KS4 Curriculum Overview: Computing Year 11

Rationale: From year 9 and year 10 students have developed their computational skills and have written algorithms and have had the opportunity to design and create a small programme. They have developed their understanding of how computers work not just only in terms of components but also in the wider world and have had the opportunity to discuss moral and ethical issues and current developments in ICT. The students in year 11 revisit the topic again to cement understanding. This year will focus on key terminology and ensure that this is being applied appropriately. Student will be looking at exam questions and exam technique and how this applied to Computer science**.**

Term / Length of Unit	Outline	Assessment	Home Learning	Resources	Knowledge/Skills End Points	Reading
Autumn	**System Architecture 1.1** Recap the Purpose of the CPU and look at the Von Neumann with how all the components work. **Memory/Storage 1.2** Compare the different types of Primary and secondary memory and be able to recommend which component is the most suitable for a given situations	End of module assessment End of Module assessment	Craig&Dave - System Archecture.PPT Revision for end of assessment module Craig&Dave Memory&Storage.PPT Revision for end of assessment module Revise for end of assessment	• System Architecture Workbooks • System mind map • CSUK – Notes • CSUK SystemArchtecture.PPT • Cover sheet mapped to grades • Memory storage Work books. • Mindmap • CSUK – Notes • CSUK –Memory.PPT • Cover Sheet mapped to grades	1.1.1 • What actions occur at each stage of the fetch-execute cycle • The role/purpose of each component and what it manages, stores, or controls during the fetch-execute cycle • The purpose of each register, what it stores (data or address) • The difference between storing data and an address 1.1.2 • Understanding of each characteristic as listed • The effects of changing any of the common characteristics on system performance, either individually or in combination 1.1.3 • What embedded systems are • Typical characteristics of embedded systems • Familiarity with a range of different embedded systems 1.2.1 • Why computers have primary storage • Key characteristics of RAM and ROM • Why virtual memory may be needed in a system • How virtual memory works 1.2.2	Guided reading Key terms Computing Revision booklet Scenario based programming

					• Why computers have secondary storage • Recognise a range of secondary storage devices/media • Differences between each type of storage device/medium • Compare advantages/disadvantages for each storage device • Be able to apply their knowledge in context within scenarios 1.2.3 • Why data must be stored in binary format • Familiarity with data units and moving between each • Calculate capacity of devices • Calculate required capacity for a given set of files • Calculate file sizes of sound, images and text files 1.2.4 • Denary number range 0 – 255 • Hexadecimal range 00 – FF • Binary number range 00000000 – 11111111 • Understanding of the terms most significant bit, and least significant bit • Conversion of any number in these ranges to another number base • Ability to deal with binary numbers containing between 1 and 8 bits • Understand the effect of a binary shift (both left or right) on a number • 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 • Understand how character sets are logically ordered • Each pixel has a specific colour, represented by a specific code • The effect on image size and quality when changing colour depth and resolution	

	Algorithms Recap the computational methods that and the skills that students have developed when writing algorithms. Pull together the sorting and searching algorithms, students should be able to recognise the pseudocode for each of the algorithms and should be able complete an algorithm on a given set of date **Programming techniques** Re-examine the work completed in the practical project. Key concepts with the main functions. Linked to pseudocode and writing programme code	End of module 2.1 End of Module 2.2 Whole school PPE	Encourage the use of Smart revise throughout the term Craig&Dave Programming techniques.ppt Revision for PPE	• Algorithm Booklet • Algorithm a day • Videos demonstrating the different algorithms • Cover sheets mapped to grades Smart revise Programming techniques workbooks. CSUK – Notes	• Metadata stores additional image information • Analogue sounds must be stored in binary • Sample rate – measured in Hertz (Hz) • Duration – how many seconds of audio the sound file contains • Bit depth – number of bits available to store each sample	

					2.1.1 • Understanding of the principles and how they are used to define and refine problems 2.1.2 • Produce simple diagrams to show the structure of a problem and subsections and their links to other subsections • Complete, write or refine an algorithm using the techniques listed • Identify syntax/logic errors in code and suggest fixes • Create and use trace tables to follow an algorithm 2.1.3 • Understand the main steps of each algorithm • Understand any pre-requisites of an algorithm • Apply the algorithm to a data set • Identify an algorithm if given the code for it 2.2.1 • Practical use of the techniques in python within the classroom • Understanding of each technique • Recognise and use operators 2.2.2 • Practical use of the data types • Ability to choose suitable data types for data in a given scenario • Understand that data types may be temporarily changed through casting, and where this may be useful	

Spring	**Networks and Topologies** Look at the difference between WAN and LAN and apply to businesses. Look at the difference between peer to peer and client server with a deeper understanding of how they work – looking at examples from school and home. B **Wired and wireless networks and protocol layers** Look at the protocols and apply them to the TCP IP model. Be able to state the protocol and what they relate to. Group together the internet, mail and file transfers. From the topic before be able to identify the protocols that are associated with packet switching	End of module End of module End of module End of Module	Mindmap network Revision for end of module assessment PG – Online worksheet Revision for end of module PG – Online work sheet Revision for end of topic module	• CSUK Networks.PPT • Wireless and Wired Workbook • Cover sheet • CSUK Topolgies.ppt • Topologies workbook • Layers diagram and protocols map • Protocol song and words. • Cover sheet	1.3.1 • The characteristics of LANs and WANs including common examples of each • Understanding of different factors that can affect the performance of a network • The tasks performed by each piece of hardware • The concept of the Internet as a network of computer networks • A DNS's role in the conversion of a URL to an IP address • Concept of servers providing services • Concept of clients requesting/using services from a server • The Cloud: remote service provision • Advantages and disadvantages of the Cloud • Advantages and disadvantages of the Star and Mesh topologies • Apply understanding of networks to a given scenario 1.3.2 • Compare benefits and drawbacks of wired versus wireless connection • Recommend one or more connections for a given scenario • The principle of encryption to secure data across network connections • IP addressing and the format of an IP address (IPv4 and IPv6) • A MAC address is assigned to devices; its use within a network • The principle of a standard to provide rules for areas of computing	Guided reading Key terms Computing Revision booklet Case studies News articles

