



GCSE Computing Revision Booklet

This booklet has been created to provide an overview of each of the topics that you need to revise. Each section is broken down and guidance given on what you need to know. Use it in conjunction with your own revision techniques, e.g. mindmaps, to prepare for the exam.

Name:

Fundamentals of Computer Systems

You need to be able to:

- (a) define a computer system
- (b) describe the importance of computer systems in the modern world
- (c) explain the need for reliability in computer systems
- (d) explain the need for adherence to suitable professional standards in the development, use and maintenance of computer systems
- (e) explain the importance of ethical, environmental and legal considerations when creating computer systems.

What is a computer system?

At its very basic, a computer system nothing more than an **input**, a **processor** and an **output**.



A computer system will usually have some **storage** capability and more often now - a way of **communicating** with other devices.

Computers are all around us and I'm not just talking about your desktop or laptop computer. Your "smartphone", your games console, your ipad, even your Sky+ or freeview recorder is a computer system of sorts.



Computer systems are integral parts of our lives. Think about all the computer systems you or your parents use on a daily or weekly basis:



Running washing programmes and sensing when clothes are dry.



On-board computers to control everything from engine efficiency to voice-activated MP3/Bluetooth hands-free system.



Cashless catering system in school to add and subtract money from your account.

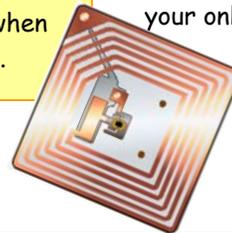


Self-service checkouts in supermarkets that scan, weigh, take payment and give change.



Contactless card payment systems where a small RFID (radio frequency ID) tag is embedded into the card and read when the card is near.

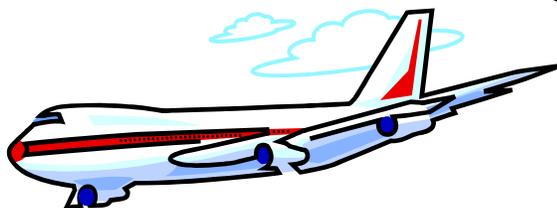
RFID technology - often used in library books and shops to detect theft, also has future applications in "smart" clothes and food products. Imagine the RFID tag in your clothes tells the washing machine how to wash them. Or a smart fridge that knows when the milk is past it's sell-by date and adds it to your online shopping list.



Since we rely so much on technology there are huge problems when it goes wrong! If a supermarket checkout system goes down this could result in lost sales of thousands of pounds.



However, if an automated system, like those onboard aircraft fails then it could end in disaster!



Why do we need to use the same professional standards when developing and maintaining computer systems?

Without standards to adhere to we risk developing systems that are not compatible with each other. Would you be happy if your Nokia mobile could only call other Nokia mobiles? Standards also improve the quality of the computer system and its maintenance by ensuring that best practice is followed and people don't cut corners.

Considerations when creating computer systems:

Legal

The Data Protection Act 1998 provides protection for all information stored on a computer system. It requires the person who owns the system to store the information securely and only use it for the purpose it was originally gathered for, e.g. the school holds data on you for the purpose of your education; it cannot sell this data to companies who may wish to sell you things.



Environmental

The speed of progress means that many gadgets become old and obsolete quite quickly. This means there are potentially stockpiles of old technology which could end up in landfill. Many companies now recycle these by selling reconditioned devices to other countries or cannibalising the parts to be used again.

Websites like Google use huge server-farms around the world. These buildings are full of computers running 24/7. There is an impact on the environment from the huge amount of electricity needed to run these server-farms.



Ethical

Technology enables us to do many things. But are they always right? Google's Street View caused controversy when it was launched after people said that taking photos of their houses was an invasion of privacy. Social networking sites, like Facebook, have made it easier to share information but have made it difficult to set privacy settings. You can now be tracked using GPS technology and Facebook places! Is this the "Big Brother" state?



Computer Software

You need to be able to:

- (a) explain the need for the following functions of an operating system: user interface, memory management, peripheral management, multi-tasking and security
- (b) describe the purpose and use of common utility programs for computer security (antivirus, spyware protection and firewalls), disk organisation (formatting, file transfer, and defragmentation), and system maintenance (system information and diagnosis, system cleanup tools, automatic updating)
- (c) discuss the relative merits of custom written, off the shelf, open source and proprietary software.

Operating System functions

Memory management	The OS manages the transfer of data between the CPU, RAM and main storage (e.g. hard drive)
User interface	The OS provides the user with a way of controlling the functions of the computer without resorting to machine code. It can be graphical (GUI) , e.g. Windows or text-based [no mouse] (CLI - Command Line Interface) , e.g. MS-DOS.
Peripheral management	The OS manages the input and output devices connected to the computer, including things like keyboard, mice and printer.
Multi-tasking	The OS allows more than one program to be run at once so users can switch back and forth between applications easily.
Security	The OS provides security features to restrict access to the system or particular files.

When trying to remember the functions of an Operating System - think MUMPS!
Memory mgt, **U**ser interface, **M**ulti-tasking,
Peripheral mgt and **S**ecurity.



