GCSE Computer Science Vocabulary

Chapters Overview

Chapter 1: Computational Thinking & Algorithms & Chapter 6: Programming Fundamentals

- Algorithm – A step-by-step set of instructions to solve a problem.
- Decomposition – Breaking down a problem into smaller, manageable parts.
- Abstraction – Removing unnecessary details to focus on the essential.
- Pseudocode – A way to plan code using structured English.
- Flowchart – A diagram that represents an algorithm visually.
- Iteration – Repeating a set of instructions (e.g., for, while loops).
- Selection – Making decisions in code (e.g., if, else).
- Linear Search – A method to find an item in a list by checking each element.
- Binary Search – A faster search method that splits the list in half each time.
- Merge Sort / Bubble Sort – Algorithms for sorting data.
- Variable – A named storage location for data.
- Constant – A value that does not change during program execution.
- Data Types – e.g., Integer, Float, Boolean, String.
- Operators – Arithmetic (+, -, *, /), Relational (==, !=, <, >), Logical (AND, OR, NOT).
- Subroutine – A reusable block of code (function or procedure).
- Array/List – A collection of elements stored under one name.
- String Manipulation – Operations like slicing, concatenation, length.
- File Handling – Reading from and writing to files.
- Syntax Error – Mistake in the code that breaks the rules of the language.
- Logic Error – Code runs but produces incorrect results.
- Runtime Error – Error that occurs while the program is running.

Practical Programming Skills

- Python Syntax – Focus on Edexcel's Programming Language Subset (PLS).
- Input/Output – input(), print()
- Iteration – for, while
- Selection – if, elif, else
- Functions – def, parameters, return values
- Testing – Test plans, trace tables, debugging
- Comments – Using # to explain code

Chapter 2: Data Representation

- Binary (Base-2) – Used by computers; only 0s and 1s.
- Denary/Decimal (Base-10) – Base-10 number system, uses 0-9
- Hexadecimal (Base-16) – Compact representation of binary, uses 0–9 and A–F.
- Nibble – 4 bits.
- Byte – 8 bits.
- Binary Addition – Adding binary numbers using carry rules.
- Overflow Error – When a calculation exceeds the maximum value a register can hold.
- ASCII – 7-bit character set for English characters.
- Extended ASCII – 8-bit version with 256 characters.
- Unicode – Universal character set supporting multiple languages.
- Pixel – Smallest unit of a digital image.
- Resolution – Number of pixels in an image (width × height).
- Colour Depth – Number of bits used per pixel (e.g., 8-bit, 24-bit).
- Metadata – Data about data (e.g., image width, height, colour depth).
- Sample Rate – Number of samples per second (Hz).
- Bit Depth – Number of bits per sample.
- Sample – A measurement of sound amplitude at a point in time.
- Sound File Size Calculation – Sample rate × bit depth × duration × channels.
- Image File Size Calculation – Resolution x bit depth
- Lossy Compression – Removes data permanently (e.g., JPEG, MP3).
- Lossless Compression – Retains all original data (e.g., PNG, ZIP).
- Run Length Encoding (RLE) – Compresses repeated data.
- Two's Complement – A method for representing negative numbers in binary.
- MSB (Most Significant Bit) – In two's complement, this bit indicates the sign (0 = positive, 1 = negative).
- Unsigned Binary – Represents only positive numbers (e.g., 8-bit unsigned max = 255).
- Signed Binary – Uses the most significant bit (MSB) as a sign bit:
- Left Shift (<<) – Moves all bits to the left, multiplying the number by 2 for each shift.
- Right Shift (>>) – Moves all bits to the right, dividing the number by 2 for each shift (integer division).
- Logical Shift – Fills empty bits with 0s (used for unsigned numbers).
- Arithmetic Shift – Preserves the sign bit (used for signed numbers). Binary Addition – Adding binary numbers using rules:
- Overflow Error – Occurs when the result of a binary addition exceeds the number of bits available (e.g., 8-bit register can't store a 9-bit result).

Chapter 3: Computer Systems

- CPU (Central Processing Unit) – The component where most of the processing are done.
- Fetch-Decode-Execute Cycle – The process the CPU uses to run instructions.
- Volatile Memory – Data stored on these components are lost once there is no power.
- Non-Volatile Memory – Data stored on these components are not lost once there is no power.
- RAM (Random Access Memory) – A type of Volatile memory.
- ROM (Read-Only Memory) – A type of Non-Volatile memory.
- Secondary Storage – e.g., HDD, SSD, Optical, Cloud.
- Embedded Systems – Computers built into other devices.
- Input/Output Devices – e.g., Keyboard, Mouse, Monitor, Printer.
- ALU (Arithmetic Logic Unit) – Performs calculations and logic.
- CU (Control Unit) – Directs operations of the processor.
- Registers – Small, fast memory locations in the CPU.
- Accumulator – Stores intermediate results.
- Program Counter (PC) – Holds the address of the next instruction.
- Memory Address Register (MAR) – Holds memory addresses.
- Memory Data Register (MDR) – Holds data being transferred.
- Fetch – Retrieve instruction from memory.
- Decode – Interpret the instruction.
- Execute – Carry out the instruction.
- Virtual Memory – Uses part of the hard drive as RAM when full.
- Cache – Small, fast memory close to the CPU.
- Primary Storage – RAM, ROM.
- Secondary Storage – HDD, SSD, Optical (CD/DVD), Flash.
- Cloud Storage – Remote storage accessed via the internet.
- Embedded System – A computer system within a larger device (e.g., washing machine).
- Code Readability – how easy the code can be read
- Indentation – Structuring code with consistent spacing to improve readability.
- Commenting – Adding notes in code to explain logic or purpose (e.g., using # in Python).
- Naming Conventions – Using meaningful and consistent names for variables, functions, and classes (e.g., camelCase, snake_case).
- Code Formatting – Writing code in a clean, organised way (e.g., spacing, line breaks).
- Test Plan – A structured approach to testing a program's functionality.
- Trace Table – A table used to track variable values through an algorithm.
- Dry Run – Manually stepping through code to check logic.
- Debugging – Identifying and fixing errors in code.
- Unit Testing – Testing individual components or functions of a program.
- Modular Programming – Breaking a program into smaller, reusable functions or modules.

