Curriculum Map

Sandringham School
'Everybody can be Somebody'

Subject: Computer Science

Year group: 10

There are 2 papers that students will sit to test their knowledge and understanding of how computer systems work (Paper 1) alongside designing, refining and creating solutions to given problems using computational methods through taught programming techniques (Paper 2).

Time period	Autumn	Spring	Summer
Content *Declarative Knowledge – 'Know What'*	**Paper 2 – Boolean Logic and Data representation** *2.4.1 Boolean Logic* • Knowledge of the truth tables for each logic gate • To create, complete or edit logic diagrams and truth tables for given scenarios. Work with more than 1 logic gate **Data representation** *1.2.3 Units:* • How data needs to be converted into a 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 between the different number bases i.e. binary, denary and hexadecimal numbers. • Understand the effect of a binary shift (both left or right) on a number. *Character:* • character set' using ASCII and UNICODE *Images:* • Pixels and their representation of a colour binary code representation. • The effect on image size and quality when changing colour depth and resolution • Metadata *Sound:* • Analogue sounds must be stored in binary, sound is sampled in hertz. Know that the number of bits required to store each sample is the bit rate. *1.2.5 – Compression:* • Advantages and disadvantages of each type of compression • Effects on the file for each type of compression	**Paper 1 - System Architecture, Memory and Storage:** **1.1 Systems Architecture and 1.2. Memory and storage** *1.1.1 Architecture of the CPU* • Von Neuman architecture i.e. MAR (Memory Address Register), MDR (Memory Data Register) Program Counter and Accumulator • The purpose of the CPU – The FDE (Fetch Decode Execute) cycle. • Students should be able to take a series of simple low-level instructions and understand how the processor uses the various general-purpose registers to do calculations during the FDE cycle. • How RAM and ROM interact with the CPU and what they contain during operation of the CPU *1.1.2 CPU Performance* • How common characteristics of CPUs affect their performance: Clock speed, cache size and number of cores *1.1.3 Embedded systems* • The purpose and characteristics of embedded systems *1.2.1 Primary storage* • Why computers have primary storage RAM and ROM • Key characteristics of RAM and ROM • Why virtual memory may be needed in a system *1.2.2 Secondary storage* • The need for secondary storage • The differences between each type of storage device/medium • Compare advantages/disadvantages for each storage device	

Curriculum Map

Sandringham School
'Everybody can be Somebody'

	Paper 2 – Algorithms and programming *2.1.2 - Designing, creating and refining algorithms* • Complete, write or refine an algorithm to solve problems • Identify syntax/logic errors in code and suggest fixes • Create and use trace tables to follow an algorithm *2.2.1 – Programming fundamentals and 2.2.2 Data types* • 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 using Sequence, Selection and Iteration (count- and condition-controlled loops) The common arithmetic operators along with the common Boolean operators AND, OR and NOT.	**Paper 2 – Algorithms and programming** *2.2.3 Additional programming techniques* • The use of basic string manipulation • The use of basic file handling operations: • Open, Read, Write and Close • The use of records to store data • The use of arrays (or equivalent) when solving problems, including both one-dimensional and two-dimensional arrays • How to use sub programs (functions and procedures) to produce structured code • Random number generation *2.3.2 Testing* • The purpose of testing • The different types of tests, i.e. iterative and final/terminal • Understand what is meant be a syntax error and a logic error. • Know and understand the different test types, i.e boundary, invalid, valid and erroneous.
Skills *Procedural Knowledge – 'Know How'*	**Paper 2 – Boolean Logic and Data representation** *2.4.1 Boolean Logic* • How to use and create simple logic diagrams using the operators AND, OR and NOT Truth tables • Combine Boolean operators using AND, OR and NOT • Apply logical operators in truth tables to solve problems **Data Representation:** *1.2.3 Units:* • How and why data must be stored in binary format • Calculate capacity of devices, required capacity for different types of files i.e. text, sound, image. *1.2.4 Number:* • How to convert positive denary whole numbers to binary numbers (1 byte) 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 *Character:* • How characters are represented in binary • How the number of characters stored is limited by the bits available • The differences between and impact of each character set and that character sets are logically ordered.	*1.1.1 Architecture of the CPU* • How the FDE cycle is accomplished. The effects of changing any of the common characteristics of CPU's on system performance, either individually or in combination • The links with memory (RAM) and how RAM is used. They will know the purpose and function of the MAR, MDR, PC, CU, ALU, Accumulator and be able to know how each register works during the FDE cycle. • The difference between storing data and an address • How RAM interacts with the CPU, how RAM and ROM is measured, their various uses *1.1.2 CPU performance* • How cache size, clock speed and cores affect the performance of the CPU. • The effects of changing any of the common characteristics on system performance, either individually or in combination. *1.1.3 Embedded systems* • Typical characteristics of embedded systems *1.2.1 Primary storage* • Why computers have primary storage RAM and ROM • How virtual memory works. *1.2.2 Secondary storage* • Understand how secondary storage is used and which secondary storage media is appropriate for a given set of circumstances • To know why computers, have secondary storage and to identify the differences between each type of storage. They will then apply their knowledge in context within scenarios

