

GCSE Computer Science (J277)

Knowledge Organisers

Unit 1: Computer Systems

System Architecture

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
The purpose of the CPU	- What the CPU is - The role of the CPU in a computer system	  
- Von Neumann architecture: - MAR (Memory Address Register) - MDR (Memory Data Register) - Program Counter - Accumulator	- The purpose of each register - What each register stores - How data and addresses are transferred between registers - The difference between storing data and an address	  
- Common CPU components and their function: - ALU (Arithmetic Logic Unit) - CU (Control Unit) - Cache	- The purpose of each component - What each component stores - How each component links to the other components in the CPU	  
The function of the CPU as fetch and execute instructions stored in memory	- What the fetch-execute cycle is - How it links to the components in the CPU and system	  
- How common characteristics of CPUs affect their performance: - clock speed - cache size - number of cores	- What clock speed, cache and cores are - How changes to each affects the performance of the CPU - What is meant by 'multi-core processor'	  
- Embedded systems: - Purpose of embedded systems - Examples of embedded systems.	- What an embedded system is - Examples of embedded systems	  

Memory and Storage

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
- Primary storage - Why computers have memory - The purpose of RAM and ROM in a computer system - The differences between RAM and ROM - What virtual memory is and how it works	- What RAM and ROM are - What the terms volatile and non-volatile mean - What is stored in ROM - The relationship between ROM and booting a computer up - What is stored in RAM - How increasing or decreasing the amount of RAM in a computer affects performance - What virtual memory is - How virtual memory works - The advantages and disadvantages of using virtual memory	  
- Secondary storage - Why computers need secondary storage - Main types of secondary storage (optical, magnetic, solid state) - Advantages & disadvantages of different storage devices based on characteristics (capacity; speed; portability; durability; reliability; cost) - Be able to recommend storage devices for different situations	- What secondary storage is - Why secondary storage is needed in a computer system (linked to system architecture) - The main features of optical, magnetic and solid state storage - The advantages and disadvantages of each storage method - A range of storage devices for each method - What is meant by each of the characteristics - How each characteristic affects the performance of storage devices - Be able to make justified recommendations for suitable storage devices in different scenarios	  
- Units - The units of data storage (bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte) - Why data needs to be stored in binary - Be able to calculate required storage capacity for given files - How to calculate file sizes of sound, images and text files	- What a range of data capacity values mean (bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte) - How to convert between different data capacities (e.g. megabytes to bits, etc) - Be able to calculate required capacity from files - How to calculate the file size of sound files - How to calculate the file size of image files - How to calculate the file size of text files	  

• Data storage • Convert from binary to denary and back • Add two binary integers together & explain what overflow errors are • Perform and explain right and left binary shifts • Characters sets • Differences between ASCII and Unicode • Images (metadata, colour depth) • Sound (sampling, sample rate, duration, bit depth)	• Be able to convert from binary to denary and back • Be able to add binary integers together • Be able to explain what an overflow error is and why it occurs • Be able to perform and explain left and right binary shifts • Be able to convert from denary to hex and back • Be able to convert from binary to hex and back • Explain what hexadecimal is • What base 2, base 10 and base 16 are • What check digits are and why they are used • Be able to convert characters to binary • Explain what a character set is • Explain why character sets being larger comes from higher bit size characters • The differences between ASCII, Extended ASCII and Unicode • Explain how images are stored • What metadata is and the role it plays • What colour depth is • How changing resolution and colour depth affects file size • What sample frequency, bit rate and sample size are • How sound is converted from analog to digital	  
• Compression • Why we need compression • Lossy and lossless compression	• Why compression is needed • Situations where compression is needed • The differences between lossy and lossless compression	  

