

|  | Autumn 1  | Spring 2   | Summer 3   |
|--|---|--|--|
| <b>Content-</b> WHAT will be learned? What previous learning can be linked? Why this order/ <b>sequence</b> ?          | <p><b>Computer Systems</b><br/>Students learn about computer architecture by investigating the main components of a computer. They investigate different programming paradigms (low level and high level languages)</p> <ul style="list-style-type: none"> <li>• Transistors &amp; Logic Gates</li> <li>• Binary Adders</li> <li>• D-Type Flip-flop circuits</li> <li>• Multiplexer</li> <li>• Binary Data, conversions and operations</li> <li>• CPU architecture</li> <li>• FDE cycle</li> <li>• Use of Registers</li> <li>• CISC/RISC</li> <li>• Computer Hardware</li> <li>• Assembly language / LMC</li> <li>• Number Systems (Binary/Hexadecimal)</li> <li>• Binary Addition. Subtractions</li> <li>• Binary Shifts &amp; Binary Masks</li> <li>• Normalised Floating Point Representation</li> <li>• Boolean Logic and Boolean Algebra</li> <li>• Karnaugh Maps</li> <li>• Procedural Programming</li> <li>• Programming Constructs</li> <li>• Data Structures</li> <li>• Lists/Arrays</li> <li>• Hash Tables</li> <li>• Stacks &amp; Queues</li> <li>• Binary Trees</li> <li>• Graph</li> </ul> | <p><b>Computer Networks &amp; Web Technologies:</b></p> <p>Students develop their programming skills using client side technologies (HTML, CSS, JavaScript). They learn key concepts of web-based technologies including the structure and characteristics of a client/server web-based application.</p> <p>They investigate back-end database concepts (Relational databases, SQL).</p> <p>They investigate other approaches used to transfer data between applications (CSV, JSON, XML)</p> <p>They learn about network concepts:</p> <ul style="list-style-type: none"> <li>• LAN/WAN/VPN</li> <li>• Network topologies</li> <li>• Network Protocols</li> <li>• Wired/Wireless/Optical Communication methods</li> <li>• Network Hardware (Switch, Hub, WAP, NIC cards, Firewall, Router)</li> <li>• IP Networks (Packet Switching, TCP Stack, IP Protocols)</li> <li>• Web hosting concepts (IP Address, Domain names &amp; DNS servers, Cloud hosting)</li> <li>• Search engine indexing and page rank algorithm</li> <li>• Network security concepts (Threats and solutions to minimise these)</li> </ul> | <p><b>Software Concepts &amp; Software Development Methodologies</b><br/>Students investigate the different types of software (OS, Bios, Drivers, Utilities, device drivers, application Software, Translators, Virtual Machines), their purposes and characteristics.</p> <p>They learn about the main functions of an operating system including memory management (Paging vs segmentation) and scheduling algorithms.</p> <p>Students compare different software development methodologies (Waterfall cycle &amp; RAD methodologies)</p> <p>Students compare the characteristics of different programming paradigms (including an introduction to Prolog) and investigate the 4 stages of the compilation process.</p> <p>Students develop an understanding of Object Oriented Programming concepts and apply these concepts developing 2D arcade games using the PyGame library.</p> <p><b>Programming Project</b></p> <ul style="list-style-type: none"> <li>• Analysis</li> <li>• Design</li> <li>• Implementation</li> <li>• Testing</li> <li>• Evaluation</li> </ul> |