Utility programs

Type	Program	Purpose
Security	Antivirus	Detects and protects the computer against virus threats. Quarantines infected files.
Security	Spyware protection	Detects and removes spyware (small programs that quietly monitor what you are doing)
Security	Firewall	Restrict communication in and out of the computer from the applications that are running.
Disk Operations	Formatting	Sets up a hard drive, memory stick or card into a format that the computer can use to store/retrieve data. The process of formatting erases any data that was previously on the storage medium.
Disk Operations	File Transfer	Allows movement of files from one location to another, either locally or via memory stick/FTP to another computer.
Disk Operations	Defragmentation	Re-organises the data on the hard drive to improve read/write access times.
System Maintenance	System info & diagnosis	Provides information about the system in order to help with diagnosis of problems.
System Maintenance	System cleanup tools	Removes temporary files and compresses/archives unused files to increase free storage space.
System Maintenance	Automatic updating	Uses the Internet to check the OS is up-to-date and applies software patches/fixes as necessary.

Software development

Custom written (a.k.a. bespoke/in- house)	Developed specifically for a particular organisation.
	Adv: custom made to meet exact purpose
	Disadv: very expensive and need technical expertise to develop.
Off-the-shelf	Developed by a software company to be sold to the general public.
	Adv: cheaper than developing it yourself and tech support available.
	Disadv: inflexible - might not do everything you want it to do.
Open source	Written by general public for everyone else to use. Source code provided so software can be tweaked.
	Adv: allows anyone to use/adapt the software to meet their needs, collaborative approach to development which can lead to better quality software.
	Disadv: by making source code available to all, provide opportunity for people to "take advantage" of software installed on users' computers compromise security.
Proprietary	Written by developer and allowed to be used by others but with certain restrictions, e.g. can't adapt software. Source code is hidden.
	Adv: can allow for different versions of same software, e.g. educational version with limited functionality or full-functionality version costing more.
	Disadv: source code cannot be inspected so may contain spyware or other malicious features within the software.

Computer Hardware

Definition: Computer hardware refers to the physical parts of a computer and related devices. Internal hardware devices include **motherboards**, **hard drives**, and **RAM**. External hardware devices include **monitors**, **keyboards**, **mice**, **printers**, and **scanners**.

The internal hardware parts of a computer are often referred to as **components**, while external hardware devices are usually called **peripherals**.

Source: www.techterms.com



Inputs and Outputs (Peripherals)

You need to be able to:

- (a) understand the need for input and output devices
- (b) describe suitable input devices for a wide range of computer controlled situations
- (c) describe suitable output devices for a wide range of computer controlled situations
- (d) discuss input and output devices for users with specific needs.

Inputs and outputs are the computers way of communicating with us - the user. Without inputs or outputs our system would be quite limited. Imagine a calculator which didn't allow you to type in numbers or display the answer on the screen!

Here are some of the common input devices used:

Keyboard



The keyboard, along with the mouse, is one of the most used input devices. Once you get the hang of where the keys are, it allows people to enter text-based data quickly into the computer. A numerical keypad (found on most keyboards on the right) provides a fast way for accountants and people who work in finance to enter numbers.

There are some interesting variations on the keyboard:



Roll-up keyboard
(portable - can be rolled up and stuffed in laptop bag)



Wireless keyboard
(no need for messy wires, frees up user needing to be near computer)



Laser Projection
(projects a keyboard layout on to any surface; sensors detect which "keys" have been pressed)

Mouse



The computer mouse is another common input device. It allows the user to interact with the software on screen. The mouse is quite versatile, with at least two buttons and the ability to move in any direction. However, it can be difficult to be precise - e.g. drawing using the mouse is quite hard!

Mice have also undergone a few changes over the years as well:



Laptop touchpad
(touch-sensitive pads that allow the mouse to be controlled by just a finger)



Trackball
(the user moves the pointer on-screen by moving rolling the ball; the unit doesn't move around like the mouse)



Optical Mouse
(old mice used balls which used to get clogged with dirt and grease - new mice use small cameras to detect movement instead).

Other Input Devices



Scanner

Scanners convert physical documents and photos into digital format. They can be used in conjunction with special software to automatically read handwriting and convert it into computer text. This is called Optical Character Recognition. Some tickbox forms are scanned in, like the Census form or your multiple choice Science exams; this is called Optical Mark Recognition.



Touchscreen

Touchscreens have been around for years in one form or another. The latest versions, like the Apple iPad interface use multi-touch technology to allow greater control.



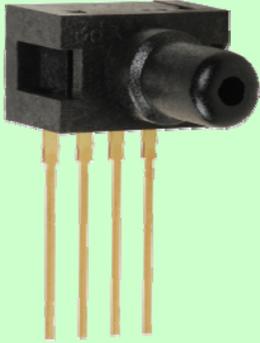
Microphone

Microphones convert analogue sounds into a digital format that the computer can understand and playback. Sounds are usually recorded as .wav or .mp3 files (the latter uses compression to make the file size smaller and is more likely to be used to store music tracks).



Digital Camera

Digital cameras convert live images into digital format. Photos are usually stored as JPEG files. The quality and size of the picture is determined by the number of megapixels the camera is capable of detecting, e.g. photos taken with an 8 megapixel camera are made up of 8 million pixels.

