> Different ways of building networks. Advantages and disadvantages of each. Awareness of where you live - WWW invented by TBL whilst working at Southampton University<br><b>Being a world citizen</b> - The internet is a network of networks which you become part of every time you connect.  |
| <b>Understanding Diversity</b>        | 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)   |  | <i>Legal issues - copyright</i><br><i>Careers - who builds computers? Who innovates - what careers are available?</i>   | Careers - all careers link to IT. What career interests you? How will you use Tech in this career?<br>Careers week 1-6 March<br>Work Exp this term: 26 April, 2 weeks  | Problem solving Attitudes & Behaviours   | Careers Q & A - invite Technicians to talk to students about the network and their roles as NW Technicians.  |
| <b>Literacy Reading, Oracy</b>        | <b>Reading</b> -Green technology - which invention could have the most impact?<br><b>Literacy</b> - summarise the text.   | Development of literacy: Syntax, debugging (compare to proof-reading), importance of accuracy in code."<br><b>Oracy:</b> Talking your code aloud to debug.<br><b>Oracy</b> - COP 26 - describe a climate change issue we are facing<br><b>Literacy</b> - researching/reading about careers in computer programming   |   | PD - digital divide  | Literacy - following / applying correct syntax in coding   | PD - protocols - what protocols exist in life/society/lockdown? Are they important?  |
| <b>Gatsby, Careers</b>                | IT trainer or teacher - you're the teacher role reversal - what's it like to be a teacher/trainer?  | Research careers as a computer programmer - Do Now task.<br>Careers - Degree apprenticeships. Show slide with available apprenticeships p32 booklet. And discuss.  | Top universities for COmputing - p37 booklet  |  | What's it like to be a SW developer or games designer? Invite one in for Q and A.  |  |
| <b>Mental and Physical Well-being</b> |   |  |   |  |  |  |
| <b>Cross-Curricular Links</b>         | Science - logic gates and circuits<br>Maths - different bases and performing calculations in binary.<br>Geog / PD - the affect of data on the environment   | Maths - image and sound file sizes and how to calculate<br>Media - compressing sound and image files   | Science - circuits<br>Numeracy/Maths - data types, particularly integer & real  |  | Maths - operators < > = != PD - supporting your peers / accepting support  |  |
| <b>Extra-Curricular Links</b>         | 26th Sept European day of languages - google translate the do now task.<br>Black History Month - Y10 - how can we make Computer Science more inclusive?   | 26-31 Oct COP26 - Climate Change<br>8 Nov STEM day<br>3 Dec International day of the disabled person<br>15-19 Nov Anti-Bullying Week<br>3 Dec International day of the disabled person<br>10 Dec Human Rights Day<br>"External Bebras Computational Thinking Competition   | 6 Jan - National Technology Day   | 26 Feb - Internet Safety Day<br>3 March World book Day<br>8 Mar - International women's day - teachers select an in  | Algorithms - the villains and heroes of the 'post-truth' era   |  |
| <b>Precise Learning Endpoints</b>     | 2.4.1 Students know the truth tables for each logic gate (AND, OR, NOT). They use and recognise each gate symbol. They understand how to create, complete or edit logic diagrams and truth tables for given scenarios. They can work with more than one gate in a logic diagram.<br>1.2.4 Convert between different bases (binary to 1 byte, hex to FF, denary to 255).<br>Perform calculations - addition and right/left shifts in binary and understand the effect of a shift.<br>Understand how characters are represented in binary, how the number of characters stored is limited by the bits available<br>Explain the differences between and impact of each character set (ASCII v Unicode)<br>Understand how character sets are logically ordered, e.g. the code for 'B' will be one more than the code for 'A'<br>*Binary representation of ASCII in the exam will use 8 bits<br>2.1.1 Understand the principles of computational thinking: Abstraction, Decomposition, Algorithmic thinking and how they are used to define and refine problems.<br>Practical programming - plan, design, test and evaluate a program that takes user input and returns it as output. Use selection and iteration. Apply specific terminology: variable, input, output, assignment, concatenation, casting, data type. They debug their code and use comments. | 1.2.4 Images - Understand that each pixel has a specific colour, represented by a specific code. Articulate the effect on image size and quality when changing colour depth and resolution.<br>Explain that Metadata stores additional image information (e.g. height, width, etc.)<br>Sound - Understand that analogue sounds must be stored in binary. Explain that process. Understand that sample rate is measured in Hertz (Hz), that duration is how many seconds of audio the sound file contains and that Bit depth is the number of bits available to store each sample (e.g. 16-bit)<br>Explain the need for compression and can give relevant examples.