GCSE Computer Science- Key Terminology

No.	Spec	Section	Term	Definition
1.	1.1	Systems architecture	CPU	**Central Processing Unit:** "The main part of the computer, consisting of the registers, ALU and control unit."
2.	1.1	Systems architecture	Von Neumann Architecture	"Traditional computer architecture that forms the basis of most digital computer systems. Instructions are fetched, decoded and executed one at a time."
3.	1.1	Systems architecture	MAR	**Memory Address Register:** "Holds the address of data ready for use by the memory data register. or the address of an instruction passed from the program counter. Step 2 of the fetch, decode, execute cycle."
4.	1.1	Systems architecture	MDR	**Memory Data Register:** "Holds the data fetched from or to be written to the memory. Step 3 of the fetch, decode, execute cycle."
5.	1.1	Systems architecture	PC	**Program Counter:** "Holds the address of the next instruction. Step 1 of the fetch, decode, execute cycle."
6.	1.1	Systems architecture	Accumulator	"Holds the result of calculations."
7.	1.1	Systems architecture	ALU	**Arithmetic Logic Unit:** "Performs calculations e.g. x = 2 + 3 and logical comparisons e.g. IF x > 3 in the CPU."
8.	1.1	Systems architecture	CU	**Control Unit:** "Decodes instructions. Sends signals to control how data moves around the CPU."
9.	1.1	Systems architecture	Cache	"Memory in the processor providing fast access to frequently used instructions and data."
10.	1.1	Systems architecture	F-D-E cycle	**Fetch-Decode-Execute Cycle:** "The complete process of retrieving an instruction from store, decoding it and carrying it out. Also known as the instruction cycle."
11.	1.1	Systems architecture	Clock Speed	"Measured in Hertz, the clock speed is the frequency at which the internal clock generates pulses. The higher the clock rate, the faster the computer may work. The "clock" is the electronic unit that synchronises related components by generating pulses at a constant rate."
12.	1.1	Systems architecture	Cache Size	"A part of the main store between the central processor and the rest of the memory. It has extremely fast access, so sections of a program and its associated data are copied there to take advantage of its short fetch cycle. The larger the size of the cache the more that can be copied and stored here without having to go back to slower main memory (RAM), this has a significant impact on the speed of processing."
13.	1.1	Systems architecture	Cores	"A part of a multi-core processor. A multi-core processor is a single component with two or more independent actual CPUs, which are the units responsibly for the fetch-decode-execute cycle."
14.	1.1	Systems architecture	Embedded System	"A computer which has been built to solve a very specific program and is not easily changed. For example the operating system placed inside a washing machine, microwave or set of traffic lights."
15.	1.2	Memory	RAM	**Random Access Memory:** "Volatile (data lost when power is off) Read and write. Purpose: temporary store of currently executing instructions and their data. E.g. applications and the operating system in use."

GCSE Computer Science- Key Terminology

16.	**1.2**	Memory	ROM	**Read Only Memory:** "Non-volatile (data retained when power is off) Read only. Purpose: stores instructions for starting the computer called the bootstrap."
17.	**1.2**	Memory	Virtual Memory	"Using part of the hard disk as if it were random access memory. Allows more applications to be open than physical memory could hold."
18.	**1.2**	Memory	Flash Memory	"Solid state (no moving parts). Faster than a hard disk drive. Robust. Used to store the BIOS."
19.	**1.3**	Storage	Secondary Storage	"Permanent storage of instructions and data not in use by the processor. Stores the operating system, applications and data not in use. Read/write and non-volatile."
20.	**1.3**	Storage	Optical Storage	"CD/R, CD/RW, DVD/R, DVD/RW Use: music, films and archive files. Low capacity. Slow access speed. High portability. Prone to scratches. Low cost."
21.	**1.3**	Storage	Magnetic Storage	"Hard disk drive. Use: operating system and applications. High capacity. Medium data access speed. Low portability (except for portable drives). Reliable but not durable. Medium cost."
22.	**1.3**	Storage	Solid State Storage	"Memory cards & solid state hard drive (SSD). Use: digital cameras and smartphones. Medium capacity. High portability. Reliable and durable. No moving parts. Fast data access speed. High cost."
23.	**1.3**	Storage	Storage Capacity	"The amount of data a storage device is able to store. "
24.	**1.3**	Storage	Storage Speed	"The read/write access speed of a storage device."
25.	**1.3**	Storage	Storage Portability	"How easy it is to transport a given storage medium. E.g. Solid state and optical storage and designed to be highly portable, whereas more traditional magnetic storage is designed to stay in place."
26.	**1.3**	Storage	Storage Durability	"How resistant to damage and wear a tear a storage device is. Devices with low durability will wear out easily over time."
27.	**1.3**	Storage	Storage Cost	"The relative price of a storage medium on a"
28.	**1.4**	Wired and wireless networks	LAN	**Local Area Network:** "Small geographic area. All the hardware for the LAN is owned by the organisation using it. Wired with UTP cable, fibre optic cable or wireless using routers and Wi-Fi access points."
29.	**1.4**	Wired and wireless networks	WAN	**Wide Area Network:** "Large geographic area. Infrastructure is hired from telecommunication companies who own and manage it. Connected with telephone lines, fibre optic cables or satellite links."
30.	**1.4**	Wired and wireless networks	Client-Server Network	"A client makes requests to the server for data and connections. A server controls access and security to one shared file store. A server manages access to the internet, shared printers and email services. A server runs a backup of data."
31.	**1.4**	Wired and wireless networks	Peer-to-Peer Network	"All computers are equal. Computers serve their own files to each other. Each computer is responsible for its own security and backup. Computers usually have their own printer."
32.	**1.4**	Wired and wireless networks	Stand-Alone Computer	"A single computing device not connected to any other on a network, either wired or wireless."
33.	**1.4**	Wired and wireless networks	WAP	**Wireless Access Point:** "A networking hardware device that allows a Wi-Fi device to connect to a wired network."