	<p>Webcam</p> <p>Webcams capture live video feeds and using appropriate software can then transmit those across a network.</p>
	<p>Sensors</p> <p>Sensors measure the physical world and translate that into a digital format that a computer can understand. There are various sensors that detect all manner of things: movement, light, heat, moisture, gases (e.g. carbon monoxide levels), location (GPS), etc. For example, the iPhone has an accelerometer so it knows which way round it is to ensure the screen is rotated the correct way.</p>

Output Devices

	<p>Printer</p> <p>Printers produce physical documents and photos from digital files. Most photo printing is done by inkjet printers which can be expensive to run. Laser printers use toner cartridges which are more expensive initially but can print more pages before being replaced.</p>
---	--



Monitor/screen

Monitors/screens provide the visual output from the computer system. Most computer monitors and mobile phones use colour LCD (liquid crystal display).



Speakers

Working the opposite way to the microphone input, the speaker converts digital sound into analogue waves.



3D Printer

A 3D printer uses lasers and a special kind of plastic to build physical 3D shapes from drawings made using CAD software (computer aided design). The object is built up layer by layer. The whole process can take many hours before the final product is finished. They are often used for creating prototypes in manufacturing, e.g. Dyson uses them to create prototype parts for their products. They are still very expensive but the price is coming down.



Motor

A motor can be driven with precision by a computer system. Often used in manufacturing, e.g. building cars, motors are used to control robotic arms.

Input and output devices for specific needs

Keyboards have been adapted for users with limited movement in their arms/hands.



A keyboard designed to minimise the movement of your hands.

Trackballs have also been used where users only have limited motor ability. They can be used by hands or feet, like the one below:



A foot-operated mouse, ideal for people with limited/no mobility in their arms.

For quadriplegic users, there are two alternatives to using the mouse or keyboard. The first is voice-recognition; this involves the user reading a passage of text from the screen so the computer "learns" their voice pattern. They can then use special software to control the computer using voice commands. The technology is getting better but can still make mistakes.

The second is eye-tracking which uses small cameras mounted on the monitor to detect eye movement and move the cursor to the point where the user is gazing. A mouse click is made by blinking slowly.



Stephen Hawking uses a special device to allow him to communicate. An infra-red sensor detects twitches in the cheek muscle below his eyes in order to enter words and then his device generates a computer synthesised voice to read them out loud.

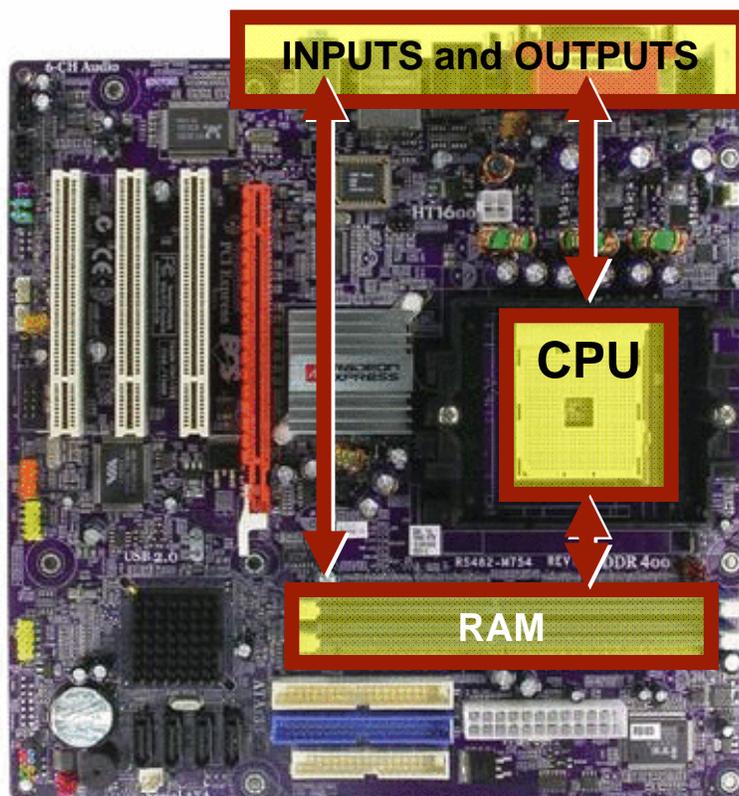
Internal Hardware Devices (Components)

Central Processing Unit (CPU)

You need to be able to:

- (a) state the purpose of the CPU
- (b) describe the function of the CPU as fetching and executing instructions stored in memory
- (c) explain how common characteristics of CPUs such as clock speed, cache size and number of cores affect their performance.

The purpose of the CPU is to **Fetch** and **Execute** instructions stored in memory. It acts as the brain of the computer and controls the rest of the system.



Fetch Execute Cycle (using Little Man Computer)

Download the software: <http://www.cs.ru.nl/~erikpoll/III/lmc>.

The screenshot shows the LMC interface with several callouts:

- Escape Button:** Allows you to quit out of the current program and go back to the Start.
- Phase:** This displays which phase of the fetch/execute cycle we are currently in.
- Program Counter:** The program that is currently being carried out. The current instruction is the one in line with the phase display.
- Instruction Details:** Provides the name of the current instruction, along with its op code and operand. This explains what the little man is doing at each stage as he follows the program.