Wired and Wireless Networks

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
Types of networks: LAN (Local Area Network) WAN (Wide Area Network)	• What a computer network is • The key features of LANs and WANs	  
Factors that affect the performance of networks	• Features which affect the performance of a network including (but not limited to): • Number of users • Bandwidth • Hardware • Outside interference	  
The different roles of computers in a client-server and a peer-to-peer network	• What is meant by peer-to-peer network • What is meant by client-server network • How peer-to-peer and client-server networks operate	  
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	• What each hardware component does • How each of the hardware components interacts with the other components	  
The internet as a worldwide collection of computer networks: DNS (Domain Name Server) Hosting The cloud	• What the internet is (from a network perspective) • The difference between the internet and the World Wide Web • What role DNS plays in the internet • What hosting is • How the cloud works	  
Star and mesh topologies	• What is meant by network topologies • What star and mesh topologies look like • The features of star and mesh topologies • The advantages and disadvantages of star and mesh topologies	  

Network Security

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
Forms of attack	Why network attacks happen	  
Threats posed to networks: Malware Phishing People as the 'weak point' in secure systems (social engineering) Brute force attacks Denial of service attacks Data interception and theft The concept of SQL injection Poor network policy	- What each threat is - How each threat could damage a computer network - How each threat can occur	  
Identifying and preventing vulnerabilities: Penetration testing Network forensics Network policies Anti-malware software Firewalls User access levels Passwords Encryption	- What each preventative measure is - How each measure can help to protect against network attacks - The benefits and drawbacks of each measure - Be able to select appropriate preventative measures (and justify selection) for different scenarios	  

System Software

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
The purpose and functionality of systems software	• What systems software is • Why systems software is important in the running of computer systems	  
Operating systems: User interface Memory management/multi-tasking Peripheral management and drivers User management File management	• What an operating system is/does • Why operating systems are important (link between hardware and user) • How each element of operating systems works and why it is important	  
Utility system software: Encryption software Defragmentation Data compression	• What utility software is • How it is used to maintain a computer system • What each piece of utility software does • How utility software benefits system performance • The differences between lossy and lossless compression	  

Ethical, Legal, Cultural & Environmental Impacts of Digital Technology

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
How to investigate and discuss Computer Science technologies while considering: Ethical issues Legal issues Cultural issues Environmental issues Privacy issues	- Why these issues are important to consider in Computer Science - An understanding of the issues surrounding each factor	😊 😐 😞
Environmental impact of Computer Science	- A range of negative environmental impacts of computers - How computers can help the environment	😊 😐 😞
Privacy issues related to Computer Science	A range of issues related to privacy which are created/solved by computers	
Cultural impact of Computer Science	- How the use of computers has changed people's behaviour - How computer systems need to adapt to be usable by different cultures - Why some people may not have access to computers	😊 😐 😞
Open source vs proprietary software	- What open source software is - What proprietary (closed source) software is - The benefits and drawbacks of each - A range of examples of open source and proprietary software	😊 😐 😞
Legislation relevant to Computer Science: The Data Protection Act 1998 Computer Misuse Act 1990 Copyright Designs and Patents Act 1988	- What each piece of legislation is designed to do - Be able to identify appropriate legislation for different scenarios	😊 😐 😞

Unit 2: Computational Thinking, Programming and Algorithms

Algorithms

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
- Computational Thinking - Abstraction - Decomposition - Algorithmic Thinking	- What is meant by computational thinking - What each computational thinking method is - How each method can be used to help solve problems	  
- Standard Searching Algorithms - Binary Search - Linear Search	- How each searching method works - Be able to use each searching method to search a data set for specific data - Be able to show the stages of searching a data set	  
- Standard Sorting Algorithms - Bubble Sort - Merge Sort - Insertion Sort	- How binary search works - How linear search works - Be able to use both methods to sort a data set - Be able to show the stages of sorting a data set using both methods	  
- How To Produce Algorithms Using: - Pseudocode - Flow Diagrams	- What is meant by pseudocode - How to write pseudocode algorithms - How to draw a flow diagram - What the main flow diagram shapes represent	  
Interpretation, Correction and Completion of Algorithms	- To be able to interpret algorithms to explain what is happening - To be able to give the output from an algorithm from a given input - To be able to identify and correct errors in algorithms - To be able to complete missing sections of algorithms accurately	  
Be able to identify syntax and logic errors in code	- What a syntax error is - What a logic error is - To be able to identify syntax and logic errors in code	  
Trace tables	- Be able to use a trace table to follow an algorithm - Know what a trace table is	