GCSE Computer Science- Key Terminology

34.	**1.4**	Wired and wireless networks	Router / Switch	"In packet-switched networks such as the internet, a router is a device or, in some cases, software on a computer, that determines the best way for a packet to be forwarded to its destination."
35.	**1.4**	Wired and wireless networks	NIC	**Network Interface Controller:** "A computer hardware component that connects a computer to a computer network."
36.	**1.4**	Wired and wireless networks	Transmission Media	"The physical media over which data is transmitted, e.g. twisted copper cable, fibre optic etc. "
37.	**1.4**	Wired and wireless networks	DNS	**Domain Name System:**
38.	**1.4**	Wired and wireless networks	Hosting	"Websites stored on dedicated servers. Reasons include: Websites need to be available 24/7. Accessed by thousands of users at a time. Strong protection from hackers. They need an IP address that doesn't change."
39.	**1.4**	Wired and wireless networks	The Cloud	"Remote servers that store data that can be accessed over the internet. Advantages: Access anytime, anywhere from any device. Automatic backup. Collaborate on files easily."
40.	**1.4**	Wired and wireless networks	Virtual Networks	"A logical software based network. Advantages: + Increased security, including more secure access to the network remotely. + The network can easily be expanded with less impact on the infrastructure and cost. + Log in to work from home."
41.	**1.5**	Network topologies, protocols & layers	Star Network Topology	"Computers connected to a central switch. If one computer fails no others are affected. If the switch fails all connections are affected."
42.	**1.5**	Network topologies, protocols & layers	Mesh Network Topology	"Switches (LAN) or routers (WAN) connected so there is more than one route to the destination. e.g. The Internet More resilient to faults but more cable needed."
43.	**1.5**	Network topologies, protocols & layers	WiFi	"Wireless connection to a network. Requires a wireless access point or router. Data is sent on a specific frequency. Each frequency is called a channel."
44.	**1.5**	Network topologies, protocols & layers	Frequency	"The number of occurrences of a repeating event per unit of time. Data is set on a specific frequency over a WiFi connection."
45.	**1.5**	Network topologies, protocols & layers	Channels	"Each given signal frequency on a WiFi connection is known as a channel."
46.	**1.5**	Network topologies, protocols & layers	Encryption	"Encoding readable data called plaintext into unreadable data called ciphertext. Only the intended recipient can decode the data using a key. Protects communications from hackers."
47.	**1.5**	Network topologies, protocols & layers	Ethernet	"A standard for networking local area networks using protocols. Frames are used to transmit data. A frame contains the source and destination address, the data and error checking bits. Uses twisted pair and fibre optic cables. A switch connects computers together."
48.	**1.5**	Network topologies, protocols & layers	IP Address	**Internet Protocol Address:** "A unique string of numbers separated by full stops that identifies each computer using the Internet Protocol to communicate over a network."
49.	**1.5**	Network topologies, protocols & layers	MAC Address	**Media Access Control Address:** "A unique identifier assigned to network interfaces for communications at the data link layer of a network segment. MAC addresses are used as a network address for most network technologies, including Ethernet and Wi-Fi."
50.	**1.5**	Network topologies, protocols & layers	Protocol	"A set of rules that allow two devices to communicate."

