

# Computer Systems

<b>Computer System</b>	A combination of hardware and software components that allow input, processing and output of data.	●
<b>Hardware</b>	The physical components that make up a computer system.	●
<b>Software</b>	The programs that run on a computer system.	●
<b>Input Devices</b>	Hardware devices that take real-world analogue data and convert it into a digital form that can be stored on a computer e.g. keyboard, mouse, microphone, webcam, scanner etc.	●
<b>Output Devices</b>	Hardware devices that use digital data from a computer to produce a form that is understandable to a person e.g. monitors, printers, speakers, projectors etc.	●

## Units

<b>Bit</b>	A single binary digit: 1 or 0	●
<b>Byte</b>	8 bits	●
<b>Nibble</b>	4 bits	●
<b>Kilobyte</b>	1024 bytes	●
<b>Megabyte</b>	1024 kilobytes	●
<b>Gigabyte</b>	1024 megabytes	●
<b>Terabyte</b>	1024 gigabytes	●

## Numbers

<b>Binary</b>	Base 2 number system. Used by computers and uses the digits 1 and 0.	●
<b>Denary</b>	Base 10 number system. How human normally count and uses the digits 0 to 9.	●
<b>Hexadecimal</b>	Base 16 number system. Used by humans to represent groups of four bits at a time and uses digits 0 to F.	●
<b>Overflow</b>	When the result of a numeric calculation is too large to be stored in the space reserved for that type of data.	●

# Characters

<b>Character Set</b>	The set symbols that can be represented by a computer. The symbols are called characters and can be letters, digits, space, punctuation marks and some control character such as 'escape'. Each character is represented by a numerical code that is stored as a binary integer.	●
<b>ASCII</b>	American Standard Code for Information Interchange: a 7-bit character set used by PCs. (There is also an extended ASCII character set that uses 8 bits).	●
<b>Unicode</b>	A 16-bit character set that allows many more characters to be coded.	●

# Images

<b>Bitmap Image</b>	An image that has been stored as a series of values per pixel. The colour of each individual pixel is stored in a file.	●
<b>Vector Graphic</b>	An image file that is made up of lines and shapes that have certain properties, for example, a line may have the following properties: start-point, end-point, line colour, line thickness, line style. The properties of each shape are stored in a file to make up the image.	●
<b>Pixel</b>	Short for picture element. It is the smallest component of a bit-mapped image.	●
<b>Colour Depth</b>	The number of bits used to represent the colour of a single pixel in a bitmapped image. Higher colour depth gives a broader range of distinct colours. For example, an image stored as .gif file uses 8 bit per pixel so that the image could use 256 different colours.	●
<b>Resolution</b>	The number of pixels in an image expressed as: the-number-of-pixels-across X the-number-of-pixels-down e.g. 400 X 600	●
<b>Metadata</b>	Data about data. In the case of image files metadata is the data the computer needs to interpret the image data in the file e.g. resolution, colour depth and image dimensions.	●

# Sound

<b>Analogue</b>	A continuously changing wave such as natural sound.	●
<b>Digital</b>	Data that is made up of separate values. How data is stored on a computer.	●
<b>Sample Rate</b>	The number of times per second that the sound wave is measured. The higher the rate the more accurately the sound wave is represented.	●
<b>Sample Interval</b>	The time gap between measurements of the sound wave being taken. Another way of expressing the sampling rate.	●
<b>Sample Resolution</b>	The number of bits used to store the value of each sample. The higher the number of bits the more accurately the value is stored.	●
<b>ADC</b>	Analogue to Digital Converter: takes a real-world analogue data and converts it to a binary representation that can be stored on a computer.	●

## Data and Information

<b>Data</b>	Facts and figures with no context or format to give them meaning.	●
<b>Information</b>	Processed data that has context and format so that it conveys meaning.	●

