



Subject: **Computer Science**

Year: **10**

	What?	Why?	National Curriculum Links
Term 1-1	<p>1.4: Network Threats and Cyber Security This unit builds directly on Year 9 <i>Networks</i> and <i>Improving the Internet</i>, where pupils learned how data travels and how systems connect. It prepares pupils for GCSE Paper 1 (SLR 1.4) and supports later real-world thinking in <i>Ethics and Law</i>. Understanding threats also helps pupils appreciate safe programming practices, preparing them for Year 11 <i>Robust Programming</i>.</p>	<p>This unit builds directly on Year 9 <i>Networks</i> and <i>Improving the Internet</i>, where pupils learned how data travels and how systems connect. It prepares pupils for GCSE Paper 1 (SLR 1.4) and supports later real-world thinking in <i>Ethics and Law</i>. Understanding threats also helps pupils appreciate safe programming practices, preparing them for Year 11 <i>Robust Programming</i>.</p>	<p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p>
	<p>Python Advanced: String Functions Pupils use built-in string methods (e.g. len(), .upper(), .lower(), .find(), .replace(), slicing).</p>	<p>Builds on earlier input/output skills to manipulate text. Links to theory on character sets and representation.</p>	<p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p>
Term 1-2	<p>1.5: Systems Software Pupils explore the purpose and functionality of operating systems, including user interfaces, memory management, multitasking, peripheral management, user management, and file management. They also study utility software, including encryption software, defragmentation, data compression, and backup tools. Pupils learn how software supports hardware and how these systems affect performance and user experience.</p>	<p>This unit builds on <i>Memory and Storage</i> from Year 9, showing how operating systems manage these resources in practice. It prepares pupils for GCSE Paper 1 (SLR 1.5) and supports later units such as <i>Ethics and Law</i> by providing context for licensing and open vs proprietary systems. It also links forward to <i>Robust Programming</i> in Year 11 by highlighting the role of environments and utilities in maintaining reliable systems.</p>	<p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p>
	<p>Python Advanced: Arrays (1D Lists) Pupils declare, populate, and manipulate lists, accessing and updating values by index. Iteration through lists is introduced.</p>	<p>Expands on variables to allow structured data handling. Prepares for 2D arrays and reinforces theory on data structures.</p>	<p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p>
Term 2-1	<p>1.6: Ethical, Legal, Cultural and Environmental Pupils study the impact of digital technology on wider society, including ethical, legal, cultural, environmental, and privacy issues. They learn about legislation relevant to computing, such as the Data Protection Act 2018, the Computer Misuse Act 1990, the Copyright Designs and Patents Act 1988, and open-source vs proprietary software licences. Pupils also examine case studies of how technology affects individuals, businesses, and the environment.</p>	<p>This unit builds on <i>Ready Player One</i> in Year 8, where pupils were first introduced to ethical and societal impacts of technology. It prepares pupils for GCSE Paper 1 (SLR 1.6) and helps them make connections between technical knowledge and real-world consequences. It also supports cross-curricular learning with Humanities and PSHE by encouraging critical evaluation of technology's place in society.</p>	<p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p>
	<p>Python Advanced: Arrays (2D Lists) Pupils create and manipulate lists of lists, accessing values with multiple indices. Applications include grids, matrices, and tables.</p>	<p>Develops more advanced problem-solving and prepares pupils for tackling exam-style algorithm problems.</p>	<p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p>



Term 2-2	<p>2.1: Computational Thinking and Search/Sort Algorithms Pupils are introduced to the key principles of computational thinking: abstraction, decomposition, and algorithmic thinking. They learn to identify inputs, processes, and outputs, and to represent problems using structure diagrams. Pupils begin creating and refining algorithms using pseudocode and flowcharts. They also study linear and binary search algorithms, understanding their processes and efficiencies.</p>	<p>This unit builds on programming foundations in Year 8 (<i>Python PRIMM</i>) and prepares pupils for the practical programming work in Year 11. It links forward to GCSE Paper 2 (SLR 2.1) and underpins the entire algorithms and programming strand. By introducing standard algorithms at this stage, pupils gain confidence in problem-solving before tackling more complex sorts and program design in the next unit.</p>	<p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p> <p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p>
	<p>Python Advanced: Random Function Pupils use the random module to generate integers, floats, and selections.</p>	<p>Introduces unpredictability into programs, enabling simulations and games. Links to algorithm design in Paper 2.</p>	<p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p>
Term 3-1	<p>2.1: Sorting and Searching Algorithms Pupils extend their knowledge of algorithms by studying bubble sort, merge sort, and insertion sort. They explore how each algorithm works step by step and compare their efficiency. Pupils also practise designing algorithms using pseudocode and flowcharts, and they apply algorithmic thinking to solve increasingly complex problems.</p>	<p>This unit builds on <i>Computational Thinking and Algorithms</i> (Term 2-2) and prepares pupils for full programming implementation in Year 11. It also supports GCSE Paper 2 content by providing pupils with a solid grounding in common algorithms. The problem-solving and design focus also links back to <i>Modelling Data</i> (Year 7) and <i>Python Programming</i> (Year 8), showing continuity in computational skills development.</p>	<p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p> <p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p>
	<p>Python Advanced: File Handling Building experience with incremental Python projects and challenges</p>	<p>Builds on string manipulation and arrays, preparing pupils for more advanced program design. Links to databases and persistent storage in Year 11.</p>	<p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p>
Term 3-2	<p>2.1: Algorithms and Trace Tables Pupils develop strategies for identifying and correcting errors in algorithms and programs. They are introduced to trace tables to track the state of variables through an algorithm. Pupils also learn to identify and explain common logic and syntax errors, and how to refine algorithms to make them more efficient or effective.</p>	<p>This unit consolidates all algorithm learning in Year 10 and prepares pupils for programming in Year 11 (<i>Programming Fundamentals and Robust Programs</i>). It builds on <i>Scratch</i> (Year 7), <i>Python PRIMM</i> (Year 8), and the earlier algorithm units in Year 10. It also prepares pupils for GCSE Paper 2, where testing and refining algorithms is an essential skill for examination success.</p>	<p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p> <p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p>
	<p>SQL Pupils are introduced to SQL syntax, including SELECT, WHERE, AND/OR, ORDER BY, and simple queries across one or two tables.</p>	<p>Develops understanding of structured data and links to the databases section of J277 Paper 2. Even though not integrated into Python, it broadens pupils' awareness of data handling.</p>	<p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p>