The interface also shows a calculator with '200' on the display, a 'Memory' chip, and a 'Program' list:

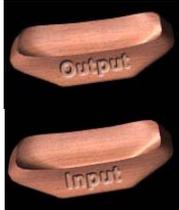
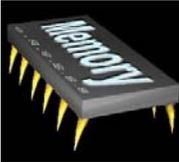
Address	Value
00	500
01	299
02	500
03	399
04	600
05	700
06	000

Instruction Details: Instruction **input** Op Code **5** Operand **00**
 Obtain contents of in tray
 Enter contents into calculator

Fetch phase

<p>Before the little man can carry out an instruction, he needs to find out which instruction he has to carry out.</p>		<p>The Program Counter The little man uses the value in the program counter to decide which address in memory he should get the next instruction from. After getting the value, but before going and getting that instruction, he increments the program counter, so that it has the correct value in it when he visits it next time.</p>
<p>He does this by first getting the value from the program counter (The thing that looks like a green mushroom in a block of wood), and then fetching the value from the corresponding address in memory.</p>		<p>Memory Memory can be used to store programs, and data. In the fetch phase of the fetch execute cycle we are only interested in the fact that we can get instructions from the memory.</p>

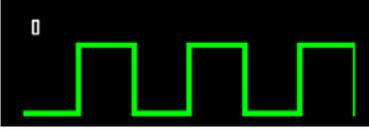
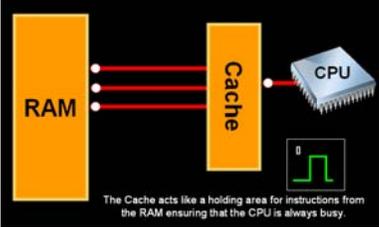
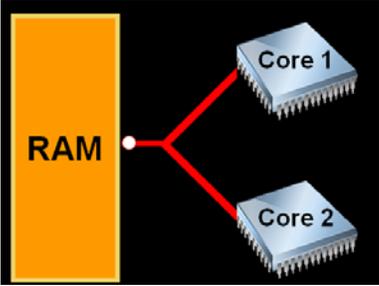
Execute phase

<p>Once the little man knows which instruction he is going to perform, he has to set about actually doing it. This is the 'Execute' phase of the fetch execute cycle.</p>		<p>The Input and Output trays These are what we (the user) use to communicate with the little man. We provide him with data via the input tray, and when he's ready, he provides us with data via the output tray.</p>
<p>For each instruction the little man will perform a series of tasks. For instance, for an input instruction, the little man will go to the input tray, take the value he finds there, and go and put it into the calculator. For an add instruction, he will take a value from memory, and add it to the value presently in the calculator.</p>		<p>Calculator (Arithmetic Logic Unit - ALU) On a 'real' computer, this would be the Arithmetic Logic Unit (ALU). It's basically a calculator that will carry out arithmetic operations.</p>
<p>Once the execute phase is complete, the little man will start the cycle again, and head to the program counter to get the value of the next instruction he has to fetch.</p>		<p>Memory Memory is accessed during the execute phase when we need to store or access data. It is possible to overwrite your programs when putting data into memory though, so be careful where you put it.</p>

Following a program (Decoding instructions)

<p>The program is a series of instructions made up of an op code and operand (either data or the address where data is stored).</p> <p>For example:</p> <div style="display: flex; align-items: center; margin: 10px 0;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;"> <p>Op code 2 means "Store"</p> </div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px; text-align: center;"> <p>2 99</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Operand 99 refers to the memory address where little man should store the data.</p> </div> </div> <p>These can be stored as binary bit patterns, e.g. 10 could refer to the op code (2) above and 11000011 could refer to the memory address (99). Remember 2 in binary is 10 and 99 is 11000011. This would give the bit pattern for the above instruction as 1011000011</p> <p>Q. How does little man (CPU) know the difference between data or an instruction?</p> <p>It all depends on which phase of the cycle the CPU is in, e.g. in the Fetch phase the CPU knows it is an instruction. In the Execute phase the CPU knows it is data.</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">1</td><td>Load</td></tr> <tr><td style="text-align: center;">2</td><td>Store</td></tr> <tr><td style="text-align: center;">3</td><td>Add</td></tr> <tr><td style="text-align: center;">4</td><td>Subtract</td></tr> <tr><td style="text-align: center;">5</td><td>Input</td></tr> <tr><td style="text-align: center;">6</td><td>Output</td></tr> <tr><td style="text-align: center;">7</td><td>Halt</td></tr> <tr><td style="text-align: center;">800</td><td>Skip If Negative</td></tr> <tr><td style="text-align: center;">801</td><td>Skip If Zero</td></tr> <tr><td style="text-align: center;">802</td><td>Skip If Positive</td></tr> <tr><td style="text-align: center;">9</td><td>Jump</td></tr> </table> <p style="text-align: center;">Op codes understood by LMC</p>	1	Load	2	Store	3	Add	4	Subtract	5	Input	6	Output	7	Halt	800	Skip If Negative	801	Skip If Zero	802	Skip If Positive	9	Jump
1	Load																						
2	Store																						
3	Add																						
4	Subtract																						
5	Input																						
6	Output																						
7	Halt																						
800	Skip If Negative																						
801	Skip If Zero																						
802	Skip If Positive																						
9	Jump																						

Characteristics which affect performance

<p>Clock Speed</p> 	<p>This is the number of cycles that the CPU can complete in one second (measured in Hertz or Hz)</p>	<p>A faster clock speed means the CPU can process more instructions per second. For example a 1MHz processor can process 1 million instructions a second; whereas a 1GHz processor can do 1000 million instructions in the same time.</p>
<p>Cache size</p>  <p>The Cache acts like a holding area for instructions from the RAM ensuring that the CPU is always busy.</p>	<p>This is the holding area for data from the RAM. It ensures the CPU is always busy with data.</p>	<p>A larger cache size means more data can be held and it's less likely that any cycles are wasted. A smaller cache size and fast clock speed may result in a bottleneck between the RAM and CPU.</p>
<p>Number of Cores</p> 	<p>This is the number of processing cores the CPU has. A dual-core has two and is able to process two instructions simultaneously in the same cycle.</p>	<p>A dual-core processor or quad-core processor is 2 to 4 times faster than a conventional chip as they can process simultaneous instructions.</p>



It is important to balance all of the above to avoid "over-specing" one area and creating bottlenecks elsewhere.