## Instructions

<b>Instruction Set</b>	The group of instructions available for that specific processor to use. The number of instructions available will depend on the number of bits used. For example, with 4 bits there could potentially be 15 different instructions.	●
<b>Op Code</b>	The group of bits in an instruction that represents the operation such as EAT, MOVE or TURN.	●
<b>Compiler</b>	A piece of system software that converts a program written in a high level programming language into machine code (binary).	●
<b>Machine Code</b>	A binary representation of a program.	●
<b>High Level Programming Language</b>	A programming language written in constructs using language we can understand. Languages include Delphi, Visual Basic, Java and C++.	●

# The Central Processing Unit

<b>CPU</b>	The central processing unit that contains the processor, main memory and cache.	●
<b>Main memory/RAM</b>	Also known as Immediate Access Store and Primary Memory. The memory in the CPU that is used to temporarily store programs while they are running and data used by these programs. The processor fetches instructions from the main memory. Memory is made up of many addressable locations.	●
<b>Processor</b>	The component in the computer that fetches, decodes and executes instructions.	●
<b>Cache</b>	High speed memory in the CPU that is used to store a copy of frequently used instructions and data. It has a faster access speed than main memory and is used to improve CPU performance.	●
<b>Clock Speed</b>	Measured in hertz or cycles per second, the clock speed represents how many instructions per second the processor can execute. The higher the clock speed the faster the CPU can operate.	●
<b>System Buses</b>	The circuits/internal wiring that connect the processor and main memory together.	●
<b>Fetch-Execute Cycle</b>	The process by which a program is run: instructions are stored in main memory, fetched by the processor one at a time, decoded and executed.	●
<b>Dual-Core/Quad-Core</b>	A CPU that contains multiple processor components (cores) that can operate independently to process more than one task at a time.	●

# Memory

<b>RAM</b>	Random Access Memory: a type of memory that is read-write and volatile. Used for main memory.	●
<b>ROM</b>	Read Only Memory: memory that is hard-coded at the time of manufacture. Stores the start-up programs and the boot-strap loader.	●
<b>Bootstrap Loader</b>	The first program that is loaded into main memory from ROM when a computer is switched on. This will load the operating system from secondary storage.	●
<b>Volatile</b>	Describes memory that loses its contents when the power is turned off e.g. main memory.	●
<b>Non-Volatile</b>	Describes memory that does not lose its contents when the power is turned off e.g. har.d-disk	●
<b>Secondary Memory</b>	Long term, non-volatile storage media such as hard disks, memory sticks, magnetic tapes and CDs.	●
<b>Virtual Memory</b>	Part of the hard disk that is configured to behave as an extension to main memory.	●
<b>Magnetic Media</b>	Secondry storage such as hard disks, tape and floppy disks.	●
<b>Optical Media</b>	Secondary storage that is read using lasers such as CDs and DVDs.	●
<b>Solid State/Flash Memory</b>	Secondary storage that has no moving parts. Used in memory sticks, cameras and phones.	●
<b>Pen Drive</b>	Another term for a USB memory stick.	●



# Binary Logic

<b>Boolean Expression</b>	An expression that is either true or false e.g. $x=10$	●
<b>Truth Table</b>	A table that shows all the possible combinations of input and their logical output value.	●
<b>Logical Operators</b>	AND, OR, NOT	●
<b>Logic Diagram</b>	A diagram of a circuit showing logic gates with inputs and the outputs they generate.	●

# Software Categories

<b>Software</b>	Programs that run on the computer.	●
<b>System Software</b>	Programs that are used to run the computer, including the operating system, library routines and programming language translators.	●
<b>Application Software</b>	Programs that perform a task for a user. This is something that the user would do with or without a computer – it is the task that matters not the computer.	●
<b>Operating System</b>	Systems software that is necessary to run the computer.	●
<b>Utility Programs</b>	System software that provides other useful functions for operating the computer.	●
<b>Library</b>	Systems software modules that perform frequently required tasks. They can be built into or called from other programs.	●
<b>Translators</b>	System software that translates high level programming languages into machine code. Include compilers, interpreters and assemblers.	●