GCSE Computer Science- Key Terminology

51.	**1.5**	Network topologies, protocols & layers	TCP/IP	**Transmission Control Protocol / Internet Protocol:** "TCP provides an error free transmission between two routers. IP routes packets across a wide area network."
52.	**1.5**	Network topologies, protocols & layers	HTTP	**Hypertext Transfer Protocol:** "A client-server method of requesting and delivering HTML web pages. Used when the information on a web page is not sensitive or personal."
53.	**1.5**	Network topologies, protocols & layers	HTTPS	**Hypertext Transfer Protocol Secure:** "Encryption and authentication for requesting and delivering HTML web pages. Used when sensitive form or database data needs to be transferred. e.g. passwords and bank account details."
54.	**1.5**	Network topologies, protocols & layers	FTP	**File Transfer Protocol:** "Used for sending files between computers, usually on a wide area network. Typically used for uploading web pages and associated files to a web server for hosting."
55.	**1.5**	Network topologies, protocols & layers	POP	**Post Office Protocol:** "Used by email clients to retrieve email from an email server."
56.	**1.5**	Network topologies, protocols & layers	IMAP	**Internet Message Access Protocol:** "Used by mail clients to manage remote mail boxes and retrieve email from a mail server."
57.	**1.5**	Network topologies, protocols & layers	SMTP	**Simple Mail Transfer Protocol:** "Sends email to an email server."
58.	**1.5**	Network topologies, protocols & layers	Packet Switching	"TCP splits data into smaller packets. Each packet takes its own route. Packets are assembled back into the correct order when they arrive at the destination. Maximises the use of the network. More secure as the full data stream is not sent in the same direction."
59.	**1.6**	System security	Malware	"Software written to cause loss of data, encryption of data, fraud and identity theft: virus, worm, trojan, ransomware and spyware."
60.	**1.6**	System security	Phishing	"Sending emails purporting to be from reputable companies to induce people to reveal personal information."
61.	**1.6**	System security	Social Engineering	"Most vulnerabilities are caused by humans. Not locking computers. Using insecure passwords. Not following/poor company network policies. Not installing protection software. Not being vigilant with email/files received. Not encrypting sensitive data."
62.	**1.6**	System security	Brute Force Attacks	"A trial and error method of attempting passwords. Automated software is used to generate a large number of guesses."
63.	**1.6**	System security	Denial of Service Attacks	"Flooding a server with so much traffic it is unable to process legitimate requests."
64.	**1.6**	System security	Data Interception	"Stealing computer-based information."
65.	**1.6**	System security	SQL Injection	"A hacking technique used to view or change data in a database by inserting SQL code instead of data into a text box on a form."
66.	**1.6**	System security	Network Policies	"Rules put in place on a Local Area Network by a systems administrator. They control aspects such as what certain types of users are allowed to / what they are allowed to access etc."
67.	**1.6**	System security	Penetration Testing	"Testing designed to check the security and vulnerabilities of a system."

GCSE Computer Science- Key Terminology

68.	**1.6**	System security	Network Forensics	"Network forensics is a sub-branch of digital forensics relating to the monitoring and analysis of computer network traffic for the purposes of information gathering, legal evidence, or intrusion detection."
69.	**1.6**	System security	Anti-Malware Software	"Antimalware software protects against infections caused by many types of malware, including viruses, worms, Trojan horses, rootkits, spyware, key loggers, ransomware and adware."
70.	**1.6**	System security	Firewalls	"A computer application used in a network to prevent external users gaining unauthorised access to a computer system."
71.	**1.6**	System security	User Access Level	"The amount of access a given user is allowed to a computer. On a network most users will have restricted access. Whereas a systems administer or network technician would be allowed much greater access with fewer restrictions."
72.	**1.6**	System security	Password	"A secret word or phrase that must be used to gain access to a computer / program / interface / system."
73.	**1.7**	Systems software	Systems Software	"Collection of systems software that manages the computer. Usually supplied with the computer. Most common operating systems are Windows, Linux, Unix, MacOS, iOS."
74.	**1.7**	Systems software	User Interface	"The means by which the user and a computer system interact, in particular the use of input devices and software."
75.	**1.7**	Systems software	Memory Management	"The process of the operating system deciding what should be in memory at any given time. Responsible for loading data and programs into and out of memory when required."
76.	**1.7**	Systems software	Multitasking	"Running more than one application at a time by giving each one a slice of processor time."
77.	**1.7**	Systems software	Peripheral Management	"The process of your operating system dealing with requests / input / output to and from any connected peripheral devices such as a mouse, keyboard, webcam, speaker, scanner, printer etc."
78.	**1.7**	Systems software	Device Drivers	"Translates commands from the operating system into hardware specific commands that a device understands. e.g. A printer driver tells the printer how to print a document from the operating system."
79.	**1.7**	Systems software	User Management	"Operating system provides for: Allowing different people to log into the same computer with a username and password. Remembering personal settings. Managing access rights to files."
80.	**1.7**	Systems software	File Management	"Operating system provides: Access permissions for files (read and write). Opening files in associated programs. Moving, deleting and renaming files. Presenting a folder structure to the user."
81.	**1.7**	Systems software	Utility System Software	"A systems program that performs some specific task in the operation of the computer, for example file backup, virus checking or a compression program."
82.	**1.7**	Systems software	Encryption Software	"Turns plaintext data into unreadable ciphertext data using a key. Protects data from being read by hackers."
83.	**1.7**	Systems software	Defragmentation Software	"Different sized files saved on disk are deleted over time creating gaps on the disk. New files fill up the gaps, but may need more space than the gap provides resulting in fragments of the file being spread across the disk. Defragmentation rearranges parts of files back to contiguous space. Makes access quicker."
84.	**1.7**	Systems software	Data Compression	"Reduces the size of a file. Takes up less disk space. Quicker to download over the internet. Compressed files must be extracted before they can be read."