Memory

You need to be able to:

- describe the difference between RAM and ROM
- explain the need for ROM in a computer system
- describe the purpose of RAM in a computer system
- explain how the amount of RAM in a personal computer affects the performance of the computer
- explain the need for virtual memory
- describe cache memory
- describe flash memory
- discuss how changes in memory technologies are leading to innovative computer designs.

RAM and ROM

Both are needed in a computer system but for different reasons. In a computer computer, **ROM** (Read Only Memory) is used to store all the instructions (the **BIOS** - Basic Input/Output System) to get the computer up and running. It seeks out the amount of memory, input and outputs available to it and then loads up the operating system. In some computer systems, like your washing machine, the ROM stores the software to control the hardware, as well as data such as cooking/washing times etc.



RAM (Random Access Memory) is used as temporary storage for instructions and data between the **inputs**, **outputs**, **secondary storage** (e.g. hard drive) and **CPU**. The larger the RAM, the more instructions/data can be held in quick access memory without needing to use **Virtual memory**; this can improve performance.



Random Access Memory	Volatile (forgets everything when power switched off)	Can be written to and read from	Usually measured in Gigabytes (Gb)
Read Only Memory	Non-volatile (remembers content even if power switched off)	Can only be read from (not written to)	For the BIOS, usually measured in Kilobytes (Kb)

Virtual Memory

Depending on the size of the RAM, it is not always possible to store all the running programs in the RAM at once and the computer will start running low on memory and slow down. This is where virtual memory comes in. It stores the data that the RAM isn't actively using on the hard drive to free up the RAM. Then, when the CPU needs the data it pulls it back from the virtual memory on the hard drive. As far as the CPU is concerned, all the programs are running from the RAM.

The operating system sets up a page file on the hard drive to be used for virtual memory. This is usually set at twice the size of the amount of RAM, e.g. 1GB of RAM should have a 2GB page file.

Cache Memory

This has been explained in the CPU section on page 17.

Flash Memory

Flash Memory is often referred to as solid-state storage, which means that there are no moving parts. It's similar to RAM but is non-volatile which means when the power is cut off, it still remembers the content). Flash memory is used in USB memory sticks and memory cards for digital cameras and some games consoles, e.g. Sony PSP.

Future developments in Memory technologies

Physical Size & Capacity	Memory chips have become more compact (storing more bits per chip) which has led to smaller computing devices and larger capacities.
Costs	The cost of memory has fallen which means it is now affordable to have large amounts of RAM available.
Speed	Read/write times are improving which mean faster access and better performance.
Power	Memory chips are more power-efficient meaning portable devices work for longer between charges and batteries can be smaller and lighter.

Secondary Storage

You need to be able to:

- explain the need for secondary storage
- describe common storage technologies such as optical, magnetic and solid state
- select suitable storage devices and storage media for a given application and justify their choice using characteristics such as capacity, speed, portability, durability and reliability.

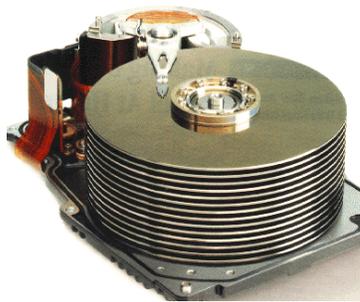
Secondary storage is needed for when we want to store files permanently or semi-permanently. We might want to transfer files between computers by using portable secondary storage media.

Storage Technologies



Optical

Codes data using microscopic dips on the surface of the disc.



Magnetic

Codes data using small changes in magnetic fields.



Solid State

Uses non-volatile memory chips to store bits of data

Units

Candidates should be able to:

- define the terms bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte

Bit	Nibble	Byte	Kilobyte (Kb)	Megabyte (Mb)	Gigabyte (Gb)	Terabyte (Tb)
1	4 bits	8 bits	1,024 bytes	1,024 kilobytes	1,024 megabytes	1,024 gigabytes

Storage Media Comparison

Storage Medium	Method	Capacity	Speed	Portability	Durability and Reliability
CD	Optical	Medium 650-700Mb	Fast access	Lightweight and flat	Easily scratched and will then not work. Can only be written to once.
Floppy Disk	Magnetic	Small 1.44Mb	Slow access	Lightweight and flat	Can be snapped. Can be damaged by strong magnets or heat.
Internal Hard Disk	Magnetic	Large 1Gb to 2+ Tb	Fast access (dependent on speed of motor RPM - rotations per minute)	Heavy and not designed to be carried around. Requires integrating into computer system.	Quite reliable but can crash - when the read/write head crashes into the platter. Sometimes happens on laptops when dropped. Can be damaged by magnets.
Portable Hard Disk	Magnetic	Large 40Gb to 640Gb	Medium-fast access (dependent on connection method, e.g. USB cable)	Medium-weight. Plugs into computer USB or Firewire port.	Same concerns as internal hard drive.
Flash drive (USB stick or memory card)	Solid-state	Medium/Large 1Gb to 64Gb	Medium-fast access (dependent on connection method, e.g. USB cable or card reader)	Lightweight and small.	Not affected by magnets. Quite hardy. Can be dropped without damaging contents.
Blu-ray disc	Optical	Large 50Gb	Fast access	Lightweight and flat.	Easily scratched and will then not work.
DVD (Digital Versatile Disc)	Optical	Large 4.7Gb	Fast access	Lightweight and flat.	Easily scratched and will then not work.

