SIR GRAHAM BALFOUR SCHOOL

CURRICULUM OVERVIEW – KEY STAGE 4 COMPUTING

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 10	**1.1 Systems Architecture** • The purpose of the CPU, how common characteristics of CPUs affect their performances, The function of the CPU as fetch and execute instructions stored in memory • Von Neumann architecture: MAR, MDR, PC, Accumulator • common CPU components and their function: ALU, CU, Cache. **1.2 Memory & Storage** • The difference between RAM and ROM • The purpose of ROM in a computer system • The purpose of RAM in a computer system • The need for virtual memory • Flash memory	**1.2 Memory & Storage cont.** • The need for secondary storage and Common types of storage • 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: • The units of data storage: How data needs to be converted into a binary format to be processed by a computer • Data capacity and calculation of data capacity requirements • How to convert positive denary whole numbers to binary numbers • How to add two binary integers together • 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 • 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 • ASCII, Unicode Images How an image is represented as a series of pixels, represented in binary Metadata, The effect of colour depth and resolution. **Sound** How sound can be sampled and stored in digital form ¨ The effect of sample rate, duration, and bit depth **Compression** Types of compression: Lossy, Lossless	**1.3 Networks and topologies** • Types of networks: LAN, WAN Factors that affect the performance of networks • The different roles of computers in a client-server and a peer-to-peer network • The hardware needed to connect stand-alone computers into a Local Area Network: Wireless access points, Routers, Switches, NIC. Transmission media • The Internet as a worldwide collection of computer networks: DNS, Hosting, The Cloud, Web servers and clients Star and Mesh network topologies	**1.4 Network Security** • Forms of attack • The concept of SQL injection • Common prevention methods **1.5 System Software** • The purpose and functionality of operating systems Peripheral management and drivers • User management • File management • The purpose and functionality of utility software • Utility system software	**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. • The Data Protection Act 2018 Computer Misuse Act 1990 Copyright Designs and Patents Act 1988 • Software licences (i.e., open source and proprietary)	**2.1 Algorithms** • Principles of computational thinking: Abstraction Decomposition, Algorithmic thinking.

Year 11

Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
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: o Sequence o Selection o Iteration The common arithmetic operators The common Boolean operators AND, OR and NOT. The use of data types: o Integer o Real o Boolean o Character and string o Casting The use of basic string manipulation The use of basic file handling operations: o Open o Read o Write o 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 arrays (2D) How to use sub programs (functions and procedures) to produce structured code Random number generation	**2.1 Algorithms** Principles of computational thinking: Abstraction Decomposition, Algorithmic thinking. **Practical Programming skills** All students must be given the opportunity to undertake a programming task or tasks during their course of study. The programming task(s) must allow them to develop skills within the following areas when programming: • Design • Write •Test •Refine Each task(s) must use one or more high-level text-based programming language, either to a specification or to solve a problem (or problems). They can use any high-level text-based programming language **Testing** The purpose of testing Types of testing: • Iterative • Final/terminal • Identify syntax and logic errors • Selecting and using suitable test data: o Normal o Boundary o Invalid/Erroneous o Refining algorithms	**Defensive design considerations** • Anticipating misuse • Authentication • Input validation • Maintainability: o Use of sub programs o Naming conventions o Indentation o Commenting **Languages** • Characteristics and purpose of different levels of programming language: o High-level languages o Low-level languages • The purpose of translators • The characteristics of a compiler and an interpreter **Integrated Development environment** • Common tools and facilities available in an Integrated Development Environment (IDE): o Editors o Error diagnostics o Run-time environment o Translators **Boolean logic** o Simple logic diagrams using the operators AND, OR and NOT Truth tables o Combining Boolean operators using AND, OR and NOT ¨ Applying lo solve problems	**Revision**	**Revision**	**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: o Sequence o Selection o Iteration The common arithmetic operators The common Boolean operators AND, OR and NOT. The use of data types: o Integer o Real o Boolean o Character and string o Casting The use of basic string manipulation The use of basic file handling operations: o Open o Read o Write o 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 arrays (2D) How to use sub programs (functions and procedures) to produce structured code Random number generation