| <b>Skills-</b> What will be developed?   | <p>Reinforcement of <b>procedural programming concepts</b> using text based programming. (Python) (Sequencing, Iteration, Selection, use of variables &amp; data structures)</p> <p><b>Problem solving through Trial &amp; Error / Troubleshooting, Abstraction &amp; Decomposition</b></p> <p><b>Algorithmic Thinking using flowcharts</b></p> <p><b>Use of key data structures</b></p> <p>Problem solving using maths concepts including arithmetic calculations (percentages, MOD/DIV, areas and volumes, series, trigonometric formulas), x-y coordinates, Boolean logic.</p>   | <p>Programming skills using a range of programming languages and paradigms (incl. HTML, CSS, JavaScript, SQL,JSON, XML)</p> <p>Problem solving through Trial &amp; Error / Troubleshooting.</p> <p>Network Design skills: Ability to design a high level network design for a given business context, identifying the relevant hardware components required and a suitable network topology.</p>   | <p>Students get started with the programming project. They develop their research and analytical skills to complete the Analysis section of their coursework, identifying the scope of their project, the problems to focus on and the computational approaches used to solve these.</p> <p>Students further develop their programming skills using OOP programming concepts.</p>  |
| Key 'How'/'Why' Questions- What <b>powerful knowledge</b> will be gained? What areas/themes/concepts will be explored? | <p>Students make connections between different concepts covered in this unit and in previous years (GCSE level) They understand the correlation between computer hardware, CPU architecture, binary data, Boolean logic and computational approaches used in programming are all inter dependent.</p>   | <p>Students learn key network design concepts (Network hardware components, topologies, characteristics of LAN and WAN, factors impacting the performance of a network, network security concepts and key Internet concepts. Most computer systems nowadays are connected to a network and/or to the Internet. Understanding core network concepts is hence essential for anyone working on a modern computer system.</p>  | <p>Students investigate the different types of software (OS, Bios, Drivers, Utilities, device drivers, application Software, Translators, Virtual Machines), their purposes and characteristics.</p> <p>They learn about the main functions of an operating system including memory management (Paging vs segmentation) and scheduling algorithms.</p>   |

|  |  |   |   |
|--|--|---|---|
|  |  |   | <p>Students compare different software development methodologies (Waterfall cycle &amp; RAD methodologies)</p> <p>Students compare the characteristics of different programming paradigms (including an introduction to Prolog) and investigate the 4 stages of the compilation process.</p> <p>Students develop an understanding of Object Oriented Programming concepts and apply these concepts developing 2D arcade games using the PyGame library.</p> |
| <b>SEND-</b> how will support be seen? Seating plans? Simplified questions?                      | <ul style="list-style-type: none"> <li>• RAG Learning Grids</li> <li>• Video Clips to support home learning</li> <li>• Structured lessons and resources</li> </ul>   | <ul style="list-style-type: none"> <li>• RAG Learning Grids</li> <li>• Video Clips to support home learning</li> <li>• Structured lessons and resources</li> </ul>  | <ul style="list-style-type: none"> <li>• RAG Learning Grids</li> <li>• Video Clips to support home learning</li> <li>• Structured lessons and resources</li> </ul>  |
| <b>Assessment-</b> What? Why?  | <ul style="list-style-type: none"> <li>• Regular formative assessment using exam questions and mark schemes from the learning grid.</li> <li>• Homework tasks (Incl. online tests)</li> <li>• Mock Exams</li> </ul>  | <ul style="list-style-type: none"> <li>• Regular formative assessment using exam questions and mark schemes from the learning grid.</li> <li>• Homework tasks (Incl. online tests)</li> <li>• Mock Exams</li> </ul>                         | <ul style="list-style-type: none"> <li>• Regular formative assessment using exam questions and mark schemes from the learning grid.</li> <li>• Homework tasks (Incl. online tests)</li> <li>• Mock Exams</li> </ul>   |