<br>Identify the advantages and disadvantages of each type of compression (lossy/lossless)<br>Recognise the effects on the file for each type of compression.<br>2.1.2 Produce simple (flow) diagrams to show the structure of a problem, subsections and their links to other subsections. Complete, write or refine an algorithm, identify syntax/logic errors in code and suggest fixes. Create and use trace tables to follow an algorithm<br>Practical programming - students manipulate strings. | 1.1 Identify the actions occurring at each stage of the fetch-execute cycle. Apply knowledge of the role/purpose of each component of the CPU (ALU, CU, Cache, Registers) and what it manages, stores, or controls during the fetch-execute cycle.<br>VN architecture: Articulate the purpose of each register (MAR, MDR, PC, Acc), what it stores (data or address) and they understand the difference between storing data and an address.<br>1.1.2 Define clock speed, cache size and number of cores and explain the effects of changing them on system performance, either individually or in combination.<br>2.2.1 Identify and utilise variables, constants, operators, inputs, outputs and assignments<br>Identify the three basic programming constructs used to control the flow of a program: Sequence, Selection, Iteration (count- and condition-controlled loops)<br>Recognise the common comparison operators (==, !=, <, >, >=) and arithmetic operators (+, -, *, /, MOD, DIV, ^)<br>Recognise the common Boolean operators AND, OR and NOT<br>2.2.2 Use data types (integer, real, Boolean, Character and string, casting) in a high-level language within the classroom. Choose suitable data types for data in a given scenario. Understand that data types may be temporarily changed through casting, and where this may be useful."<br><br>REVISION & ASSESSMENT 1 | 1.1.3 Identify the purpose and characteristics of embedded systems with examples.<br>1.2.1 Understand the need for primary storage<br>They know that this usually consists of RAM and ROM, the difference between these and can identify the key characteristics and purpose of both.<br>Understand and explain why virtual memory may be needed in a system and how VM works (the transfer of data between RAM and HDD when RAM is filled).<br>1.2.2 Explain why computers have secondary storage.<br>Recognise a range of secondary storage devices/media (optical, magnetic, solid state). Identify differences between each type of storage device/medium (Capacity, Speed, Portability, Durability, Reliability, Cost) compare advantages/disadvantages for each storage device.<br>Apply their knowledge in context within given scenarios.<br>2.2.1 Use and identify variables, constants, operators, inputs, outputs and assignments<br>Use and can identify the three basic programming constructs used to control the flow of a program: Sequence, Selection, Iteration (count- and condition-controlled loops)<br>Recognise the common comparison operators (==, !=, <, >, >=) and arithmetic operators (+, -, *, /, MOD, DIV, ^)<br>2.2.3 Recognise the common Boolean operators AND, OR and NOT<br>Apply the additional programming techniques in a high-level language (Python)<br>Manipulate strings, including: Concatenation, Slicing, Arrays as fixed length static structures, Discriminate between functions & procedures (the difference between them / where to use them)<br>Apply basic file handling techniques: open, read, write, close<br>Generate random numbers in code | 2.2.3 Recognise the common Boolean operators AND, OR and NOT<br>Apply additional programming techniques in a high-level language (Python)<br>Manipulate strings, including: Concatenation, Slicing, Arrays as fixed length static structures, The use of functions & procedures (and the difference between them / where to use them)<br>Apply basic file handling techniques: open, read, write, close<br>Generate random numbers in code<br>Explain basic concepts in SQL and its use of records to store and search for data.<br>Understand and apply SQL commands: SELECT, FROM, WHERE | 1.3.1 Explain and apply knowledge of the characteristics of LANs and WANs including common examples of each.<br>Distinguish between different factors that can affect the performance of a network, e.g.: Number of devices connected, Bandwidth<br>Describe the tasks performed by each piece of network hardware (WAPs, Routers, Switches, NIC, Transmission media)<br>Explain the concept of the Internet as a network of computer networks (DNS, hostin, the CCloud, Web servers and clients)<br>Explain a DNS's role in the conversion of a URL to an IP address<br>Articulate the different roles of computers in a client-server and a peer-to-peer network, Understand the concept of servers providing services (e.g. Web server " Web pages, File server " file storage/retrieval) , the concept of clients requesting/using services from a server<br>Describe the Cloud: remote service provision (e.g. storage, software, processing)<br>Identify advantages and disadvantages of the Cloud and the advantages and disadvantages of the Star and Mesh topologies<br>Apply understanding of networks to a given scenario<br>1.3.2 Compare benefits and drawbacks of wired versus wireless connection (wired - ethernet v wireless - wi-fi, bluetooth). Recommend one or more connections for a given scenario.