# Operating Systems

<b>Memory Management</b>	One of the main functions of the operating system: managing multiple applications using the space in main memory so all programs can be executed efficiently.	●
<b>Peripheral</b>	A device (hardware) that is connected to the CPU to provide input, output or storage.	●
<b>Peripheral Management</b>	One of the main functions of the operating system: managing the input to the CPU and output from the CPU.	●
<b>Multi-tasking</b>	One of the main functions of the operating system: managing how several tasks or programs, which are all running at the same time, share the processor.	●
<b>Security</b>	One of the main functions of the operating system: protecting the computer system from various hazards such as unauthorised users, viruses, hackers and accidental damage.	●
<b>User Interface</b>	The method of communication between the computer and its user. Sometimes called HCI (Human-Computer Interface) or MMI (Man-Machine Interface).	●
<b>HCI</b>	Human-Computer Interface: another term for user interface.	●

<b>MM</b>	Man-Machine Interface: another term for user interface.	●
<b>GUI</b>	Graphical User Interface: a style of user interface which is based on icons rather than text.	●
<b>WIMP Interface</b>	Windows, Icons, Menus and Pointers: describes a type of user interface where the user selects icons and menu items with a pointer of some kind (a mouse, stylus or finger).	●
<b>Command-line Interface</b>	A style of user interface that is only text based. Commands are typed in at a text prompt.	●
<b>Address</b>	A numerical reference to a location in memory.	●
<b>Process</b>	A program that is running in main memory.	●
<b>Single-user</b>	Describes the operating system of a computer where only one person can use the computer at any one time.	●
<b>Multi-user</b>	Describes the operating system of a computer where more than one person can use the computer at any one time.	●

# Utilities

<b>Virus</b>	A program that is installed on a computer without your knowledge or permission with the purpose of doing harm. It includes instructions to replicate automatically on a computer and between computers.	●
<b>Antivirus Software</b>	A utility program that prevents harmful programs being installed and important files being changed. If a virus does install itself, the antivirus software detects and removes it.	●
<b>Firewall</b>	A utility program that prevents unauthorised access to computer or a LAN from the internet and controls what sites computers on the LAN can access.	●
<b>Spyware</b>	A program that secretly records the user's actions on the computer including passwords and personal details they type when accessing a secure site.	●
<b>Disk Defragmenter</b>	A utility program that optimises the use of the hard disk space by collecting together the separate parts of each file in one location on the disk as well as grouping together the space so newly saved files do not have to be fragmented (split up).	●
<b>System Cleanup</b>	A utility program that deletes unnecessary files and settings to optimise the computer's performance.	●

<b>Automatic Update</b>	A utility program that, for any software already installed on the computer, will regularly check on the internet for newer versions and updates, download and install them.	●
<b>System Information and Diagnosis</b>	A utility program that presents information about the computer hardware and usage as well as information to help diagnose problems.	●
<b>Formatting</b>	A utility program that formats secondary storage devices such as hard disks.	●
<b>File Transfer and File Management</b>	A utility program that allows the user to create a logical view of how their files are organised using folders. Allows the user to move files within folders, copy, paste, name and delete files.	●

# Applications

<b>Open Source</b>	Software that is provided under licence but free of charge to anyone. The source code is made available and can be modified.	●
<b>License</b>	An agreement that defines the conditions for using the software.	●
<b>Freeware</b>	Software that is provided free of charge under license but without the source code. It is copyrighted and cannot be modified.	●
<b>Proprietary Software</b>	Software that is copyrighted and the license is old under a patented or trademarked name.	●
<b>Off-the-Shelf Software</b>	Software that can be purchased from a high street or online store as it is. Not custom-written.	●
<b>Custom Written Software</b>	Software that is written for a customer's specific requirements, just for that customer.	●