| What <b>memory for learning</b> skills will be required- modelling? Concrete answers? Retrieval? | <ul style="list-style-type: none"> <li>• Teacher demonstrations</li> <li>• Students practice</li> <li>• Interleaving of key concepts, making connections between concepts covered in this unit at KS4.</li> </ul>  | <ul style="list-style-type: none"> <li>• PRIMM's approach: Predict-Run-Investigate-Modify-Make</li> <li>• Trial and Error</li> <li>• Teacher demonstrations</li> <li>• Students practice</li> <li>• Interleaving of key concepts</li> </ul> | <ul style="list-style-type: none"> <li>• Teacher demonstrations</li> <li>• Students practice</li> <li>• Interleaving of key concepts (OOP/Procedural programming)</li> </ul>  |
| <b>Literacy-</b> reading, extended accurate writing and oracy opportunities                      | <ul style="list-style-type: none"> <li>• Exam questions</li> <li>• Homework tasks</li> </ul>   | <ul style="list-style-type: none"> <li>• Exam questions</li> <li>• Homework tasks</li> </ul>  | <ul style="list-style-type: none"> <li>• Exam questions</li> <li>• Programming Project Documentation</li> </ul>   |
| <b>Numeracy/computing</b> skills   | <ul style="list-style-type: none"> <li>• Binary &amp; hexadecimal conversions including normalised floating point conversions</li> <li>• Boolean logic</li> </ul>  | <ul style="list-style-type: none"> <li>• RGBA colour codes</li> <li>• Relative vs. Absolute CSS Positioning</li> </ul>  | <ul style="list-style-type: none"> <li>• Real Physics concepts in video games</li> <li>• 2D/3D techniques used in video games</li> </ul>  |
| <b>Character</b> development   | <ul style="list-style-type: none"> <li>• Being resourceful and resilient when being exposed to challenging new concepts such as low level programming</li> <li>• Being kind and considerate, helping other students, working as a team for problem solving.</li> </ul> | <ul style="list-style-type: none"> <li>• Perseverance using a trial-and-error approach</li> <li>• Helping other students troubleshoot their code using C3B4Me</li> </ul>  | <ul style="list-style-type: none"> <li>• Being resourceful and resilient when being exposed to challenging new concepts such as low level programming</li> <li>• Being kind and considerate, helping other students, working as a team for problem solving.</li> </ul>  |
| <b>Equality/Diversity</b> opportunities  | <ul style="list-style-type: none"> <li>• The role of Alan Turing as a code breaker. (LGBTQ+)</li> <li>• The impact women at Bletchley Park</li> </ul>  | <ul style="list-style-type: none"> <li>• Job roles on web technologies (See video clips)</li> </ul>   | <ul style="list-style-type: none"> <li>• The role of Grace Hopper in the development of high-level programming languages (and the discovery of the first computer bug).</li> <li>• The work of Ada Lovelace on algorithm design.</li> </ul>   |
| <b>Homework/Independent</b> learning   | <p>Extra-Curricular Opportunities</p> <ul style="list-style-type: none"> <li>• See activities on Super Curriculum Grid.</li> <li>• Computing challenges (e.g edabit, Codecademy, w3schools)</li> </ul>   | <p>Extra-Curricular Opportunities</p> <ul style="list-style-type: none"> <li>• See activities on Super Curriculum Grid.</li> <li>• Computing challenges (e.g edabit, Codecademy, w3schools)</li> </ul>                                      | <p>Extra-Curricular Opportunities</p> <ul style="list-style-type: none"> <li>• See activities on Super Curriculum Grid.</li> <li>• Computing challenges (e.g edabit, Codecademy, w3schools)</li> </ul>  |
| <b>CIAG</b> coverage/links   | <ul style="list-style-type: none"> <li>• Careers in hardware engineering, electronics, software development.</li> </ul>  | <ul style="list-style-type: none"> <li>• Careers in Software Development, Web-designer, graphic designer, web-Author, copywriter, digital marketing expert, SEO expert, Network Design</li> </ul>   | <ul style="list-style-type: none"> <li>• IT Project Manager, Business Analyst, Solution Designer, Software Developer.</li> </ul>  |

|  | Autumn 2  | Spring 2   | Summer 3  |