GCSE Computer Science- Key Terminology

85.	**1.7**	Systems software	Full Backup	"Every file is copied to an alternative storage device. E.g. portable hard drive. Files can be recovered if it is deleted or corrupted. Can be slow to copy large numbers of files."
86.	**1.7**	Systems software	Incremental Backup	"Only the files that have changed since the last backup are copied. Files can be recovered if it is deleted or corrupted. Much quicker than a full backup."
87.	**1.8**	Ethical, legal, cultural & environmental concerns	Ethical Issues	"The ethical and moral issues which have come about in modern society due to the increase use of computer science and its related technologies. e.g. Losing/changing jobs. Efficiency: robots work 24/7. Access to IT is not equal (digital divide). Invasion of privacy. Responsibility for content on the internet. "
88.	**1.8**	Ethical, legal, cultural & environmental concerns	Legal Issues	"The legal issues which have come about in modern society due to the increase use of computer science and its related technologies. e.g. Copyright and ownership of digital content, different laws in different countries (crime may be committed in a certain country, but the people committing the crime could be physically located in another), hacking, piracy. "
89.	**1.8**	Ethical, legal, cultural & environmental concerns	Cultural Issues	"The cultural moral issues which have come about in modern society due to the increase use of computer science and its related technologies. e.g. Censorship to prevent political unrest and preserve culture. Geography & economy of a country affects access to networks and power. Increased mobile technology impacts on how people communicate: cyberbullying. "
90.	**1.8**	Ethical, legal, cultural & environmental concerns	Environmental Issues	"The environmental issues which have come about in modern society due to the increase use of computer science and its related technologies. e.g. Manufacturing computers uses fossil fuels. Limited number of natural resources. Data centres use 2% of global energy. Computers contain hazardous materials, often shipped to other countries for disposal. "
91.	**1.8**	Ethical, legal, cultural & environmental concerns	Privacy Issues	"The privacy issues which have come about in modern society due to the increase use of computer science and its related technologies. e.g. Increase in always on, voice activated devices in the home. Rise in CCTV. Rise in social networking and GPS tracking."
92.	**1.8**	Ethical, legal, cultural & environmental concerns	Stakeholder	"Anyone with an interest in a business. e.g. Company owners: profits & reputation. Workers: jobs & salary. Customers: price & convenience. Suppliers: costs. Local community: environment & jobs."
93.	**1.8**	Ethical, legal, cultural & environmental concerns	Open Source Software	"Users can modify and distribute the software. Can be installed on any number of computers. Support provided by the community. Users have access to the source code. May not be fully tested."
94.	**1.8**	Ethical, legal, cultural & environmental concerns	Proprietary Software	"Users cannot modify the software. Copyright protected. Usually paid for. Licensed per user or per computer. Support provided by developers. Users do not have access to the source code. Fully tested and supported by developers."
95.	**1.8**	Ethical, legal, cultural & environmental concerns	Data Protection Act	"Legislation which protects individuals from unreasonable use of their store personal data."
96.	**1.8**	Ethical, legal, cultural & environmental concerns	Computer Misuse Act	"Legislation which defines electronic vandalism, unauthorised access to computer systems and theft of information."
97.	**1.8**	Ethical, legal, cultural & environmental concerns	Copyright Designs and Patents Act	"Legislation which gives creators of literacy, dramatic, musical and artistic works the right to control the ways in which their material may be used."
98.	**1.8**	Ethical, legal, cultural & environmental concerns	Creative Commons Licensing	"A way to grant copyright permissions to creative work. It allows an author to retain copyright while allowing others to copy, distribute, and make some uses of their work."

