Computer Science

Note-taking planne

✓	Excellent understanding	✓	Fair understanding	✓	More revision needed

Computer systems	the purpose of the CPU	Von Neumann architecture and its registers	common CPU components and their function	the function of the CPU as fetch and execute instructions stored in	how common characteristics of CPUs affect their performance		embedded systems: purpose and examples of embedded systems													
Memory	describe the difference between RAM and ROM	explain the need for ROM in a computer system	describe the purpose of RAM in a computer system	explain the need for virtual memory	describe flash memory															
Storage	the need for secondary storage	data capacity and calculation of data capacity requirements	Describe optical, magnetic and solid state storage	suitable storage devices and storage media for a given application	advantages and disadvantages of devices-capacity,speed,portability,durability,reliability, cost															

Wired and wireless networks	types of networks: ○ LAN (Local Area Network) ○ WAN (Wide Area Network)		describe the differences between a local area network and a wide area network such as the internet		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:		the internet as a worldwide collection of computer networks: ○ DNS (Domain Name Server) ○ hosting ○ the cloud		the concept of virtual networks.								
Network topologies, protocols and layers	explain the term IP addressing	explain the term MAC addressing	explain the term packet switching	explain the term protocol including:○ TCP/IP ○ HTTP○ HTTPS ○ FTP ○ POP○ IMAP ○ SMTP		explain the layers of TCP/IP	explain the need for security measures in networks, such as user access levels, suitable passwords and encryption techniques		Describe the difference between star and mesh network topologies	Wifi: ○ frequency and channels ○ encryption	compare Advantages and Disadvantages of Wi-Fi and Ethernet	describe and justify network policies								
Systems Security	Explain forms of attacks	Describe Malware, Phishing, People as weak point	Describe threats posed to networks :Brute force attacks, DOS, Data interception, sql injection, poor policy		identifying and preventing vulnerabilities: penetration testing, network forensics, network policies			identifying and preventing vulnerabilities: anti-malware, firewalls, user access levels, password, encryption												
Systems Software	the purpose and functionality of systems software	explain the need for a user interface and describe different types		explain the need for the following functions of an operating system: memory management, peripheral management, multi-tasking and s security			describe the purpose and use of common utility programs for computer security (antivirus, spyware protection and firewalls)			describe the purpose and use of common utility programs for disk organisation (formatting, file transfer, and defragmentation)			describe the purpose and use of common utility programs for system maintenance (Data compression and backup)							

Ethical, Legal, cultural and environmental concerns	how to investigate and discuss Computer Science technologies while considering: ◯ ethical issues ◯ legal issues ◯ cultural issues ◯ environmental issues. ◯ privacy issues				how key stakeholders are affected by technologies		Describe environmental impact of Computer Science		Describe cultural implications of Computer Science	Explain the difference between open source & proprietary software		Explain and identify the relevant legislation for a given scenario.								
Algorithms	computational thinking: ◯ abstraction ◯ decomposition ◯ algorithmic thinking		standard searching algorithms: ◯ binary search ◯ linear search		standard sorting algorithms: ◯ bubble sort ◯ merge sort ◯ insertion sort		how to produce algorithms using: ◯ pseudocode ◯ using flow diagrams		interpret, correct or complete algorithms.											
Programming Techniques	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 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 and two dimensional arrays		how to use sub programs (functions and procedures) to produce structured code		the use of data types: ◯ integer ◯ real ◯ Boolean ◯ character and string ◯ casting		the common arithmetic operators	the common Boolean operators			
Producing robust programs	Developing programs with input sanitisation/validation		defensive design considerations: ◯ planning for contingencies ◯ anticipating misuse ◯ authentication		Writing maintainable programs using comments and indentation		Explain the purpose of testing	Explain the purpose of testing	describe syntax errors and logic errors which may occur while developing a program		understand and identify syntax and logic errors	select and justify test data for a program, stating the expected outcome of each test.								

Computational Logic	Explain why data is represented in computer systems in binary form	Draw simple logic diagrams using the operations AND, OR and NOT	Create and complete truth tables for a given expression or circuit	Combining Boolean operators using AND, OR and NOT to two levels	Applying logical operators in appropriate truth tables to solve problems	Applying computing-related mathematics: ○ + ○ – ○ / ○ * ○ Exponentiation (^) ○ MOD ○ DIV								

Translators & Facilities	explain the difference between high level code and machine code	explain the need for translators to convert high level code to machine code	describe the characteristics of an interpreter	describe the characteristics of an assembler	describe the characteristics of a compiler	describe common tools and facilities available in an IDE: editors.	describe common tools and facilities available in an IDE: : error diagnostics	describe common tools and facilities available in an IDE: run-time environment	describe common tools and facilities available in an IDE: translators	describe common tools and facilities available in an IDE: auto-documentation.			

Data Representation	UNITS define the terms bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte	UNITS understand that data needs to be converted into a binary format to be processed by a computer.	NUMBER convert positive denary whole numbers (0-255) into 8-bit binary numbers and vice versa	NUMBER add two 8-bit binary integers and explain overflow errors which may occur	NUMBER convert positive denary whole numbers (0-255) into 2-digit hexadecimal numbers and vice versa	NUMBER convert between binary and hexadecimal equivalents of the same number	NUMBER explain the use of hexadecimal numbers to represent binary numbers.	CHARACTER explain the use of binary codes to represent characters	CHARACTER explain the term character set			

Data Representation	CHARACTER describe with examples the relationship between the number of bits per character in a character set and the number of characters which can be represented.	IMAGES explain the representation of an image as a series of pixels represented in binary	IMAGES explain the need for metadata to be included in the file such as height, width and colour depth	IMAGES discuss the effect of colour depth and resolution on the size of an image file.	SOUND explain how sound can be sampled and stored in digital form	SOUND explain how sampling intervals and other considerations affect the size of a sound file and quality of its playback.	INSTRUCTION explain how instructions are coded as bit patterns	INSTRUCTION explain how the computer distinguishes between instructions and data.			