Curriculum Map

Sandringham School
'Everybody can be Somebody'

	Images: • Pixels represented in binary • The effect of colour depth and resolution on the quality and 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 and the size of a sound file *1.2.5 Compression:* • How and why there is a need for compression • The differences between lossy and lossless compression.	
	Paper 2 – Algorithms and programming *2.1.2 - Designing, creating and refining algorithms* • Know how to Identify the inputs, processes, and outputs for a problem • Understand pseudocode and flow diagrams to interpret, correct or complete algorithms • Abstract and decompose a problem to its simplest level, where code can be designed and written. Test and validate inputs and outputs from the program • Identify common errors • Use trace tables to predict the logic flow in an algorithm. *2.2.1 Programming fundamentals and 2.2.2 Data Types:* • How to use Python to solve problems using a range of operators, data structures and calling functions. • Understand pseudocode and flow diagrams to interpret, correct or complete algorithms • Abstract and decompose a problem to its simplest level, where code can be designed and written. Test and validate inputs and outputs from the program.	**Paper 2 – Algorithms and programming** *2.2.3 Additional programming techniques* • Practical use of the additional programming techniques in a high-level language with given problems. • Within their program code know how to manipulate strings using concatenation and slicing. • Design and write program code to manipulate both 1D and 2D array data structures to solve problems. Effectively use slicing on the arrays. • Know and understand the difference between functions and procedures and effectively use these in their program code solutions. *2.3.2 Testing* • Know how to write an effective test plan using suitable test data. • The difference between testing modules of a program during development and testing the program at the end of production
Key Questions	What is Boolean Logic? What are the different gates used? Why do computers use logic? What are transistors? Switches? What ae number bases? What is the binary number base? Denary? Hexadecimal? What effect does applying a left or right shift have on binary numbers? What is a character set? What is ASCII? What is extended ASCII? What is UNICODE? What is the difference between these character sets? What is a bitmap image? What is a pixel? What are vector graphics? What is meant by image bit depth? What is image resolution? What is sound sampling? Why do we measure amplitude of soundwaves when sampling? What do we mean by sampling frequency? What is audio bit depth? What do we mean by sound resolution? What is lossy and lossless compression? When do we use these? What is an algorithm? What are inputs, processes and outputs? What is a flowchart? What is pseudocode? What are the rules for pseudocode? What is a variable? What is a constant? When is it suitable to use a constant in a program in place of a variable? What are the programming data types? Where do we use them in a program? What is What are syntax and logic errors? What are the 3 different programming constructs? What is sequence? What is selection?	What are the Von Neumann and Harvard architectures? What are the different approaches the architectures take to storing instructions and data in memory? What are the benefits of each approach? What is the CPU and all the registers used in it? What is the purpose of the FDE cycle? How are the registers used within the FDE cycle? How are data and instructions executed? What is the purpose of RAM? ROM? How do the RAM and ROM interact with the CPU? Where do we use each? What do we mean by volatile? Non-Volatile? What is virtual memory? Why do we need virtual memory? What is clock speed? Cache size? Cores? How do each of these factors affect the overall performance of the CPU? What is a general-purpose device? What is an embedded device? What is secondary storage? What are the different types of secondary storage devices? What do we mean by magnetic storage? optical storage? Flash memory storage? What are the advantages and disadvantages of each type of storage in terms of portability, cost, capacity, speed, durability and reliability? What is string manipulation? What are the basic file handling operations? How do we apply these to strings? What is a data structure? What is a dynamic and static data structure? How do we manipulate arrays? What is a 1D or 2D array? What is a text file? What is an HTML file? How do I read, write and append to a

Curriculum Map

	What is an IF statement? What is a SWITCH/CASE statement? What is iteration? What is the difference between definite and indefinite iteration? What is a count-controlled loop? What is a condition-controlled loop? When is it appropriate to use each of these structures for a given problem? What is an infinite loop? How do we get out of an infinite loop? What is a counter variable? What is a subroutine? What is the difference between a function and a procedure? What is a parameter? What is a function definition? What is the function call?	text file? How can I manipulate the data from a text file? Why do we need to test our inputs? What are the 5 different types of tests? Why do we need to test erroneous data? What is boundary test data? What is success criteria? What is syntax error? What is a logic error? What are the differences between final and iterative testing?
Assessment	• A test in class based on past questions and on those provided by the exam board which are part of the end of unit test. Students will be given a grade based on published grade boundary data. Suggestions on how to improve answers to the next grade boundary will be provided. • Regular full exam papers and assessment practice. • HBL task focus on exam style questions • Exam questions will focus on command words examined by the OCR specification along with the marks allocated. • There will be some interim on-line testing based on multiple choice questions through different blended learning media. • Students will design algorithms and use these to practically implement code using the Python programming language both in the classroom and throughout HBL tasks.	
Literacy/Numeracy/ SMSC/Character	• Programming language literacy • Computational literacy • Exemplar modelling of answers • Understanding of key word definitions. • Scaffolded answers to LAQ, guided through AO1, AO2 and AO3 evaluative skills • Mathematical computation • Data handling • Linear Algebra • Discrete mathematics	