GCSE Computer Science- Key Terminology

99.	**1.8**	Ethical, legal, cultural & environmental concerns	Freedom of Information Act	"Members of the public can request information from public bodies about their activities. Public authorities are obliged to publish data about their activities when asked."
100.	**2.1**	Algorithms	Computational Thinking	"The thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out."
101.	**2.1**	Algorithms	Abstraction	"The process of separating ideas from specific instances of those ideas at work. Computational structures are defined by their meanings, while hiding away the details of how they work. Abstraction tries to factor out details from a common pattern so that programmers can work close to the level of human thoughts, leaving out details which matter in practice, but are immaterial to the problem being solved."
102.	**2.1**	Algorithms	Decomposition	"The process by which a complex problem or system is broken down into parts that are easier to conceive, understand, program and maintain."
103.	**2.1**	Algorithms	Algorithmic Thinking	"A way of getting to a solution by identifying the steps needed."
104.	**2.1**	Algorithms	Binary Search	"A particularly efficient search method. It only works if records in the file are in sequence. A binary search involvers accessing the middle record in the file and determining if the target record has been found or, if not, if it is before or after in the sequence. This process is repeated on the part of the file where the target record is expected, until it is found."
105.	**2.1**	Algorithms	Linear Search	"Involves examining each entry in turn in the file until the time is found or the end of the file is reached. Unless the file is in some useful order a serial search has to be used."
106.	**2.1**	Algorithms	Bubble Sort	"A simple algorithm popular with inexperienced programmers. It is inefficient when sorting large amounts of data as the time taken is related to the square of the number of items. If 10 items take 1ms then 100 times will take 100ms (this is 10 times the number of items and so the time will be 102 or 100 times longer)."
107.	**2.1**	Algorithms	Merge Sort	"A type of divide and conquer algorithm that was incited by John von Neumann. First the list is divided into the smallest unit (1 element), then each element is compared with the adjacent list to sort and merge the two adjacent lists. Finally all elements are sorted and merged."
108.	**2.1**	Algorithms	Insertion Sort	"A simple sorting algorithm that builds the final sorted array (or list) one item at time. It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort, or merge sort."
109.	**2.1**	Algorithms	Algorithm	"A sequence of steps designed to perform a particular task. An algorithm may be constructed to describe the operation of a complete system or to describe a particular part of it."
110.	**2.1**	Algorithms	Pseudocode	"A language independent description of the steps of an algorithm. Intended for humans to express and design algorithms before coding."
111.	**2.1**	Algorithms	Flow Diagram	"A method of designing algorithms before coding using symbols."
112.	**2.2**	Programming techniques	Variable	"A value that can change, depending on conditions or on information passed to the program."
113.	**2.2**	Programming techniques	Constant	"A value that cannot be altered by the program during normal execution, i.e., the value is constant."