# Database Concepts

<b>Database</b>	A persistent organised store of data.	●
<b>Persistent Storage</b>	Non-volatile storage on a secondary storage medium such as a hard disk.	●
<b>Data Duplication/Data Redundancy</b>	Where the same data is stored more than once unnecessarily.	●
<b>Dara Inconsistency</b>	Where different versions of data become different because duplicate versions have been stored and updated differently.	●
<b>Program-Data Independence</b>	Where the applications that use a shared database are separated from the actual data by a database management system. Changes can be made to one application without it affecting another.	●
<b>DBMS</b>	Database Management System: the system that separates the applications from the data and provides features that allow database systems to be created, interrogated and maintained.	●
<b>Views</b>	A feature of a DBMS that provides each application or user with specific access rights and views of the database.	●



# Networking

<b>LAN</b>	A collection of computers and peripheral devices connected together within a single site.	●
<b>WAN</b>	A collection of computers and LANs connected together over a geographically remote area, using leased infrastructure.	●
<b>Topology</b>	A description of how devices are connected together. Does not necessarily represent physical layout.	●
<b>Bus (Topology)</b>	A topology where each device is connected to a main cable, referred to as the bus. Any device can transmit at any one time but only one transmission can occur on the main bus at any one time.	●
<b>Ring (Topology)</b>	A topology where each device is connected to the next in a loop. Uses a token-passing protocol to manage transmission by one device at one time.	●
<b>Star (Topology)</b>	A topology where each device has its own cable connecting it to a central device, which can be a switch or server.	●
<b>Peer-to-Peer</b>	A method of organising devices in a network where devices are all of equal status rather than having specialised roles. Each computer can access resources on another computer, assuming access rights have been granted by the other computer.	●

<b>Client-Server</b>	A method of organising devices in a network where some computers have specialised roles: servers. The servers provide resources and services to the other computers, known as clients. Management of the network and shared resources/files is centralised at the server.	●
<b>Hub</b>	A hardware device that provides connectivity onto a LAN cable. A multiport box that has a connection to the LAN from one side and several computers on the other. Can be wireless or cabled.	●
<b>Switch</b>	A hardware device that is similar to a hub but it has a built-in intelligence. Computers connected to a switch form a star topology LAN.	●
<b>Wireless Access Point</b>	The device to which a computer connects wirelessly. Can be a wireless hub or a wireless switch.	●
<b>NIC</b>	Network Interface Card: the card that goes into a computer to provide a connection to a LAN. Can be wireless or cabled.	●
<b>MAC Address</b>	A unique hardware number allocated to every NIC. It is a 48-bit address, usually written in hex, e.g. 00-09-7C-F1-F7-85	●
<b>Message</b>	A communication between devices. Split into packets for sending over a network and put back together again at the other end.	●

<b>Packet</b>	A fixed size chunk of a message created to send a message over a network. It has its own header containing data such as the destination address and packet number (so the message can be put back together in the right order).	●
<b>Protocol</b>	A set of rules that defines how devices communicate e.g. IP, HTTP, HTTPS.	●

# Internet

<b>Internet</b>	A worldwide network where computers and networks in geographically separate locations are connected together using a variety of communication links. Devices communicate using Internet Protocol (IP).	●
<b>Routers</b>	The hardware devices that make up the backbone of the internet as well as (small ones) providing connectivity from a LAN to the internet. Use Internet Protocol to communicate with each other.	●
<b>Modem</b>	The hardware device used to convert the digital transmission from a computer into an analogue signal that can be carried over the analogue telephone network. A method of accessing the internet.	●
<b>Digital</b>	A transmission signal that is made up of separate values (numbers) as opposed to the continuously changing signal in analogue transmissions.	●
<b>Analogue</b>	A transmission signal that is continuously changing as opposed to being made up of separate values (numbers). Sound in the real world is analogue.	●
<b>Broadband</b>	A method connecting to the internet using the site's normal phone line to carry digital transmissions. Allows more than one device (phone and computer) to share the link.	●