Curriculum Map

Sandringham School
'Everybody can be Somebody'

Subject: Computer Science

Year group: 11

There are 2 papers that students will sit to test their knowledge and understanding of how computer systems work (Paper 1) alongside designing, refining and creating solutions to given problems using computational methods through taught programming techniques (Paper 2).

Time period	Autumn	Spring	Summer
Content *Declarative Knowledge – 'Know What'*	**Paper 2 – Algorithms and programming** *2.2.1 Programming fundamentals & 2.2.2 Data types &2.2.3 Additional programming techniques & 2.3.2 Testing* • Students will undertake a mini style (non-examined) NEA practice to showcase their programming skills. **Paper 1 – 1.5 Systems software** *1.5.1 Operating systems and 1.5.2 Utility software* • The purpose and functionality of operating systems: User interface, Memory management and multitasking. Peripheral management and drivers, user management and file management • The utility system software on a computer system understanding typical routines such as defragmentation, firewall and backup. **Paper 1 – 1.3 Computer networks, connections and protocols** *1.3.1 Networks and topologies* • The types of networks – LAN and WAN • The topologies of a network – Star and Mesh. • The advantages and disadvantages of types of hardware and where they are needed in different circumstances. • Factors affecting performance such as bandwidth and number of devices. • The different roles of computers in a client-server and a peer-to peer network. • The Internet as a worldwide collection of computer networks	**Paper 2 – Searching and sorting algorithms** *2.1.3 searching and sorting algorithms* • The main algorithms to sort and search data, bubble, inser□on and merge sort, linear and binary searches. **Paper 1 – 1.3 Computer networks, connections and protocols** *1.3.2 Wired and wireless networks, protocols and layers* • The differences between wired and wireless connections • Encryption used on a network • IP addressing and MAC addressing • Communication standards and protocols • Layers used in protocols **Paper 2 – Algorithms and programming** *2.2.3 Additional programming techniques* • The use of records to store data • The use of SQL to search for data *2.3.1 Defensive Design* Defensive design considerations: • Anticipating misuse • Authentication • Input validation • Maintainability: use of sub programs, naming conventions, Indentation and commenting	**Paper 2 –Programming languages and Integrated Development Environments** *2.5.1. Languages* • Characteristics and purpose of different levels of High level and low level programming languages: • The purpose of translators • Transla□on of source code through interpreter and compila□on, 2.5.2 *The Integrated Development Environment (IDE)* • Common tools and facilities available in an Integrated • Development Environment (IDE): Editors, Error diagnostics, Run-time environment Translators • Techniques used to trap errors, types of errors. Breakpoints, stepping variable watches. **Paper 1 – Ethical, legal, cultural and environmental impact** • The laws that govern how computers, data and information can be used. • The moral and ethical dilemmas of computer use, including environmental impacts, social well-being. • Software licences (i.e. open source and proprietary)

Curriculum Map

Sandringham School
'Everybody can be Somebody'