|--|---|--|---|
| <b>Content-</b> WHAT will be learned? What previous learning can be linked? Why this order/ <b>sequence</b> ?          | <p><b>Abstraction, Decomposition &amp; Algorithmic Thinking</b></p> <p>Students understand the need for abstraction and decomposition when designing a complex computer system. They apply these concepts to a range of different scenarios and apply these concepts to complete the analysis and design stage of their coursework. (components of the problem, components of the solution, computational approaches used to solve problems, top modular design of their proposed solutions).</p> <ul style="list-style-type: none"> <li>• Abstraction</li> <li>• Decomposition</li> <li>• Algorithmic Thinking</li> <li>• Procedural Programming</li> <li>• OOP Programming</li> <li>• Hashing algorithms</li> <li>• Searching algorithms</li> <li>• Sorting algorithms</li> <li>• Iterative vs recursive algorithms</li> <li>• Data structures algorithms</li> <li>• Short Path Algorithms</li> <li>• Big O Notation</li> </ul> <p><b>Programming Project</b></p> <ul style="list-style-type: none"> <li>• Analysis</li> <li>• Design</li> <li>• Implementation</li> <li>• Testing</li> <li>• Evaluation</li> </ul> | <p><b>Applications of computer science:</b></p> <p>Students investigate different computational approaches and their use including:</p> <ul style="list-style-type: none"> <li>• Backtracking,</li> <li>• data mining,</li> <li>• heuristics,</li> <li>• performance modelling,</li> <li>• pipelining,</li> <li>• visualisation,</li> <li>• <i>machine learning</i>.</li> </ul> <p>They study the different legal acts relevant to Computer Science:</p> <ul style="list-style-type: none"> <li>• Data Protection Act (<i>and GDPR</i>)</li> <li>• Computer Misuse Act</li> <li>• Copyright Design &amp; Patent Act</li> <li>• Regulation of Investigatory Powers Act</li> </ul> <p><b>Programming Project</b></p> <ul style="list-style-type: none"> <li>• Analysis</li> <li>• Design</li> <li>• Implementation</li> <li>• Testing</li> <li>• Evaluation</li> </ul> | <p><b>Final Revisions – Getting ready for the summer examination focusing on all aspects of the course.</b></p> |
| <b>Skills-</b> What will be developed?   | <p>Students reinforce their programming skills using both <b>procedural &amp; OOP programming concepts</b> to complete their <b>programming project using an iterative approach. (RAD development cycle)</b>. They use a range of strategies to test (iterative and final testing) and troubleshoot their programs (Debugging tools) and to make their code more robust (e.g. Validation routines) and easier to maintain.</p> <p><b>Key Algorithms:</b></p> <p>Students study the key algorithms for:</p> <ul style="list-style-type: none"> <li>• Hashing algorithms</li> <li>• Searching algorithms</li> <li>• Sorting algorithms</li> <li>• Iterative vs recursive algorithms</li> <li>• Data structures algorithms</li> <li>• Short Path Algorithms</li> </ul> <p>They evaluate the complexity and effectiveness of a range of algorithms using the <b>Big O Notation</b>.</p>   | <p>Reinforcement of <b>procedural programming and OOP concepts</b>.</p> <p><b>Problem solving through Trial &amp; Error / Troubleshooting,</b></p> <p><b>Abstraction, Decomposition &amp; Algorithmic Thinking</b></p>   |   |
| Key 'How'/'Why' Questions- What <b>powerful knowledge</b> will be gained? What areas/themes/concepts will be explored? | <p>Abstraction, Decomposition and Algorithmic thinking are the essential skills required to work on larger IT projects. Students will also apply these skills when working on their computer science project.</p>   | <p>Students discuss the individual (moral), social (ethical) and cultural opportunities and risks of digital technology: Computers in the workforce, Automated decision making, Artificial intelligence, Environmental effects, Censorship and the Internet, Monitor</p>   |   |

|  |  |   |  |
|--|--|---|--|