<br>Describe the principle of encryption to secure data across network connections<br>IP addressing and the format of an IP address (IPv4 and IPv6), A MAC address is assigned to devices; its use within a network<br>Describe the principle of a standard to provide rules for areas of computing, Standards allows hardware/software to interact across different manufacturers/producers<br>Understand the principle of a (communication) protocol as a set of rules for transferring data. Packet switching<br>That different types of protocols are used for different purposes (TCP/IP, HTTP(s), FTP, POP, IMAP, SMTP)<br>The basic principles of each protocol i.e. its purpose and key features<br>How layers are used in protocols, and the benefits of using layers;the 4-layer TCP/IP model<br><br>REVISION & ASSESSMENT 2 |
| <b>Computer Science</b>               | Autumn Term   |  | Spring term   |  | Summer term  |  |
| <b>Year 11</b>                        | 1   | 2  | 3   | 4  | 5  | 6  |
| <b>Topic Summary</b>                  | 1.4.1 Threats to computer systems and networks<br>1.4.2 Identifying and preventing vulnerabilities<br>1.5.1 Operating Systems<br>2.3.1 Defensive design<br>2.3.2 Testing  | 2.3.2 Testing<br>2.5.1 Languages<br>2.5.2 The IDE<br>1.5.2 Utility Software<br>1.6.1 Ethical, Legal, cultural & Environmental impact   | Programming Revision<br>2.1.3 Searching & Sorting Algorithms<br>Searching & sorting Practical Programming skills  | Practical Prgramming skills Revision<br>Theory Revision<br>Mocks & Therapy   | Topics to be picked according to skills analysis / areas of weakness.  |  |

|                                       |  |  |  |  |   |
|---------------------------------------|--|--|--|--|---|
| <b>Thinking Hard</b>                  | <b>Changing the world:</b> Can any computer system be truly safe?<br><br>Is there more to the Operating System than what you see?<br>What is future-proofing? Why don't all tech companies do it?<br>What is thorough testing? When do you write a test plan?  | <b>Acquiring knowledge:</b> What is thorough testing? How do you write a test plan?<br>How did programming languages develop? Why do we code in HLL if they need translating?<br><b>Changing the world:</b> Do you share music/movies/software? Is this right?<br>Where does your waste Tech go? Does it matter? Do you care?<br>Creating independence: Are you being manipulated by Tech companies?   | Acquiring knowledge: How does a computer search for data?<br>How does a computer sort data? How do you search and sort?<br>What's the difference between searching and sorting?<br>Mastery: How do you recognise searches and sorts in algorithms and code | Not fearing failure: Do you know your strengths and weaknesses?<br>Creating independence: What do you need to do to prepare for the exams?<br>Being creative: What strategies do you have in place?<br>Where do you need to focus your revision?<br>How are you feeling about the exams? | <b>Creating independence:</b> Algorithms/code - writing, completing, getting the marks.<br><b>Creating ambition:</b> Exam technique. How to respond to different command words. How to manage your time.  |
| <b>Developing Character</b>           | <b>Self assurance</b> - life online - are you responsible? Who should pay if you lose money online? You install dodgy SW - who's to blame?<br><b>Mindfulness:</b> Whose fault is it if you get hacked? What if your password was weak?   | <b>Curiosity:</b> Why test with bad data?<br><b>Self-control:</b> Copyright - how much of your data do you own? Think before you share<br><b>Self assurance:</b> Taking part in international computational thinking competition.<br><b>Optimism:</b> Are you happy with your IDE?   | Grit: Determining between different searches and sorts.<br>Curiosity: Why do we need more than one search or sort?   | Self-assurance: progress tracking<br>mindfulness/optimism - do you approach exams in the best mindset?<br>Self-assurance - still time to make the changes  | <b>Grit:</b> Keeping going, not giving up<br>Self assurance - be prepared, using your revision time and exam time to the maximum potential.<br><b>Kindness/Gratitude:</b> Working at different levels. Practising different tasks to your peers. Making online revision quizzes for use in class. |
