

KS3 Curriculum Map – Computer Science

Year 9 Transition Year	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Topic	Topic 1.1 Systems Architecture	Topic 1.2 Memory and Storage	Topic 1.3 Computer networks, connections and protocols	Topic 1.4 Network security	Year 9 Exam	Practical Programming
Components	<u>Component 1.1.1 Architecture of the CPU</u> 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 Register) - MDR (Memory Data Register) - Program Counter - Accumulator <u>Component 1.1.2 CPU Performance</u> How common characteristics of CPUs affect their performance: - Clock speed - Cache size - Number of cores <u>Component 1.1.3 Embedded Systems</u>	<u>Component 1.2.1 Primary Storage</u> 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 <u>Component 1.2.2 Secondary Storage</u> 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	<u>Component 1.3.1 Network & Topologies</u> Types of network: - 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 a peer-to-peer network The hardware needed to connect stand-alone computers into a Local Area Network: - Wireless access points - Routers - Switches - NIC (Network Interface Controller/Card) - Transmission media The Internet as a worldwide collection of computer networks: - DNS (Domain Name	<u>Component 1.4.1 Threats to computer systems and networks</u> 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 <u>Component 1.4.2 Identifying and preventing vulnerabilities</u> Common prevention methods: - Penetration testing - Anti-malware software - Firewalls - User access levels - Passwords - Encryption		Week 1: Introduction to Programming Understand the basics of programming, introduction to programming languages, and setup development environment. Resources: - Codecademy - Python Course (https://www.codecademy.com/learn/learn-python-3) - Python.org (https://www.python.org/) Week 2: Control Structures Learn about conditional statements and loops. Resource: - W3Schools - Python Conditions (https://www.w3schools.com/python/python_conditions.asp) - Khan Academy - Intro to JS: Drawing & Animation (https://www.khanacademy.org/computing/computer-programming/programming) Week 3: Functions Understand functions, their purpose, and how to use them. Resources: - Real Python - Defining Your Own Python Function

	The purpose and characteristics of embedded systems Examples of embedded systems	• Speed • Portability • Durability • Reliability • Cost <u>Component 1.2.3</u> <u>Units</u> The units of data storage How data need to be converted into a binary format to be processed by a computer Data capacity and calculation of data capacity requirements <u>Component 1.2.4</u> <u>Data Storage</u> 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 number into 2-digit hexadecimal numbers and vice versa How to convert binary integers to their hexadecimal equivalents and vice versa	Server) • Hosting • The Cloud • Web servers and client Star and Mesh network topologies <u>Component 1.3.2</u> <u>Wired and wireless networks, protocols and layers</u> Modes of connection: Wired • Ethernet o Wireless • Wi-Fi • Bluetooth Encryption IP addressing and MAC addressing Standards Commo 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	• Physical security	(https://realpython.com/defining-your-own-python-function/) - GeeksforGeeks - Python Functions (https://www.geeksforgeeks.org/python-functions/) Week 4: Data Structures (Lists, Tuples, and Dictionaries) Learn about different data structures and their uses. Resources: - Tutorialspoint - Python Lists (https://www.tutorialspoint.com/python/python_lists.htm) - Programiz - Python Dictionary (https://www.programiz.com/python-programming/dictionary) Week 5: Working with Strings Learn how to manipulate strings in Python. Resources: - W3Schools - Python Strings (https://www.w3schools.com/python/python_strings.asp) - HackerRank - Python String Practice (https://www.hackerrank.com/domains/tutorials/10-days-of-python) Week 6: File Handling Understand how to read from and write to files. Resources: - Real Python - Reading and Writing Files in Python (https://realpython.com/read-write-files-python/) - Programiz - File Handling in Python (https://www.programiz.com/python-programming/file-operation) Week 7:
--	--	---	---	---	---

		Binary shifts <u>Component 1.2.5</u> <u>Compression</u> The need for compression Types of compression: • Lossy • Lossless			Error Handling and Debugging Learn how to handle errors and debug code. Resources: - Python.org - Errors and Exceptions (https://docs.python.org/3/tutorial/errors.html) - JetBrains - Debugging in PyCharm (https://www.jetbrains.com/help/pycharm/debugging-your-first-python-application.html) Additional Practice Resources - LeetCode (https://leetcode.com/) - Coding challenges and algorithm practice. - HackerRank (https://www.hackerrank.com/) - Practice problems and competitions. - CodeWars (https://www.codewars.com/) - Practice coding with challenges. - GeeksforGeeks (https://www.geeksforgeeks.org/) - Comprehensive tutorials and problem sets. - Project Euler (https://projecteuler.net/) - Mathematical and computational problems. Notes: - Weekly quizzes to assess understanding. - Regular assignments and feedback sessions. - Encourage participation in coding forums and study groups.
Assessment Topics	What actions occur at each stage of the FDE cycle? The role and purpose of each component and what it stores, controls and manages during the FDE cycle The purpose of each register	Why computers have primary storage (memory) How this usually consists of RAM and ROM Key characteristics of RAM and ROM			

	The difference between storing data and an address What are embedded systems Typical characteristics of embedded system Familiarity with a range of different embedded systems	Why virtual memory may be needed in a system How virtual memory works Transfer of data between RAM and HDD when RAM is full				
Practical Programming	All students will be given the opportunity to undertake a programming task(s), either to a specification or to solve a problem (or problems), during their course of study. The practical programming assignment does not count towards the final GCSE grade.					
Link with GCSE skills	Paper 01 Mathematics & English	Paper 01 Mathematics & English	Paper 01 Mathematics & English	Paper 01 Mathematics & English		Mathematics & English
AO's	AO1 Demonstrate knowledge and understanding of the key concepts and principles of Computer Science. 3 AO2 Apply knowledge and understanding of key concepts and principles of Computer Science. AO3 Analyse problems in computational terms: - to make reasoned judgements - to design, program, evaluate and refine solutions					
Enrichment	- Coding Club – exploring in a practical way the aspects of designing/developing a program - Targeted support sessions for Paper 01 Computer Science					