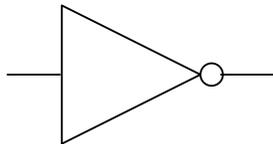
Binary Logic

You need to be able to:

- explain why data is represented in computer systems in binary form
- understand and produce simple logic diagrams using the operations NOT, AND and OR
- produce a truth table from a given logic diagram.

Computers use binary as it's a lot simpler. Each CPU is made up of millions of transistors which can only have two states (ON/OFF). Anything can be converted into a binary number so a computer can understand, process and store it.

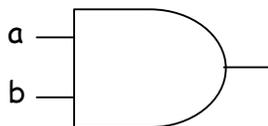
NOT Gate



Flips the input, e.g. 0 becomes 1 and 1 becomes 0.

NOT Truth Table	
Input	Output
0	1
1	0

AND Gate



Waits for 1 on both inputs before outputting a 1.

AND Truth Table		
Input a	Input b	Output
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate



Wait for 1 on either input before outputting a 1.

OR Truth Table		
Input a	Input b	Output
0	0	0
0	1	1
1	0	1
1	1	1

You can practice with NOT, AND and OR gates using the following website:
<http://logic.ly/demo/>

Representation of data in computer systems

Numbers

You need to be able to:

- convert positive denary whole numbers (0-255) into 8-bit binary numbers and vice versa
- add two 8-bit binary integers and explain overflow errors which may occur
- convert positive denary whole numbers (0-255) into 2-digit hexadecimal numbers and vice versa
- convert between binary and hexadecimal equivalents of the same number
- explain the use of hexadecimal numbers to represent binary numbers.

Denary and Binary conversions

Numbers greater than 1 can be represented in binary by using more bits.

e.g. 129 can be represented in 8 bit (byte) binary as 10000001.

Denary	128	64	32	16	8	4	2	1
Binary	1	0	0	0	0	0	0	1

This means that if you put a 1 on each bit (i.e. 1111111) the highest denary number 8 bits (a byte) can store is 255; $128+64+32+16+8+4+2+1 = 255$.

Notice how the number increased by double for each new bit added to the left.



To convert binary into denary, just add together the denary numbers above each bit that has 1 in it, e.g. $10000001 = 128 + 1 = 129$.

To convert a denary number into binary, find the highest value that will go into it and then continue down the number of bits with the remainder until there is nothing remaining.

For example, to convert 44 into binary 8-bit format:

44 is smaller than 128 and 64 so we place 0s in those bits:

Denary	128	64
Binary	0	0

32 does go into 44 with remainder of 12, so we place a 1 in that bit:

Denary	128	64	32
Binary	0	0	1

The remainder of 12 is smaller than 16 so we place a 0 in that bit:

Denary	128	64	32	16
Binary	0	0	1	0

8 does go into 12 with remainder of 4, so we place a 1 in that bit:

Denary	128	64	32	16	8
Binary	0	0	1	0	1

4 does go into 4 with with nothing remaining, so we place a 1 in that bit:

Denary	128	64	32	16	8	4
Binary	0	0	1	0	1	1

and 0s in all the remaining bits:

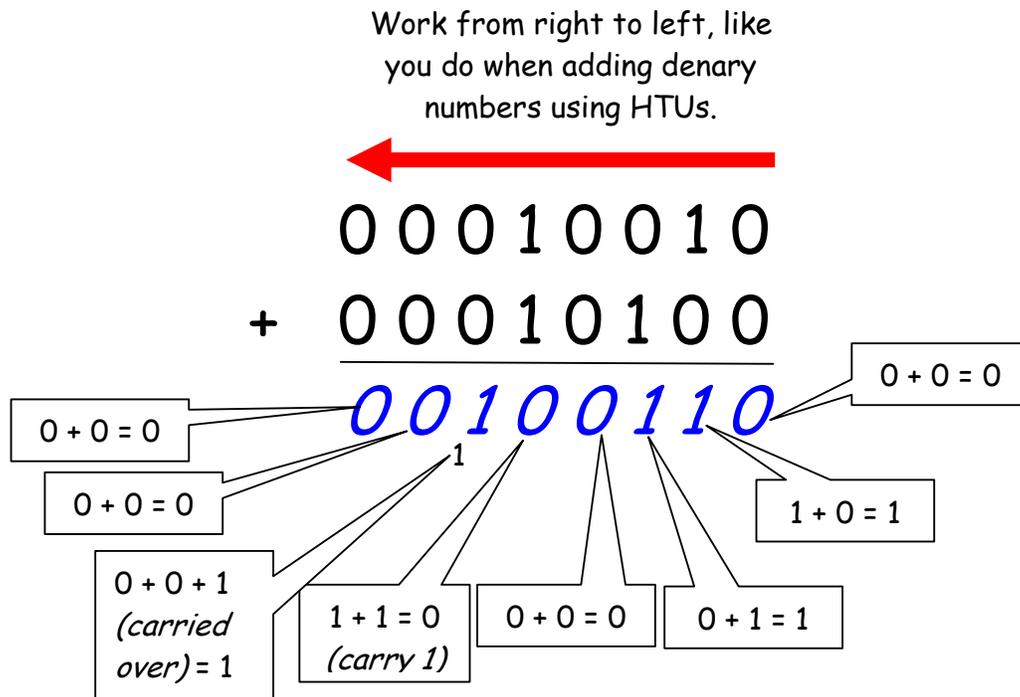
Denary	128	64	32	16	8	4	2	1
Binary	0	0	1	0	1	1	0	0