GCSE Computer Science- Key Terminology

114.	**2.2**	Programming techniques	Inputs	"Any information or data which goes into a system."
115.	**2.2**	Programming techniques	Outputs	"Any information of data which leaves a system."
116.	**2.2**	Programming techniques	Assignments	"Giving a variable or constant a value. e.g. counter = 0"
117.	**2.2**	Programming techniques	Sequence	"One of the 3 basic programming constructs. Instructions happening one after the other in order is sequence."
118.	**2.2**	Programming techniques	Selection	"One of the 3 basic programming constructs. Instructions which can evaluate a Boolean expression and then branch the code to one or more alternatives paths is branching / selection."
119.	**2.2**	Programming techniques	Iteration	"One of the 3 basic programming constructs. A selection of code which can be repeated either a set number of times (count controlled) or a variable number of times based on the evaluation of a Boolean expression (condition controlled) is iteration."
120.	**2.2**	Programming techniques	Count Controlled Loop	"An iteration which loops a fixed number of times. The count is kept in a variable called an index or counter. When the index reaches a certain value (the loop bound) the loop will end. Count-controlled repetition is often called definite repetition because the number of repetitions is known before the loop begins executing."
121.	**2.2**	Programming techniques	Condition Controlled Loop	"A way for computer programs to repeat one or more various steps depending on conditions set either by the programmer initially or real-time by the actual program."
122.	**2.2**	Programming techniques	String Manipulation	"Commands and techniques which allow you to alter and extract information from textual strings e.g. LENGTH, LEFT, RIGHT, SUBSTRING, UPPER, LOWER, ASC, CHAR etc."
123.	**2.2**	Programming techniques	File Handling: Open	"File handling is the process of dealing with input to and from files. Files first have to be opened, this creates a handle to the file and allows reading and writing."
124.	**2.2**	Programming techniques	File Handling: Read	"File handling is the process of dealing with input to and from files. Once a file has been opened it is possible to use commands to read its contents and return them to your program."
125.	**2.2**	Programming techniques	File Handling: Write	"File handling is the process of dealing with input to and from files. Once a file has be opened it is possible to use commands to write data to file from your program."
126.	**2.2**	Programming techniques	File Handling: Close	"File handling is the process of dealing with input to and from files. Once you are done reading / writing it is important to close a file, this releases the file handle and breaks the connection between it and your program."
127.	**2.2**	Programming techniques	SQL	"The language and syntax used to write and run database queries"
128.	**2.2**	Programming techniques	Arrays / Lists	"A set of data items of the same type grouped together using a single identifier. Each of the data items is addressed by the variable name and a subscript."
129.	**2.2**	Programming techniques	Subroutine	"A block of code given a unique identifiable name within a program. Supports code reuse and good programming technique."

GCSE Computer Science- Key Terminology

130.	**2.2**	Programming techniques	Procedure	"A block of code given a unique identifiable name within a program. A procedure can take either zero or more parameters when it is called. The procedure should be designed and written to perform one task or action which is clearly indicated by its name."
131.	**2.2**	Programming techniques	Function	"A block of code given a unique identifiable name within a program. A function can take either zero or more parameters when it is called and should return a value. The function should be designed and written to perform one task or action which is clearly indicated by its name."
132.	**2.2**	Programming techniques	Data Type	"The basic data types provided by a programming language as building blocks. Most languages allow more complicated composite types to be recursively construction starting from basic types. E.g. char, integer, float, Boolean. As an extension a 'string' data type is constructed behind the scenes of many char data types."
133.	**2.2**	Programming techniques	Integer	"A data type used to store positive and negative whole numbers."
134.	**2.2**	Programming techniques	Real	"A data type used to store an approximation of a real number in a way that can support a trade-off between range and precision. A number is, in general, represented approximately to a fixed number of significant digits and scaled using an exponent."
135.	**2.2**	Programming techniques	Boolean	"Used to store the logical conditions TRUE / FALSE. Often translated to On/Off, Yes/No etc."
136.	**2.2**	Programming techniques	Character	"A single alphanumeric character or symbol."
137.	**2.2**	Programming techniques	String	"A sequence of alphanumeric characters and or symbols. e.g. a word or sentence."
138.	**2.2**	Programming techniques	Casting	"Converting a variable from one data type to another. e.g. variable entered as a string, but needs to be an integer for calculation. age = INPUT("Enter your age: ") age = INT(age)"
139.	**2.2**	Programming techniques	Arithmetic Operator	"+, -, /, *, ^. Used in mathematical expressions e.g. num1 + num2 = sum"
140.	**2.2**	Programming techniques	Boolean Operator	"AND, OR, NOT. Used in conditions. e.g. IF choice < "1" OR choice > "3""
141.	**2.3**	Producing robust programs	Defensive Design	"Defensive design is the practice of planning for contingencies in the design stage of a project or undertaking."
142.	**2.3**	Producing robust programs	Input Sanitisation / Validation	"Ensuring data input by the user meets specific criteria before processing. **Range check**. E.g. between 1 and 31. **Type check**. E.g. number not symbol. **Presence check**. E.g. data has been input. **Format check**. E.g. postcode is LLN(N) NLL. "
143.	**2.3**	Producing robust programs	Authentication	"Verifying a user identity before they can use a program with username and password. Strong passwords over a certain length with symbols and mixed case are advised."
144.	**2.3**	Producing robust programs	Maintainability	"A selection of techniques and methods that make code easy to debug, update and maintain."
145.	**2.3**	Producing robust programs	Comments	"Used by a programmer to explains sections of code. Ignored by the compiler."