<b>WWW</b>	World-Wide Web: a collection of pages distributed on servers connected to the internet. Uses HTTP to request and send pages to browsers.	●
<b>HTTP</b>	HyperText Transfer Protocol: the protocol used by a browser to send page requests to a server and also by the server to send back the required page.	●
<b>HTTPS</b>	A secure version of HTTP where transmissions are encrypted.	●
<b>IP Addressing</b>	A method of labelling any device connected to the network with a unique numerical value. Uses four bytes usually expressed in this notation: 122.123.003.243	●
<b>Domain Name</b>	The text label for a website in the internet e.g. www.bbc.co.uk. It corresponds to an IP address for that site.	●
<b>DNS Servers</b>	Domain Name System Server: a database of domain names and associated IP addresses stored as servers. There are many DNS servers distributed across the internet, which communicate with each other.	●
<b>HTML</b>	HyperText Markup Language: the programming language used to define the layout and content of a webpage. Uses tags in conjunction with a CSS to control how content is displayed.	●
<b>CSS</b>	Cascading Style Sheet: defines the formatting and layout of the content defined by the HTML code e.g. <H1> may be 32pt Arial in Green	●
<b>Tags</b>	Labels that go around the content (text, pictures etc.) to define what they will look like on the page e.g. <H1> A Heading </H1>	●

# Security

<b>User Access Levels</b>	A network policy that defines which users can see which folders and files and the type of access they have to them e.g. Read-Only or Read-Write	●
<b>Encryption</b>	Where the data is changed, using a key, before it is transmitted so that it can only be deciphered by another device with the appropriate key. To anyone intercepting the message it would be unintelligible.	●
<b>Acceptable Use Policy</b>	An agreement that computer users will sign/agree to before being allowed access to a computer or the network.	●
<b>Failover</b>	When a hardware component fails, the computer switches over to a redundant component without the service to the user being interrupted.	●
<b>Redundant</b>	Spare, ready to be used if another component fails. Relates to spare hardware components in fault-tolerant systems that use failover.	●
<b>Fault-Tolerant</b>	A system that has been designed to cope with hardware failures. Uses redundant hardware and failover usually.	●
<b>Backup</b>	A copy of data is taken from a live computer system as a precaution against system failure or corruption/deletion of individual files/folders. To be restored in the event of data loss.	●

**Archiving**

Files are removed from the main computer system but kept in long-term storage, just in case they are needed in the future or because the law requires they be kept. Creates space on the main system

**Disaster Recovery**

A collection of precautions that ensures the computer system can be re-established very quickly after a catastrophe. Includes backup policy, complete hardware system available offsite at short notice and policies to restore data and applications on the replacement hardware.



# Compression

<b>Compression</b>	Making files smaller for quicker transmission over a network.	●
<b>Lossless Compression</b>	File is compressed with no loss of essential data.	●
<b>Lossy Compression</b>	Files are compressed by removing some data that is less essential for the purpose. For example, using fewer colours in a picture (reduce colour depth).	●



# Algorithms

<b>Algorithm</b>	A series of steps to solve a problem. Can be expressed in structured English, pseudo code or as a system flowchart.	●
<b>System Flowchart</b>	A diagram using commonly defined symbols to express an algorithm.	●
<b>Structured English</b>	A way of writing an algorithm in natural language using some basic programming constructs such as IF... THEN...ELSE and loops. More structured than just natural language/prose.	●
<b>Pseudo Code</b>	A way of writing an algorithm that is close to actual programming language, using coding-style constructs such as IF, THEN ELSE, loops and array notation as appropriate.	●
<b>Hungarian Notation</b>	The convention of prefixing identifiers to indicate what type of object they are. Commonly used with forms where, for example, the prefix txt might indicate a textbox and lst might prefix a list box. The prefix is conventionally in lowercase.	●
<b>camelCase</b>	The use of capital letters in an identifier to make them more readable. With camelCase the first word is not capitalised:	●