Skills *Procedural Knowledge – 'Know How'*	**Paper 2 – Algorithms and programming** • Pupils will know how to design a solution to a given set of problems that will vary in their complexity • They will abstract and decompose a problem to its simplest level, where code can be designed and written. Test and validate inputs and outputs from the program. **Paper 1 – 1.5 Systems software** *1.5.1 Operating systems and 1.5.2 Utility software* • what each function of the operating system does their purpose and how they can be configured to maintain computer systems • How data is transferred between devices and the processor • Purpose of utility software and why they are required to maintain and repair computer systems. **Paper 1 – 1.3 Computer networks, connections and protocols** *1.3.1 Networks and topologies* • When building a network, what hardware and software is required and the weaknesses of the different types • Factors such as bandwidth and number of devices can impact network performance. • Clients make request for services into a client-server network. • DNS's convert IP addresses • Cloud computing services work along with its advantages and disadvantages	**Paper 2 – Searching and sorting algorithms** *2.1.3 searching and sorting algorithms* • Understand how the main sort and search algorithms are applied and in what circumstances they are useful. • Demonstrate the order in which they can reorganise data and why this might be useful in a given set of circumstances. **Paper 1 – 1.3 Computer networks, connections and protocols** *1.3.2 Wired and wireless networks, protocols and layers* • Compare the benefits and drawbacks of wired versus wireless connection • Encryption and why it is needed on a network. The principle of encryption to secure data across network connections • How the internet operates through routers, packet switching and IP addresses. • The internet and the protocols that make it work, with particular reference to the TCP/IP model and how each layer in the stack contributes a different protocol that allows for network and internet communication. POP3, HTTPS, HTTP, FTP, SMTP, IMAP **Paper 2 – Algorithms and programming** *2.2.3 Additional programming techniques* • Using SQL commands SELECT FROM WHERE to search data *2.3.1 Defensive Design* • Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values including invalid data • Authentication to confirm the identity of a user • Practical experience of designing input validation and simple authentication • Understand why commenting is useful and apply this appropriately	**Paper 2 –Programming languages and Integrated Development Environments** *2.5.1. Languages* • The differences between high- and low-level programming languages • The need for translators • The differences, benefits and drawbacks of using a compiler or an interpreter 2.5.2 *The Integrated Development Environment (IDE)* • Using their own practical knowledge of the tools that an IDE provides • How each of the tools and facilities can be used to help a programmer develop a program **Paper 1 – Ethical, legal, cultural and environmental impact** • Social, ethical and moral issues regarding environment, health and wealth. • Under what circumstances the law can be applied and the sort of activities they are • intended to combat. • The general problems encountered with the blind application of technology. • The need to license software and the purpose of a software licence
Key Questions	Why do devices need an operating system? What different tasks does the OS perform? How do operating systems manage memory? What are the different classifications of operating systems? What are the key features of each? What is the role of the BIOS in a computer system? What is meant by device drivers? Why are they needed for communication between hardware and the operating system? What is the purpose of applications? Utility? General purpose?	What is the importance of using protocols. What are network standards? What are the different layers within the TCP/IP stack? What is the purpose of each layer? What is MAC addressing? What is encryption? What is ciphertext? Plaintext? What is a public and private key? Why do we need to encrypt data? What is a searching algorithm? Why do we need to use searching algorithms? What is divide and conquer? Why does a list need to be sorted in order for a binary search to be carried	What is a High-level language? What is a low-level language? What is assembly code? What is a translator? What type of translators are there for High level languages? What type of translator do I use for a low-level language? What is a compiler? What is source code? What is object code? What is an executable file? What is an interpreter? What is an Assembler? What is an IDE? How can the IDE's

Curriculum Map

	Defragmentation? What is the definition and purpose of a network? What is a LAN and WAN? What is a network topology? What is bandwidth? Hoe does it affect network performance? How does the number of users affect the network performance? What are the networking hardware components used in a network? What is the difference between a switch and a hub? What is the internet protocol? What is a client-server network? What is a peer to peer network? What are the benefits and drawback of each? How is data transmitted on the Internet? How are IP addresses and packets used in the transfer of data? How is the Domain Name System is used to find the IP of a URL? What is the purpose, function, benefits and drawbacks of both packet and circuit switching?	out? What is a sorting algorithm? Why do we need to sort data? How does a bubble sort work? A merge and insertion sort? What is a record? What is SQL? Why do we need to use SQL in a database? What are the main keywords that we use in a SQL search query? What is defensive design? Why do we need defensive design in our programs? What is input sanitation? Authentication? Maintainability of code, Validation and verification? Where do we need to apply these to our program code?	development tools help in the development of a program? How can an IDE can be used to produce code? What are the range of features and tools that are within an IDE, that can be used to help produce and debug a program? What is meant by moral, social, ethical and cultural issues in relation to the use of computers. What are the moral implications of technology expansion? What about privacy? What about the social implication of using technology? Are there dangers in using social media? What are the ethical dilemas imposed by the advancement of technology? What are the cultural implication of the advancement and extended use of technology?
Assessment	• A test in class based on past questions and on those provided by the exam board which are part of the end of unit test. Students will be given a grade based on published grade boundary data. Suggestions on how to improve answers to the next grade boundary will be provided. • Regular full exam papers and assessment practice. • HBL task focus on exam style questions • Exam questions will focus on command words examined by the OCR specification along with the marks allocated. • There will be some interim on-line testing based on multiple choice questions through different blended learning media. • Students will design algorithms and use these to practically implement code using the Python programming language both in the classroom and throughout HBL tasks.		
Literacy/Numeracy/ SMSC/Character	• Programming language literacy • Computational literacy • Exemplar modelling of answers • Understanding of key word definitions. • Scaffolded answers to LAQ, guided through AO1, AO2 and AO3 evaluative skills • Mathematical computation • Data handling • Linear Algebra Discrete mathematics		