

Subject: GCSE Computer Science **Year** 10 **Ability** Mixed

Term / Date(s)	Component 1.2	Components 2.1 and 2.2	Component 1.1	Component 2.1, 2.2 and 2.4	Component 1.6	Component 1.5
Topic	Memory and Storage	Computational thinking and programming fundamentals	System Architecture	Boolean logic and programming	Ethical, legal, cultural and environmental impacts	Programming Languages and Integrated Development Environments (IDE)
Topic overview	Students will explore how computers store data in binary digits, identify units of measurement for data, calculate file sizes, explore how binary data can be used to represent numbers, text, images, sound, and the need for compression when transmitting data.	Students will explore, through use of a high-level-language (Python) the use of sequencing, selection and iteration to solve problems in the form of algorithms and understand how abstraction, decomposition and algorithmic think can be used to define and refine problems.	Students will investigate the relationship between the Central Processing Unit (CPU), how it processes data and instructions and explore component parts, their roles and their impact upon the performance of computer systems.	Students will develop internal circuits to model how transistors in the CPU combine states to process binary data, as well as apply Boolean logic to operations within programming.	Students will investigate how technology impacts upon society; including legislation, cultural, ethical and environmental issues. By exploring these issues, students will be able to respond to wider issues where technology may have positive and negative impacts.	Students will explore the relationship between low-level languages (used by the CPU) and high-level languages used by programmers to develop software. Through this, students will investigate IDEs used to program and investigate common tools within them.
Pupils will learn...				Students will develop and implement algorithms that use Boolean logic to make decisions and process data.		

Subject: GCSE Computer Science **Year** 11 **Ability** Mixed

Term / Date(s)	Component 1.3	Component 1.4, 2.1 and 2.2	Component 1.5, 2.1 and 2.2	Components 2.1, 2.2 and 2.3	Components 2.1, 2.2 and 2.3	Component 1 and Component 2
Topic	Networks and Topologies	Network protocols and embedded systems, Computational thinking and programming fundamentals	System software, Computational thinking and programming fundamentals	Algorithms, programming techniques and producing robust programs	The programming project	Revision, Mastery and targeted revisit activities
Topic overview	Students will investigate types of networks (LAN and WAN / Peer to Peer and Client-Server), the hardware needed for each, how they can be represented in the form of topologies and the rules (protocols) that are used to govern the networks we use every day	Students will investigate the biggest threats to computer networks, the impact they can have and the best methods of protecting against them. They will also explore the use of hacking as an ethical, employment or malicious act.	Students will investigate how system software, such as operating systems are used to control all aspects of a computer system. They will also explore specific features that impact the security and performance of computer systems, such as encryption, defragmentation and compression.	Students will revisit, through use of a high-level-language (Python) the use of sequencing, selection and iteration to solve problems in the form of algorithms, understand how specific programming techniques (sorting, searching, functions, procedures and arrays) can improve the efficiency of a program and consider the impact of testing and defensive design on the quality of solutions.	Students will complete a programming project that requires them to:	Students will revisit all component areas in order to target misconceptions, refresh recall and close knowledge gaps. Topics are identified as below:
Pupils will learn...		Students will also revisit through use of a high-level-language (Python) the use of sequencing, selection and iteration to solve problems in the form of algorithms, as well as develop solutions to problems via flowcharts	Students will also revisit through use of a high-level-language (Python) the use of sequencing, selection and iteration to solve problems in the form of algorithms, as well as develop solutions to problems via flowcharts		<ul style="list-style-type: none"> analyse a computational problem produce planning in the form of flowcharts / pseudocode develop a test plan, use a wide variety of developed programming techniques evidence a solution and carry out testing. <p>This will enable students to show understanding of developing solutions and use key algorithmic thinking techniques, while using a high-level language to present evidence of a functioning program.</p>	<p>Component 1</p> <ul style="list-style-type: none"> System architecture Memory and storage Networks Network threats and system security System Software Legal, cultural, ethical and environmental impacts <p>Component 2</p> <ul style="list-style-type: none"> Algorithms Programming techniques Designing robust programs Boolean Logic Languages and IDEs