Queen Elizabeth's Girls' School

Educating Women of the Future

Founded in 1888

Computing Key Stage 4 Curriculum

	Topic	Focus	Enrichment
Year 10	Introduction to the Course	● Understanding the GCSE Computer Science course ● Setting expectations of classwork and homework ● Establishing classroom routines ● Introduction of resources: ○ Paul Long ○ Craig n Dave ○ Smart Revise ○ Google Classroom	- Weekly in lesson quizzes - Past paper practice - In Lesson exam style questions: Developing exam technique - Smart Revise - Home learning - KS4 Coding Club
	1.1 - Systems Architecture	1.1.1 Architecture of the CPU ● The purpose of the CPU: ○ The fetch-execute cycle ● Common CPU components and their function: ○ ALU (Arithmetic Logic Unit) ○ CU (Control Unit) ○ Cache ○ Registers ● Von Neumann Architecture ○ MAR (Memory Address Reigster) ○ MDR (Memory Data Register) ○ Program Counter ○ Accumulator 1.1.2 CPU performance ● How common characteristics of CPUs affect their performance ○ CLock speed ○ Cache size ○ Number of cores 1.1.3 Embedded Systems ● The purpose and characteristics of embedded system ● Examples of embedded systems	

	1.2 - Memory & Storage	1.2.1 Primary Storage (Memory) ● The need for primary storage ● The difference between RAM and ROM ● The purpose of ROM in a computer system ● The purpose of RAM in a computer system ● Virtual Memory 1.2.2 Secondary Storage ● The need for secondary storage ● Common types of storage ○ Optical ○ Magnetic ○ Solid State ● Suitable storage devices and storage media for a given application ● The advantages and disadvantages of different storage devices and storage media relating to these characteristics: ○ Capacity ○ Speed ○ Portability ○ Durability ○ Reliability ○ Cost 1.2.3 Units ● The units of data storage ○ Bit ○ Nibble (4 bits) ○ Byte (8 bits) ○ Kilobyte (1,000 bytes or 1 KB) ○ Megabyte (1000 KB) ○ Gigabyte (1000 MB) ○ Terabyte (1000 GB) ○ Petabyte (1000 TB) ● How data needs to be converted into binary format to be processed by a computer ● Data capacity and calculation of data capacity requirements 1.2.4 Data Storage **Numbers** ● How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa ● How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur ● How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa

		● How to convert binary integers to their hexadecimal equivalents and vice versa ● Binary shifts **Characters** ● The use of binary codes to represent characters ● The term 'character set' ● The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.: ○ ASCII ○ Unicode **Images** ● How an image is represented as a series of pixels, represented in binary ● Metadata ● The effect of colour depth and resolution on ○ The quality of the image ○ The size of an image file **Sound** ● How sound can be sampled and stored in digital form ● The effect of sample rate, duration and bit depth on ○ The playback quality ○ The size of a sound file 1.2.5 Compression ● The need for compression ● Types of compression ○ Lossy ○ Lossless
	1.3 - Computer Networks, Connections and Protocols	1.3.1 Networks and Topologies ● Types of Networks ○ LAN (Local Area Network) ○ WAN (Wide Area Network) ● Factors that affect the performance of networks ● The different roles of computers in a client-server and peer-to-peer network ● The hardware needed to connect stand-alone computers into a Local Area Network (LAN) ○ Wireless Access Point (WAP) ○ Routers ○ Switches ○ NIC (Network Interface Controller/ Card) ○ Transmission Media ● The Internet as a worldwide collection of computer networks: ○ DNS (Domain Name Server)

		○ Hosting ○ The Cloud ○ Web servers and clients ● Star and Mesh network topologies 1.3.2 Wired and Wireless Networks, Protocols and Layers ● Modes of connection ○ Wired ■ Ethernet ○ Wireless ■ Wi-fi ■ Bluetooth ● Encryption ● IP addressing and MAC addressing ● Standards ● Common protocols including ○ TCP/IP (Transmission Control Protocol/ Internet Protocol) ○ HTTP (Hyper Text Transfer Protocol) ○ HTTPS (Hyper Text Transfer Protocol Secure) ○ FTP (File Transfer Protocol) ○ POP (Post Office Protocol) ○ IMAP (Internet Message Access Protocol) ○ SMTP (Simple Mail Transfer Protocol) ● The concept of layers
	1.4 - Network Security	1.4.1 Threats to Computer Systems and Networks ● Forms of Attack ○ Malware ○ Social Engineering, e.g. phishing, people as the ‘weak point’ ○ Brute-force attacks ○ Denial of service attacks ○ Data interception and theft ○ The concept of SQL injection 1.4.2 Identifying and Preventing Vulnerabilities ● Common Prevention Methods ○ Penetration testing ○ Anti-malware software ○ Firewalls ○ User access levels ○ Passwords ○ Encryption ○ Physical Security