GCSE Computer Science- Key Terminology

146.	**2.3**	Producing robust programs	Indentation	"Indenting makes it easy to see where structures begin and end. Conditions and iterations should be indented. Code inside procedures and functions should be indented."
147.	**2.3**	Producing robust programs	Testing	"This involves testing the program under various conditions to make sure it is going to work. You need to think about what devices it could be used on and what might cause the program to crash."
148.	**2.3**	Producing robust programs	Iterative Testing	"Each module of a program is tested as it is developed."
149.	**2.3**	Producing robust programs	Final / Terminal Testing	"Testing that all the modules of a program work together as expected. Checking the program meets the expectations of the user with real data."
150.	**2.3**	Producing robust programs	Syntax Errors	"Rules of the language have been broken. The program will not run. Variables not being declared before use. Incompatibility of variable types. E.g. sum = A Using assignments incorrectly. E.g. 2 + 2 = x Keywords misspelt. E.g. PRNT("Hello")"
151.	**2.3**	Producing robust programs	Logic Errors	"The program runs but does not give the expected output. Division by zero. Infinite loop. Memory full. File not found."
152.	**2.3**	Producing robust programs	Test Data	"Values used to test a program, includes normal test data, boundary test data and erroneous test data."
153.	**2.4**	Computational logic	Data Representation	"The way in which different types of data (images, sound, text) are stored in a digital format."
154.	**2.4**	Computational logic	Binary Data Representation	"The patterns of 1's and 0's used to represent all forms of data in a digital format."
155.	**2.4**	Computational logic	Logic Diagram	"A method of expression Boolean Logic in a diagrammatic form using a set of standard symbols representing the various Logic Gates such as AND NOT OR NAND etc."
156.	**2.4**	Computational logic	AND	"A logical operator used within a program. AND works by only returning TRUE if both values being compared are TRUE."
157.	**2.4**	Computational logic	OR	"A logical operator used within a program. OR works by returning TRUE as long as either value being compared is TRUE."
158.	**2.4**	Computational logic	NOT	"A logical operator used within a program. NOT works by returning FALSE if the input is TRUE, and returning TRUE if the input is FALSE."
159.	**2.4**	Computational logic	Truth Table	"A notation used in Boolean algebra for defining the output of a logic gate or logic circuit for all possible combinations of inputs."
160.	**2.4**	Computational logic	Computing-Related Mathematics: +	"One of the standard operators you can use in virtually all programming languages to carry out computing-related mathematics: + is the standard symbol used for addition."
161.	**2.4**	Computational logic	Computing-Related Mathematics: -	"One of the standard operators you can use in virtually all programming languages to carry out computing-related mathematics: - is the standard symbol used for subtraction."
162.	**2.4**	Computational logic	Computing-Related Mathematics: *	"One of the standard operators you can use in virtually all programming languages to carry out computing-related mathematics: * is the standard symbol used for multiplication."
163.	**2.4**	Computational logic	Computing-Related Mathematics: /	"One of the standard operators you can use in virtually all programming languages to carry out computing-related mathematics: / is the standard symbol used for real division."

GCSE Computer Science- Key Terminology

164.	**2.4**	Computational logic	Computing-Related Mathematics: ^	"One of the standard operators you can use in virtually all programming languages to carry out computing-related mathematics: ^ is the standard symbol used for exponent."
165.	**2.4**	Computational logic	Computing-Related Mathematics: MOD	"One of the standard operators you can use in virtually all programming to carry out integer division: MOD gives you remainder left over e.g. 10 MOD 3 would give you 1."
166.	**2.4**	Computational logic	Computing-Related Mathematics: DIV	"One of the standard operators you can use in virtually all programming to carry out integer division: DIV gives you the number of times a number fits into another number e.g. 10 MOD 3 would give you 3."
167.	**2.5**	Translators and facilities of languages	Low Level Language	"A language which is close to machine code. Related closely to the design of the machine. A one-to-one language."
168.	**2.5**	Translators and facilities of languages	High Level Language	"A language designed to help a programmer express a computer program in a way that reflects the problem that is being solved, rather than the details of how the computer will produce the solution. One-to-many language."
169.	**2.5**	Translators and facilities of languages	Translators	"A program that translates a program written in assembly language into machine code."
170.	**2.5**	Translators and facilities of languages	Assembler	"A program that translates (assembles) a program written in assembly language into machine code."
171.	**2.5**	Translators and facilities of languages	Compiler	"A program that translates a high-level language program, source code, into a computer's machine code."
172.	**2.5**	Translators and facilities of languages	Interpreter	"Translates and executes a program one statement at a time."
173.	**2.5**	Translators and facilities of languages	IDE	**Integrated Development Environment:** "A software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger."
174.	**2.5**	Translators and facilities of languages	Error Diagnostics	"These are tools provided by IDE's which give detailed feedback on errors in your code. "
175.	**2.5**	Translators and facilities of languages	Run-Time Environment	"A configuration of hardware and software. It includes the CPU type, operating system and any runtime engines or system software required by a particular category of applications."
176.	**2.6**	Data representation	Bit	"The smallest unit of storage in a computer system, represented by either a binary 1 or 0."
177.	**2.6**	Data representation	Nibble	"Half a byte / 4 bits."
178.	**2.6**	Data representation	Byte	"A collection of eight bits."
179.	**2.6**	Data representation	Kilobyte	"1 kilobyte is 1000 bytes. This is a powers of 10 decimal prefix."
180.	**2.6**	Data representation	Megabyte	"1 megabyte is 1000 kilobytes. This is a powers of 10 decimal prefix."
181.	**2.6**	Data representation	Gigabyte	"1 gigabyte or 1000 megabytes. This is a powers of 10 decimal prefix."
182.	**2.6**	Data representation	Terabyte	"1 terabyte is 1000 gigabytes. This is a powers of 10 decimal prefix."