|  |  | behaviour, Analyse personal information, Piracy and offensive communications, Layout, colour paradigms and character sets   |  |
| <b>SEND-</b> how will support be seen? Seating plans? Simplified questions?                      | <ul style="list-style-type: none"> <li>• RAG Learning Grids</li> <li>• Video Clips to support home learning</li> <li>• Structured lessons and resources</li> </ul>   | <ul style="list-style-type: none"> <li>• RAG Learning Grids</li> <li>• Video Clips to support home learning</li> <li>• Structured lessons and resources</li> </ul>  |  |
| <b>Assessment-</b> What? Why?  | <ul style="list-style-type: none"> <li>• Regular formative assessment using exam questions and mark schemes from the learning grid.</li> <li>• Homework tasks (Incl. online tests)</li> <li>• Mock Exams</li> </ul>  | <ul style="list-style-type: none"> <li>• Regular formative assessment using exam questions and mark schemes from the learning grid.</li> <li>• Homework tasks (Incl. online tests)</li> <li>• Mock Exams</li> </ul> |  |
| What <b>memory for learning</b> skills will be required- modelling? Concrete answers? Retrieval? | <ul style="list-style-type: none"> <li>• Teacher demonstrations</li> <li>• Students practice</li> <li>• Interleaving of key concepts, making connections between concepts covered in this unit at KS4.</li> </ul>  | <ul style="list-style-type: none"> <li>• Teacher demonstrations</li> <li>• Students practice</li> <li>• Interleaving of key concepts, making connections between concepts covered in this unit at KS5</li> </ul>    |  |
| <b>Literacy-</b> reading, extended accurate writing and oracy opportunities                      | <ul style="list-style-type: none"> <li>• Exam questions</li> <li>• Programming Project Documentation</li> </ul>  | <ul style="list-style-type: none"> <li>• The importance of accurate syntax when writing code</li> <li>• Programming Terminology (focusing on algorithm, Sequencing, Selection and Iteration)</li> <li>•</li> </ul>  |  |
| <b>Numeracy/computing</b> skills   | <ul style="list-style-type: none"> <li>• Logical and arithmetic concepts used in programming</li> <li>• Sorting Algorithms</li> <li>• Short Path Algorithms</li> </ul>   | <ul style="list-style-type: none"> <li>• Logical and arithmetic concepts used in programming</li> <li>• Data Visualisation</li> </ul>   |  |
| <b>Character</b> development   | <ul style="list-style-type: none"> <li>• Being resourceful and resilient when being exposed to challenging new concepts such as low-level programming</li> <li>• Being kind and considerate, helping other students, working as a team for problem solving.</li> </ul> | <ul style="list-style-type: none"> <li>• Perseverance using a trial-and-error approach</li> <li>• Helping other students troubleshoot their code using C3B4Me</li> </ul>  |  |
| <b>Equality/Diversity</b> opportunities  | <ul style="list-style-type: none"> <li>• Video Clips from Code.org showing successful IT specialists from a range of background (Women in STEM, Minority Ethnic groups)</li> </ul>   | <ul style="list-style-type: none"> <li>• Video Clips from Code.org showing successful IT specialists from a range of background (Women in STEM, Minority Ethnic groups)</li> </ul>                                  |  |
| <b>Homework/Independent</b> learning   | <p>Extra-Curricular Opportunities</p> <ul style="list-style-type: none"> <li>• See activities on Super Curriculum Grid.</li> <li>• Computing challenges (e.g edabit, Codecademy, w3schools)</li> </ul>   | <p>Extra-Curricular Opportunities</p> <ul style="list-style-type: none"> <li>• See activities on Super Curriculum Grid.</li> <li>• Computing challenges (e.g edabit, Codecademy, w3schools)</li> </ul>              |  |
| <b>CIAG</b> coverage/links   | IT Project Manager, Business Analyst, Solution Designer, Software Developer.   | Careers in Software Development, System Testing, IT Project Management  |  |