| <b>Understanding Diversity</b>        | Apple want to scan your photos to find child abusers - is this OK?<br>Who makes your OS. Are you OK with this?<br>Understanding mental & physical diversity - Different OS, good for different purposes, eg phone/tablet/computer. Diff features - linux. How can an OS be adapted for people with different needs?<br><b>Understanding democracy</b> - Open source v proprietary - the developer community - freeware, shareware. | <b>Understanding environmental diversity:</b> Considering the impact our use of Tech has on the Earth. What is technology doing to the earth? Is it improving it?<br><b>Awareness of where you live:</b> What happens locally to our tech waste?<br><b>Being a world citizen:</b> What about mankind? Are we better off with or without tech?<br><b>Respecting human rights:</b> Digital divide: Technology in different parts of the world - . Ewaste | Acquiring cultural capital: What are computers searching for? Why do we need more than one search or sort?   | No limit to your destination: Your exam, your target - what do you want from the exam? How will you get it?  | <b>No limit to your destination:</b> Fill in the knowledge gaps.  |
| <b>Literacy Reading, Oracy</b>        | <b>Reading:</b> Read article about <a href="#">apple scanning photos for child abuse</a> or <a href="#">the evolution of Operating Systems</a>   | <b>Literacy:</b> Answering long answer questions, Understanding the mark scheme.<br>Planning your answer (can do this as a verbal exercise first)<br><b>Oracy</b> - COP 26 - describe a climate change issue we are facing   | Literacy, planning - developing a program  |  | Exam skills - read the question, recognise and respond to exam command words.<br>Development of literacy: Long answer exam Qs.<br>Command words in exams.   |
| <b>Gatsby, Careers</b>                | SW developer. What OS are they using in Industry for what jobs?<br>Cyber Security  | SW Developer   |  | Next steps - college and careers.<br>Careers week 1-6 March  | Exam skills, college, careers - next steps  |
| <b>Mental and Physical Well-being</b> | Positives and negatives of digital technologies  | Changing circumstances - the effects of corona on disab  | The importance of cscreen breaks   | Managing your revision time. Healthy habits for productive revision.   | Managing your revision time. Healthy habits for productive revision.  |
| <b>Cross-Curricular Links</b>         | Maths - encryption<br>PD - Social Engineering, responsible use of Internet and systems   | PD /Geog - Environmental effects of Tech<br>MFL - translating languages  | Maths - searching, sorting, comparisons, patterns  | Revision techniques and exam strategies  | Exam skills apply across all subjects. Students will have similar command words in other subjects: identify, explain, discuss, advantages, disadvantages  |
| <b>Extra-Curricular Links</b>         | 26th Sept European day of languages - google translate the do now task.<br>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 Bebras Competition<br>8 Nov STEM day<br>15-19 Nov Anti-Bullying Week<br>3 Dec International day of the disabled person<br>10 Human Rights Day<br>26-31 Oct COP26 - Climate Change   | 6 Jan - National Technology Day  | 26 Feb - Internet Safety Day<br>3 March World book Day<br>8 Mar - International women's day - teachers select an inspirational woman to share with the class.  |   |

|                                   |  |   |  |   |  |  |
|-----------------------------------|--|---|--|---|--|--|
| <p>Precise Learning Endpoints</p> | <p>1.4.1: Identify and explain forms of attack/threats to systems: o Malware o Social engineering, e.g. phishing, people as the 'weak point' o Brute-force attacks o Denial of service attacks o Data interception and theft o Explain the concept of SQL injection. Explain how each attack is used and the purpose of each attack.</p> <p>1.4.2 Discuss how to limit the threats posed in 1.4.1, ie methods to remove vulnerabilities, knowledge of the principles of each prevention method, what each prevention method may limit/prevent, how it limits the attack.</p> <p>Evaluate these common prevention methods: o Penetration testing o Anti-malware software o Firewalls o User access levels o Passwords o Encryption o Physical security</p> <p>1.5.1 Explain the purpose and functionality of operating systems including o User interface - features (GUI) o Memory management and multitasking (the transfer of data between memory, and how this allows for multitasking) o Peripheral management and drivers (Data is transferred between devices and the processor, This process needs to be managed and what this entails (e.g. the use of buffers when transferring data to a printer o User management ( Allocation of an account, Access rights, Security, etc). o File management (key features: Naming, Allocating to folders, Moving files, Saving, etc.)