<b>Sequence</b>	Where instructions are executed one after another in series.	●
<b>Selection</b>	Where the program will execute certain instructions based on conditions. Selection statements include IF... THEN...ELSE and CASE... OF to select which commands to execute.	●
<b>Iteration</b>	Where a program will execute a group of instructions zero or more times based on a condition. FOR loops will execute instructions a specific number of times, REPEAT...UNTIL loops for one or more times and WHILE...DO loops for zero or more times.	●
<b>Condition</b>	A Boolean expression that controls an iteration or selection statement e.g. REPEAT...UNTIL X=10 (X=10 is the condition).	●
<b>Boolean Expression</b>	An expression that is true or false e.g. continue="Y"	●

# Languages

<b>High Level Programming Language</b>	A programming language where programming constructs are written in a way that is close to natural language instead of in mnemonics or machine code such as Delphi, Pascal, Visual Basic, Java C++ etc.	●
<b>Assembly Language</b>	2nd generation programming language where instructions are in the form of mnemonics.	●
<b>Mnemonics</b>	A way of writing programming instructions using abbreviations of commands and the data to be used e.g. LDA #34, ADD &5F3A	●
<b>Machine Code</b>	1st generation code, binary instructions where some bits are used to define the operation (opcode) and some bites define the data to be used.	●
<b>Translator</b>	The piece of systems software used to convert different programming languages into machine code. Three types: assembler, interpreter and compiler.	●
<b>Interpreter</b>	A translator that converts high level languages into machine code. Works one line at a time checking syntax, converting to machine code and executing the code.	●
<b>Compiler</b>	A translator that converts high level languages into machine code. Works through the whole program (source code) checking the syntax, then converting to machine code and creating the executable object code. The object code is executed, not the source code.	●
<b>Source Code</b>	The original high level program,	●
<b>Object Code</b>	The executable version of the program after it has been compiled.	●
<b>Assembler</b>	The translator that converts assembly language programs into machine code.	●

# Data handling in Algorithms

<b>Identifier</b>	A unique name for something (variable, constant, program, procedure etc) within the program.	●
<b>Constant</b>	A named value within a program that is assigned a specific value. Its value does not change while the program is running.	●
<b>Variable</b>	An identifier associated with a particular memory location, used to store data. Its value may change as the program is run and new values are assigned to it.	●
<b>Data Type</b>	A formal description of the type of data being stored in a variable. It defines the amount of memory required and the type of operations that can be performed on that variable.	●
<b>Integer</b>	Data type for whole numbers, typically 2 bytes.	●
<b>Real</b>	Data type for fractional numbers, typically 4 bytes.	●
<b>Char</b>	Data type for a single character, 1 byte.	●
<b>String</b>	Data type for text, more than one character. Usually up to 255 characters and takes 1 byte per character.	●
<b>Boolean</b>	True or false, typically 1 byte.	●
<b>Operations</b>	The actions that can be performed on a variable.	●
<b>Arithmetic Operations</b>	Add, subtract, multiply, divide, integer division (DIV) and modulus (MOD)	●
<b>Comparison Operations</b>	= < > <= >= <>	●
<b>Logical Operators</b>	NOT AND OR	●
<b>Boolean Expressions</b>	Expressions that resolve to true or false e.g. $x \leq 20$	●
<b>Array</b>	A group of data items of the same data type that use a single identifier. Individual data items are accessed using a subscript.	●

# Errors and Testing

<b>Syntax</b>	A set of rules that defines how program statements must be written in order for the translator to understand them.	●
<b>Syntax Errors</b>	An error in the format of the program statements such as missing semicolons or keywords spelt incorrectly.	●
<b>Logic Errors</b>	An error in the algorithm that means the outcome is not as expected, even though the program will run.	●
<b>Valid Data</b>	Data that should be allowed by the program. For example, if a range of 1-10 is allowed, then valid data will be any number between 1 and 10 (including 1 and 10).	●
<b>Invalid Data</b>	Data that is not valid and should be rejected by the program. For example, if a range of 1-10 is allowed, then invalid data will include -251, 0, 11 and 345.	●
<b>Boundary Data</b>	Data either side of the range extremes. For example, in a range of 1-10 boundary data will include 0, 1, 10 and 11.	●