Therefore 44 as an 8-bit binary number is: 00101100.

Adding two binary numbers together

Adding two binary numbers together is not as scary as it sounds or looks. It uses the same principles at the denary number system (Hundreds, Tens, Units).

For example:



Let's convert it back into denary to check it is correct:

Denary	128	64	32	16	8	4	2	1
Binary	0	0	0	1	0	0	1	0

$$16 + 2 = 18$$

Denary	128	64	32	16	8	4	2	1
Binary	0	0	0	1	0	1	0	0

$$16 + 4 = 20$$

Denary	128	64	32	16	8	4	2	1
Binary	0	0	1	0	0	1	1	0

$$32 + 4 + 2 = 38$$

So as $18 + 20 = 38$. Then we know the binary calculation is correct!

Hexadecimal Numbers

Programmers started to use hexadecimal to reduce mistakes and make it easier for them to understand than using lots of 0s and 1s.

Denary and hexadecimal conversions

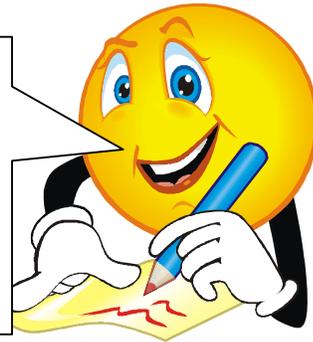
Hexadecimal uses a base 16 system = 0-9 and then A-F.

That means that the denary number 15 is equal to F in hexadecimal. The denary number 16 is equal to 10 in hexadecimal (which means 1 lot of 16)

The highest denary number represented in 8-bit (byte) binary is 255 and was represented by 11111111. In hexadecimal this would be represented as FF (15 lots of 16 + 15).

Confused?!

- ❑ Remember, F in hexadecimal represents 15.
- ❑ Therefore F0 would be $15 \times 16 = 240$.
- ❑ The second F is 15, therefore FF = $240 + 15 = 255$.



To convert denary to hexadecimal you can divide the number by 16.

For example, to convert 141 into hexadecimal do the following:

$$141 / 16 = 8 \text{ remainder } 13.$$

$$8 / 16 = 0 \text{ remainder } 8.$$

Therefore, working from right to left, the hexadecimal number is 8 13 (and 13 is represented as D in hex) so it would be 8D.

Alternatively, convert the denary number into binary and then binary to hexadecimal.

Binary to hexadecimal conversions

To convert binary to hexadecimal you need to break it down into nibbles (blocks of 4 bits).

For example, 141 as a binary number is: 10001101.

The two nibbles are 1000 and 1101. Converting each nibble into denary we can see that 1000 = 8 and 1101 = 13. Remembering that 13 = D in hexadecimal then 141 would be represented as 8D.

To convert hexadecimal to binary you just reverse the process. Convert each part of the hexadecimal number into nibbles of binary numbers.

For example:

Hex	8				D			
Denary	8	4	2	1	8	4	2	1
Binary	1	0	0	0	1	1	0	1

}
}

nibble
nibble



This conversion chart below may help to make sense of it all!

Denary	Binary	Hexadecimal	Denary	Binary	Hexadecimal
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

Using binary to represent characters

You need to be able to:

- (a) explain the use of binary codes to represent characters
- (b) explain the term character set
- (c) describe with examples (for example ASCII and Unicode) the relationship between the number of bits per character in a character set and the number of characters which can be represented.

Computers use the ASCII (American Standard Code for Information Interchange) format to use binary codes to represent numbers, letters and special characters. It uses 8-bits (byte) to represent each character, so every character has its own unique 8-bit ASCII code.

Using the ASCII character set on the next page we can convert characters into binary numbers.

For example, Isambard would be represented in binary as:

I	=	73	=	0100 1001
s	=	115	=	0111 0011
a	=	97	=	0110 0001
m	=	109	=	0110 1101
b	=	98	=	0110 0010
a	=	97	=	0110 0001
r	=	114	=	0111 0010
d	=	100	=	0110 0100

Notice how the capital I is different from the lowercase i in the ASCII format. An upper and lower case version of each letter is present in the chart to enable the computer to tell the difference between them.

If you remember what we said about binary numbers back on page 23 then you know that 8-bits (a byte) can store up to 255 in denary. This means that the ASCII format can have up to 255 characters represented in a character set.