- DRY Principle (Don't Repeat Yourself) – Avoiding code duplication by using functions or loops.
- Code Reusability – Writing code that can be used in multiple places or projects.
- Input Validation – Checking user input to prevent errors or malicious data.
- Error Handling – Managing unexpected inputs or failures
- Documentation – Written descriptions of code, functions, and usage.
- Version Control – Tracking changes to code over time (e.g., using Git).
- Code Review – Having others examine code for quality and correctness.
- Source Code – The human-readable instructions that make up a program.
- Proprietary Software – Software that is owned by an individual or company and not freely available to modify or distribute.
- Open Source – Software with publicly available source code that can be modified and shared.
- Peer Review – A process where fellow developers review code for quality, readability, and correctness before it is merged.
- Expert Review – A more formal review by a senior or specialist developer, often used for critical or complex code.

Chapter 4: Networks

- LAN (Local Area Network) – Covers a small area (e.g., home, school).
- WAN (Wide Area Network) – Covers a large geographical area.
- Client-Server Model – Clients request services from a central server.
- Peer-to-Peer (P2P) – Devices share resources without a central server.
- Router – Connects networks and directs data.
- Switch – Connects devices in a LAN.
- Star Topology – All devices connect to a central hub or switch. Easy to manage but single point of failure.
- Bus Topology – All devices share a single backbone cable. Cheap but prone to collisions.
- Mesh Topology – Every device connects to every other. Very reliable but expensive and complex.
- Hybrid Topology – Combination of two or more topologies.
- NIC (Network Interface Card) – Enables a device to connect to a network.
- WAP (Wireless Access Point) – Connects wireless devices to a network.
- Packet – A formatted unit of data carried by a network.
- Packet Switching – Data is split into packets and sent independently across the network.
- Circuit Switching – A dedicated path is established for the duration of a communication session.
- Protocol – Rules for data transmission (e.g., TCP/IP, HTTP, FTP).
- Star Topology – All devices connect to a central hub or switch. Easy to manage but single point of failure.
- Bus Topology – All devices share a single backbone cable. Cheap but prone to collisions.
- Ring Topology – Devices connected in a circle. Data travels in one direction. Failure in one device can affect the whole network.
- Mesh Topology – Every device connects to every other. Very reliable but expensive and complex.
- Hybrid Topology – Combination of two or more topologies.
- IP Address – Unique identifier for a device on a network
- Bandwidth – Maximum data transfer rate.
- Latency – Delay in data transmission.

Chapter 5: Cyber Security & Ethical and Environmental Issues

- Threats – Malware, phishing, brute force, denial of service (DoS).
- Malware – Malicious software designed to harm or exploit systems.
- Virus – Attaches to files and spreads when opened.
- Worm – Self-replicates and spreads without user action.
- Trojan Horse – Disguised as legitimate software.
- Ransomware – Encrypts files and demands payment.
- Spyware – Secretly gathers user information.
- Adware – Displays unwanted ads, may track user behaviour.
- Phishing – Fraudulent emails or messages tricking users into revealing personal data.
- Shoulder Surfing – Observing someone's screen or keyboard to steal information.
- Brute Force Attack – Trying many password combinations to gain access.
- Denial of Service (DoS) – Overwhelms a system to make it unavailable.
- Data Interception – Capturing data as it travels across a network.
- Social Engineering – Manipulating people to gain confidential info.
- Firewall – Filters incoming and outgoing traffic to block threats.
- Antivirus Software – Detects and removes malware.
- Encryption – Converts data into unreadable form without a decryption key.
- Authentication – Verifies identity (e.g., passwords, biometrics, 2FA).
- Access Control – Restricts user access based on roles or permissions.
- User Education – Training users to recognise threats like phishing.
- Penetration Testing – Simulated attacks to find vulnerabilities.
- Network Policies – Rules for acceptable use and security practices.
- Automatic Software Updates – Patches vulnerabilities in software.
- Backup and Recovery – Regularly saving data to restore after an attack.
- Data Protection Act (2018) – UK law ensuring personal data is used fairly and securely.
- Computer Misuse Act (1990) – Criminalizes unauthorized access to computer systems.
- Copyright, Designs and Patents Act (1988) – Protects intellectual property.
- GDPR (General Data Protection Regulation) – EU regulation on data privacy and protection.
- Privacy – Right to control personal data and digital footprint.
- Digital Divide – Inequality in access to technology and the internet.
- AI Bias – Algorithms making unfair decisions due to biased data.
- Surveillance – Monitoring of individuals, raising privacy concerns.
- Ethical Hacking – Testing systems to improve security.
- E-Waste – Discarded electronic devices contributing to pollution.
- Energy Consumption – High power usage of data centres and devices.
- Water Consumption – Water used to create digital devices
- Blood mineral – gathering of previous mineral through exploitation.
- Sustainability – Designing systems with minimal environmental impact.