	Unit 1.5 System Software It is a theoretical unit covering the purpose and functionality of Operating systems and Utility software. 1.5.1 Operating Systems The unit explains the purpose and functionality of operating systems: • User interface • Memory management and multitasking • Peripheral management and drivers • User management • File management. 1.5.2 Utility Software The purpose and functionality of utility software:	End of module Whole School PPE		• CSUK SystemSecurity.ppt • System Security workbook • Cover sheet • CSUK SystemSoftware.ppt • System Software workbook • Cover sheet	• Standards allows hardware/software to interact across different manufacturers/producers • The principle of a (communication) protocol as a set of rules for transferring data • That different types of protocols are used for different purposes • The basic principles of each protocol i.e. its purpose and key features • How layers are used in protocols, and the benefits of using layers .5 All Pupils should be able to : • list the functions of an operating system: user interface, memory management, multi-tasking, peripheral management, user and file management • explain briefly what is meant by memory management and multi-tasking • describe briefly the purpose of encryption, defragmentation and data compression software Explain what each function of an operating system does • Features of a user interface • Memory management, e.g. the transfer of data between memory, and how this allows for multitasking. Understand that: • Data is transferred between devices and the processor • This process needs to be managed and what this entails (e.g. the use of buffers when transferring data to a printer) • User management functions, e.g: Allocation of an account, Access rights, Security, etc. • File management, and the key features, e.g.: Naming, Allocating to folders, Moving files, Saving, etc.	

	• Encryption software • Defragmentation • Data compression **Ethical, legal, cultural and environmental concerns** Bing together their own understanding of a range of computer technology issues and relate to a range of situation. Examine how it impacts. Should bring all the legislations together and what they are there for and		Essay question Revision for end of topic test Issue Boaster packs for extra revision over the Easter Holidays Revision for whole school PPE	How to write 9 mark questions Silent debate Word bank and sentence starters	1.6 All Pupils should be able to : • List some ethical, legal, cultural or environmental issues in relation to a given scenario • List one attribute and advantage of open source software and proprietary software • Understand technology introduces ethical, legal, cultural, environmental and privacy issues • Demonstrate knowledge of a variety of examples of digital technology and how this impacts on society • An ability to discuss the impact of technology based around the issues listed • Explain the purpose of each piece of legislation and the specific actions it allows or prohibits • The need to license software and the purpose of a software licence • Features of open source (providing access to the source code and the ability to change the software) • Features of proprietary (no access to the source code, purchased commonly as off-the-shelf. • Recommend a type of licence for a given scenario including benefits and drawbacks	
Summer	2.3 Producing Robust Programs 2.3.1 Defensive design.	End of module			2.3 Producing Robust Programs • Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values • Understanding of how to deal with invalid data in a program	Guided reading Key terms Computing Revision booklet

	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			Walkthrough examples • CSUK Programme Robust.ppt • CSUK - Notes • workbook • Cover sheet Walkthrough examples, maths question, some	• Practical experience of designing input validation and simple authentication e.g. password • Understand why commenting is useful and apply this appropriately. • The difference between testing modules of a program during development and testing the program at the end of production • Syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated • Logic errors as errors which produce unexpected output • Normal test data as data which should be accepted by a program • Boundary test data as data of the correct type • Invalid test data as data of the correct type but outside accepted • Erroneous test data as data of the incorrect type which should be rejected by a computer system • Ability to identify suitable test data for a given scenario and complete test plan 2.4 All Pupils should be able to : • Recognise standard symbols used to represent NOT, AND OR, NAND, NOR and XOR logic gates • Draw truth tables for the above logic gates	

	2.4 Boolean Logic Pupils will be taught Boolean logic diagrams and truth tables. Pupils will be taught how to create 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 and IDE 2.5.1 Programming Languages Pupils will be taught the characteristics and purpose of different levels of programming language: • High-level languages • Low-level languages • The purpose of translators • The characteristics of a compiler and an interpreter			example programming with the code.	• Complete a trace table to trace through a simple algorithm • Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios Ability to work with more than one gate in a logic diagram 2.5 All Pupils should be able to : • Explain the differences between high- and low-level programming languages • The need for translators • Explain the differences, benefits and drawbacks of using a compiler or an interpreter. • Demonstrate knowledge of the tools that an IDE provides • Explain how each of the tools and facilities listed can be used to help a programmer develop a program • Practical experience of using a range of these tools within at least one IDE	

	2.5.2 Integrated development Environment Pupils will be taught common tools and facilities available in an Integrated Development Environment (IDE): • Editors • Error diagnostics • Run-time environment Translators					