GCSE Computer Science- Key Terminology

183.	**2.6**	Data representation	Petabyte	"1 petabyte is 1000 terabytes. This is a powers of 10 decimal prefix."
184.	**2.6**	Data representation	Denary	"A numerical system of notation which uses 10 as its base. The 10 Decimal base digits are 0-9."
185.	**2.6**	Data representation	Binary	"Binary describes a numbering scheme in which there are only two possible values for each digit: 0 and 1. The term in computing refers to any digital encoding system in which there are exactly two possible states. E.g. in memory, storage, processing and communications, the 0 and 1 values are sometimes called "low" and "high", respectively."
186.	**2.6**	Data representation	Hexadecimal	"A numerical system of notation which uses 16 rather than 10 as its base. The 16 Hex base digits are 0-9 and the letters A-F."
187.	**2.6**	Data representation	Binary Addition Overflow	"The generation of a number that is too large to be represented in the device meant to store it."
188.	**2.6**	Data representation	Binary Shift	"Allows you to easily multiple and divide base-2 binary numbers. A left shift multiplies by 2 and a right shift divides by 2.
189.	**2.6**	Data representation	Check Digit	"A calculation on data to create a number included with the data for error checking. e.g. check digit = sum of digits DIV 3: 12345 (data) 15/3 = 5 (check digit) = 123455. When the number is input the check digit is recalculated to check it matches. Therefore valid data entry can be assumed."
190.	**2.6**	Data representation	Character-Set	"The set of symbols that may be represented in a computer at a particular time. These symbols, called characters, can be letters, digits, spaces or punctuations marks, the set includes control characters."
191.	**2.6**	Data representation	ASCII	"America Standard Code for Information Interchange: "A character set devised for early telecommunication systems but proved to be ideal for computer systems. ASCII codes use 7-bits giving 32 control codes and 96 displayable characters (the 8th bit is often used for error checking).""
192.	**2.6**	Data representation	Unicode	"Standard character set that replaces the need for all the different character sets. It incorporates characters from almost all the world's languages. It is a 16-bit extension of ASCII."
193.	**2.6**	Data representation	Pixel	"A pixel is the smallest unit of a digital image or graphic that can be displayed and represented on a digital display device. A pixel is represented by a dot or square on a computer monitor display screen."
194.	**2.6**	Data representation	Metadata	"A set of data that describes and gives information about other data."
195.	**2.6**	Data representation	Colour Depth	"Also known as bit depth, is either the number of bits used to indicate the colour of a single pixel, in a bitmapped image or video frame buffer, or the number of bits used for each colour component of a single pixel."
196.	**2.6**	Data representation	Resolution	"The number of pixels (individual points of colour) contained on a display monitor, expressed in terms of the number of pixels on the horizontal axis and the number on the vertical axis."
197.	**2.6**	Data representation	Bit Rate	"The number of bits per second that can be transmitted along a digital network."
198.	**2.6**	Data representation	Sampling Frequency	"Number of samples stored per second. Sample rate multiplied by bit depth. The higher the number the better the quality. The higher the number the larger the file size. CD quality is 44,100 samples per second."

GCSE Computer Science- Key Terminology

199.	**2.6**	Data representation	Compression	"The process of reducing the size of a file in terms of its storage size."
200.	**2.6**	Data representation	Lossy Compression	"A compression scheme where their generally involves a loss of resolution in parts of the image where experiences shows that it will be least noticed."
201.	**2.6**	Data representation	Lossless Compression	"A compression scheme that allows the original images to be recreated."