</p> <p>2.3.1 Evaluate defensive design considerations: o Anticipating misuse, o Authentication<br/>     - Input validation<br/>     - Maintainability:<br/>     o Use of sub programs, o Naming conventions, o Indentation, o Commenting<br/>     Demonstrate the issues a programmer should consider to ensure that a program caters for all likely input values<br/>     Demonstrate to deal with invalid data in a program, authentication to confirm the identity of a user, practical experience of designing input validation and simple authentication (e.g. username and password), understand why commenting is useful and apply this appropriately</p> <p>2.3.2 Testing: Discuss the difference between testing modules of a program during development and testing the program at the end of production<br/>     Identify syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated<br/>     Identify logic errors as errors which produce unexpected output<br/>     Identify normal test data as data which should be accepted by a program without causing errors<br/>     Identify boundary test data as data of the correct type which is on the very edge of being valid<br/>     Identify invalid test data as data of the correct type but outside accepted validation limit<br/>     Identify erroneous test data as data of the incorrect type which should be rejected by a computer system<br/>     Identify and implement suitable test data for a given scenario and to create/complete a test plan</p> | <p>2.3.2 Testing: Discuss the difference between testing modules of a program during development and testing the program at the end of production<br/>     Identify syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated<br/>     Identify logic errors as errors which produce unexpected output<br/>     Identify normal test data as data which should be accepted by a program without causing errors<br/>     Identify boundary test data as data of the correct type which is on the very edge of being valid<br/>     Identify invalid test data as data of the correct type but outside accepted validation limit<br/>     Identify erroneous test data as data of the incorrect type which should be rejected by a computer system<br/>     Identify suitable test data for a given scenario and to create/complete a test plan</p> <p>2.5.1 Discuss the differences between high- and low-level programming languages. They explain the need for translators (assembler and translator). They can discuss the differences, benefits and drawbacks of using a compiler or an interpreter</p> <p>2.5.2 Identify the tools that an IDE provides and how each of the tools and facilities listed can be used to help a programmer develop a program. Employ a range of these tools within at least one IDE</p> <p>1.5.2 Understand that computers often come with utility software, and how this performs housekeeping tasks including Encryption software, Defragmentation and data compression<br/>     Demonstrate the purpose of the identified utility software and why it is required</p> <p>1.6.1 Discuss that technology introduces ethical, legal, cultural, environmental and privacy issues:<br/>     Identify a variety of examples of digital technology and how this impacts on society<br/>     Discuss the impact of technology based around the issues listed<br/>     State the purpose of each piece of legislation and the specific actions it allows or prohibits (GDPR, CMA, Copyright, Design &amp; Patents Act)<br/>     Explain the need to license software and the purpose of a software licence<br/>     Identify the features of open source (providing access to the source code and the ability to change the software) and proprietary (no access to the source code, purchased commonly as off-the-shelf)<br/>     Analyse and recommend a type of licence for a given scenario including benefits and drawbacks</p> <p>REVISION &amp; ASSESSMENT 1</p> | <p>Re-evaluate key coding principles: sequence, selection, iteration</p> <p>2.1.3 Comprehend the main steps of each searching &amp; sorting algorithm (binary and linear search, bubble, Merge and Insertion sort) algorithm.<br/>     Identify any pre-requisites of an algorithm (eg binary only works in sorted lists).<br/>     Apply the algorithm to a data set and identify an algorithm if given the code for it<br/>     Develop practical programming skills:<br/>     • Design<br/>     • Write<br/>     • Test<br/>     • Refine<br/>     their programs in a HLL (Python).<br/>     Recognise key searching and sorting algorithms in code.</p> | <p>Identify many techniques to help them revise.<br/>     Access a use a wide range of resources<br/>     Focus their revision in class and at home on areas identified with their teacher<br/>     Reinforce the knowledge required to answer both papers. Provide accurate Python code in Paper 2. Access both exams with confidence and are able to apply their knowledge and exam techniques well.<br/>     Perform to the best of their ability in the exams, do not leave gaps, check their answers, use the time available.</p> <p>REVISION &amp; ASSESSMENT 2</p> | <p>Possess the knowledge required to answer both papers.<br/>     Create accurate Python code in Paper 2.<br/>     Apply their knowledge and exam techniques well, give them confidence in both exams.<br/>     Perform well in the exams, do not leave gaps, check their answers, use the time available.</p> |  |
|                                   |  |   |  |   |  |  |