Programming Fundamentals

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
The use of Variables, Operators, Inputs, Outputs, Assignments	- What a variable is and how to declare one - The print function - How to use inputs (including specifying integer inputs) - How to assign values to variables	  
The use of the three basic programming constructs to control the flow of a program (Sequence, Selection, Iteration)	- What sequence, selection and iteration mean - For loops - While loops - IF statements	  
The use of basic string manipulation	- What is meant by string manipulation - How to slice strings	  
Casting	- Be able to use casting - Be able to explain why casting is useful	  
Random numbers	- Be able to create random numbers in a program - Be able to use random numbers in a program	  
The use of basic handling operations (Open, Read, Write, Close)	- How to open, read from, write to and close a file - Using file handling with other operators	  
The use of records to store data	How records work as a data structure	  
The use of SQL to search for data	- What SQL is - How to structure and write SQL statements - What the terms SELECT, FROM, WHERE mean	  
The use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays	- What an array is - How to structure 1D and 2D arrays - How to retrieve data from an array	  
How to use sub-programs (functions & procedures) to produce structured code	- What sub-programs are - How to write a function - How to write a procedure	  
The use of data types (integer, real, Boolean, character/string, casting)	- The characteristics of each data type - Be able to select appropriate data types for different uses in algorithms	  
The common arithmetic operators	Be able to use each arithmetic operator in algorithms (+, -, /, *, DIV, MOD)	  
The common Boolean operators	Be able to use each Boolean operator in algorithms (AND, NOT, OR)	  

Producing Robust Programs

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
- Defensive design considerations - Input sanitisation/validation - Planning for contingencies - Anticipating misuse - Authentication	- What input sanitisation does - How to validate inputs - Why it is important to plan for contingencies - How to authenticate inputs	  
- Maintainability - Comments - Indentation	- Why maintainability is important in programming - How comments and indentation help with maintainability	  
The purpose of testing	- Why testing is important - The potential issues created by not testing effectively	  
- Types of testing - Iterative - Final/terminal	The differences between iterative and final testing	  
Selecting and using suitable test data	- How to select and use suitable data for different situations - Different test data types (Normal; Boundary; Invalid/Erroneous) - Be able to create a test plan	  

Boolean Logic

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
Why data is represented in computer systems in binary form	- What binary is - Why binary is used in computer systems	  
Simple logic diagrams using the operations AND, OR and NOT truth tables	- How to draw logic diagrams for AND, OR and NOT gates - What a truth table is and what it does - How to complete truth tables from logic diagrams	  
Combining Boolean operators using AND, OR and NOT to two levels	How to combine logic diagrams which use more than one of the Boolean operators	  
Applying logical operators in appropriate truth tables to solve problems	How to complete truth tables from logic diagrams	  

Programming Languages & IDEs

Knowledge Organiser

Topics	What do I need to know?	How Well Do I Understand It?
Characteristics and purpose of offering levels of programming language, including low level languages	- The differences between high and low-level programming languages - A range of examples of each	  
The purpose of translators	- What translators do - Why they are important in running programs	  
The characteristics of an assembler, a compiler and an interpreter	- What compilers, assemblers and interpreters do - Benefits and drawbacks of using a compiler or an interpreter	  
- Common tools and facilities available in an integrated development environment (IDE): - Editors - Error diagnostics - Run-time environment - Translators	- What an IDE is - What each of the characteristics does - How each of the characteristics helps programmers to develop programs effectively	  