	1.5 - System Software	1.5.1 Operating Systems ● The purpose and functionality of systems software ○ User interface ○ Memory management/multitasking ○ Peripheral management and drivers ○ User management ○ File management 1.5.2 Utility Software: ● The purpose and functionality of utility software ● Utility system software ○ Encryption Software ○ Defragmentation ○ Data compression	
	1.6 - Ethical, Legal, Cultural, and Environmental Impact	● Impacts of digital technology on wider society including ○ Ethical issues ○ Legal issues ○ Cultural issues ○ Environmental issues ○ Privacy issues ○ How key stakeholders are affected by technologies ○ Environmental impact of Computer Science ○ Cultural implications of Computer Science ○ Open source vs proprietary software ● Legislation relevant to Computer Science: ○ The Data Protection Act 1998 ○ Computer Misuse Act 1990 ○ Copyright Designs and Patents Act 1988 ○ Software licences (i.e. open source and proprietary)	

	Topic	**Focus**	**Enrichment**
Year 11	2.1 - Algorithms	2.1.1 Computational Thinking: ● Principles of Computational Thinking ○ Abstraction ○ Decomposition ○ Algorithmic thinking 2.1.2 Designing, Creating and Refining Algorithms	- Weekly in lesson quizzes - Past paper practice - In Lesson exam style questions: Developing exam technique - Smart Revise

		● Identify the inputs, processes, and outputs for a problem
		● Structure diagrams
		● Create, interpret, correct, complete, and refine algorithms using
		  ○ Pseudocode
		  ○ Flowcharts
		  ○ Reference language/ high-level programming language
		● Identify common errors
		● Trace tables
		
		2.1.3 Searching and Sorting Algorithms
		● Standard searching algorithms
		  ○ Binary search
		  ○ Linear search
		● Standard sorting algorithms
		  ○ Bubble sort
		  ○ Merge sort
		  ○ Insertion sort
			- Home learning
			- KS4 Coding Club

	2.2 - Programming Fundamentals	2.2.1 Programming Fundamentals
		● The use of variables, constants, operators, inputs, outputs and assignments
		● The use of the three basic programming constructs used to control the flow of a program:
		  ○ Sequence
		  ○ Selection
		  ○ Iteration (count- and condition-controlled loops)
		● The common arithmetic operators
		● The common Boolean operators AND, OR and NOT
		
		2.2.2 Data Types
		● The use of data types:
		  ○ Integer
		  ○ Real
		  ○ Boolean
		  ○ Character and string
		
		2.2.3 Additional Programming Techniques
		● The use of basic string manipulation
		● The use of basic file handling operations
		  ○ Open
		  ○ Read
		  ○ Write
		  ○ Close
		● The use of records to store data
		● The use of SQL to search for data
		● The use of arrays (or equivalent) when solving problems, including both one-dimensional	

		(1D) and two-dimensional (2D) ● How to use subprograms (functions and procedures) to produce structured code ● Random number generation
	2.3 - Producing Robust Programs	2.3.1 Defensive Design ● Defensive design considerations: ○ Anticipating misuse ○ Authentication ● Input Validation ● Maintainability ○ Use of sub-programs ○ Naming conventions ○ Indentation ○ Commenting 2.3.2 Testing ● The purpose of testing ● Types of testing: ○ Iterative ○ Final/terminal ● Identify syntax and logic errors ● Selecting and using suitable test data ○ Normal ○ Boundary ○ Invalid/ Erroneous ● Refining algorithms
	2.4 - Boolean Logic	2.4.1 Boolean Logic ● Simple logic diagrams using the operators AND, OR and NOT ● Truth tables ● Combining Boolean operators using AND, OR and NOT ● Applying logical operators in truth tables to solve problems
	2.5 - Programming Languages and Integrated Development Environment	2.5.1 Languages ● Characteristics and purpose of different levels of programming language ○ High-level languages ○ Low-level languages ● The purpose of translators ● The characteristics of an assembler, a compiler and an interpreter 2.5.2 The Integrated Development Environments (IDE) ● Common tools and facilities available in an integrated development environment (IDE) ○ Editors ○ Error diagnostics

		○ Run-time environment ○ Translators	