The Ellesmere Port
C of E College

Computer Science

Part of the
Frank Field
Education Trust

KS4 Curriculum Plan 2022-23

		LP1	LP2	LP3	LP4	LP5
Year 10	TOPIC	*Boolean logic, units, computational thinking, memory and storage, algorithms.*	*Algorithms, memory and storage.*	*Programming fundamentals, systems architecture*	*Memory and storage, programming fundamentals*	*Computer networks, connctions and protocols*
	Knowledge	**Boolean Logic** 2.4.1 Understand simple logic diagrams using the operations AND, OR and NOT **Units** 1.2.3 The units of data storage Converting data to be processed by a computer **Memory and storage** 1.2.4 The units of data storage How data needs to be converted into a binary format to be processed by a computer How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa The use of binary codes to represent characters The relationship between the number of bits per character in a character set, and the number of characters which can be represented **Computational thinking** 2.1.1 **Algorithms** 2.1.2 Understand the principles of computational thinking including abstraction, decomposition and algorithmic thinking, how to design, create and refine algorithms.	**Algorithms** 2.1.2 Identify the inputs, processes, and outputs for a problem Know how to create, interpret, correct, complete and refine algorithms using pseudocode, flowcharts and reference language **Memory and storage** 1.2.4 & 1.2.5 Know that an image is represented as a series of pixels, represented in binary Metadata Understand the effect of colour depth and resolution on the quality of the image and the size of an image file Know how sound can be sampled and stored in digital forms Understand the effect of sample rate, duration and bit depth on the playback quality and the size of a sound file Understand the need for compression	**Programming fundamentals** 2.2.1 & 2.2.2 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 and iteration) The common arithmetic operators The common Boolean operators AND, OR, NOT Understand the need for various data types **System architecture** 1.1.1 & 1.1.2 & 1.1.3 Be able to state the function of the CPU (fetch and execute instructions stored in memory) Be able to describe the common CPU components and their function Explain computer systems. Explain memory and I/O devices. Explain the different buses. Explain the types of Peripherals Understand the Von Neumann architecture. Describe the MAR and MDR. Understand the importance of the Program Counter in the fetch-execute cycle.	**Memory and storage** 1.2.1 & 1.2.2 The need for primary storage The difference between RAM and ROM The purpose of RAM and ROM in a computer system The need for virtual memory **Programming fundamentals** 2.2.3 Basic string manupulation Understanding of basic file handling operations Using records to store data Using SQL to search for data Using arrays when solving problems	**Networks and topologies** 1.3.1 The characteristics of networks Understanding common network topologies Understanding the structure of URLs The role and function of the web browser **Wired and wireless networks, protocols and layers** 1.3.2 Hardware required Understanding the importance of networking standards The purpose of each layer in the OSI model Know the contents of a TCP/IP packet and packet switching Know the purpose of DNS servers and how they work Understand the threats to computer systems
	Skills	**Boolean Logic** - Drawing simple logic diagrams Creating truth tables Applying logical operators in truth tables to solve problems **Memory and storage** Calculating data capacity Adding two binary integers together (up to and including 8 bits) and explain overflow errors which may occur Converting positive denary whole numbers into 2-digit hexadecimal numbers and vice versa Converting from binary to hexadecimal equivalents and vice versa **Algorithms** Identifying the inputs, processes and outputs for a problem and creating structure diagrams.	**Algorithms** Create structure diagrams Identify common errors in algorithms Create trace tables **Memory and storage** Convert pixels into binary Calculate storage of image files and sound files depending on the size of the file	**Programming fundamentals** The use of constants and variables in a computer program Understanding programming constructs Writing program code Creating algorithms **Systems architecture** Describe the purpose of the CPU Describe the fetch-execute-cycle Explain a computer system Describe the purpose of the accumulator, ALU and CU Describe the PC in the FDE cycle.	**Memory and storage** Identifying the advantages and disadvantages of storage types Apply knowledge of storage devices to compare the three mediums of storage Apply the knowledge of storage devices to recommend an appropriate device. **Programming fundamentals** Using SQL to search for data Using arrays when solving problems **Practical programming skills**	**Networks and toplogies** Identifying the benefits and problems associated with networks List, describe, and compare the different types of networks depending on topology Compare the different types of communication models encountered in networks **Wired and wireless networks, protocols and layers** Usign the contemporary networking protocols in the 7-layer OSI model Routing traffic on a network Calculating routing costs
	Key Vocab	Binary, Boolean Logic, Converted, Algorithm, Linear, Merge, Pseudocode, Flowcharts, reference language	Assignment, Identifier, Variable, Constant, Concantenation, RAM, Memory locations, pixels, metadata, resolution, sample rate, playback	Von Neumann Architecture, MAR (Memory Address Register), MDR (Memory Data Register), Program Counter, Accumulator, ALU (Arithmetic Logic Unit), CU (Control Unit), Cache, Fetch/Execute, Buses, Decode, Storage, CPU	Storage, RAM, ROM, virtual memory, manipulation, operations, arrays	Topologies, networks, web browser, communication, protocols, traffic, routing

		LP1	LP2	LP3	LP4	LP5
Year 11	TOPIC	*Network security, Systems software, Ethical, legal, cultural and environmental impacts of digital technology*	*Producing robust programs, Programming languages and integrated design environments*	*Algorithms, Exam revision*	*Exam revision*	
	Knowledge	**Network security** 1.4.1 & 1.4.2 Threats to computer systems and networks Identifying and preventing vulnerabilities **Systems software** 1.5.1 & 1.5.2 Understanding of operating systems Understanding of utility software **Ethical, legal, cultural and environmental impact** 1.6.1 Understand the impacts of digital technology on wider society Understand the legislation relevant to computer science	**Producing robust programs** 2.3.1 & 2.3.2 Understanding defensive design Understanding Input validation Understanding maintainability The purpose of testing **Programming languages and integrated design environments** 2.5.1 & 2.5.2 Identifying the characteristics and purpose of different programming languages The purpose of translators Identifying the characteristics of a compiler and interpreter	**Algorithms** 2.1.3 Searching and sorting algorithms		
	Skills	**Network security** Identify forms of attack Select common prevention methods **Systems software** Explain the purpose and functionality of operating systems and utility software **Ethical, legal, cultural and environmental impact** Research skills to understand each issue and legislation	**Producing robust programs** Anticipating misuse Use of sub programs Use of naming conventions Identifying syntax and logic errors Selecting and using suitable test data **Programming languages and integrated design environments** Using common tools and facilities available in an integrated development environment	**Algorithms** Using bibary and linear searches Using bubble, merge and insertion sort		
	Key Vocab	Malware, social engineering, phishing, brute-force attack, interception, SQL injection, firewalls, passwords, encryption, physical security, interface, peripheral, defragmentation	Defensive design, maintainability, testing, translators, compiler, interpreter, syntax, logic, integrated development environment, diagnostics, run-time	Binary, linear, bubble, merge, insertion		