Character	Decimal Number	Binary Number	Character	Decimal Number	Binary Number
blank space	32	0010 0000	^	94	0101 1110
!	33	0010 0001	-	95	0101 1111
“	34	0010 0010	`	96	0110 0000
#	35	0010 0011	a	97	0110 0001
\$	36	0010 0100	b	98	0110 0010
A	65	0100 0001	c	99	0110 0011
B	66	0100 0010	d	100	0110 0100
C	67	0100 0011	e	101	0110 0101
D	68	0100 0100	f	102	0110 0110
E	69	0100 0101	g	103	0110 0111
F	70	0100 0110	h	104	0110 1000
G	71	0100 0111	i	105	0110 1001
H	72	0100 1000	j	106	0110 1010
I	73	0100 1001	k	107	0110 1011
J	74	0100 1010	l	108	0110 1100
K	75	0100 1011	m	109	0110 1101
L	76	0100 1100	n	110	0110 1110
M	77	0100 1101	o	111	0110 1111
N	78	0100 1110	p	112	0111 0000
O	79	0100 1111	q	113	0111 0001
P	80	0101 0000	r	114	0111 0010
Q	81	0101 0001	s	115	0111 0011
R	82	0101 0010	t	116	0111 0100
S	83	0101 0011	u	117	0111 0101
T	84	0101 0100	v	118	0111 0110
U	85	0101 0101	w	119	0111 0111
V	86	0101 0110	x	120	0111 1000
W	87	0101 0111	y	121	0111 1001
X	88	0101 1000	z	122	0111 1010
Y	89	0101 1001	{	123	0111 1011
Z	90	0101 1010		124	0111 1100
[91	0101 1011	}	125	0111 1101
/	92	0101 1100	~	126	0111 1110
]	93	0101 1101			

American Standard Code for Information Interchange character set

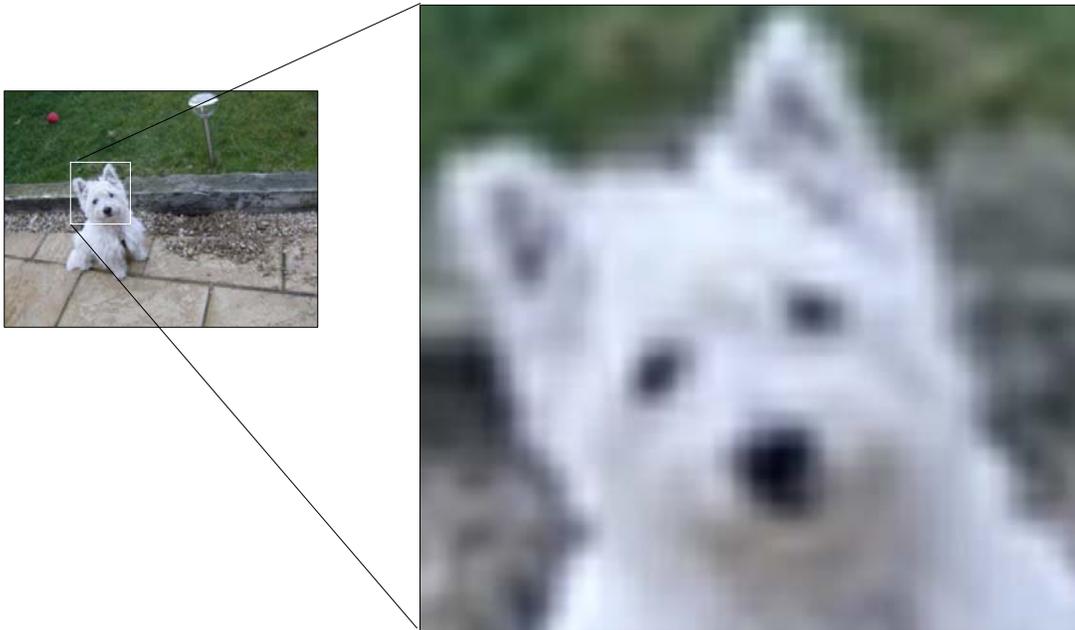
Note: You don't need to remember these codes for your exam but just be aware of how it works.

Using binary to represent images

You need to be able to:

- (a) explain the representation of an image as a series of pixels represented in binary
- (b) explain the need for metadata to be included in the file such as height, width and colour depth
- (c) discuss the effect of colour depth and resolution on the size of an image file.

Images on the computer are made up of lots of individual pixels (small squares of colour). If you zoom into any image enough times you will see these little blocks of colour.



We can use binary numbers to represent each pixel.

In a simple image, where there are only two colours (colour depth of 1) then we can simply use a 1 for one colour, and 0 for another.

For example, look at this small 3 × 3 pixel image below. The 1s represent the black colour and the 0s represent the white.

1	0	0
0	1	0
0	0	1

That is the principle behind storing images as binary. However, we also need to know a few other things about the image in order to display it correctly, i.e. its height, width and colour depth.

Image Metadata

Height	The height of the image in pixels
Width	The width of the image in pixels
Colour depth	The number of bits available to represent each pixel
Resolution	The pixels per inch (PPI) - if the pixels per inch is high, then the resolution and image quality is also high.

Without this metadata the computer would not be able to reconstruct the image properly.



Colour Depth

In a simple 1-bit colour depth image, we can only display two colours. However, in an 8-bit colour depth image (like the gif format), we can display 256 colours (one for each binary number from 00000000 to 11111111).

In a 24-bit colour depth image (like the jpeg format), we can display just over 16 million colours!

If we use more bits to store each pixel, then the file size will be larger. That is why .gif images are often smaller in file size than .jpeg images.

Resolution

If the image has a **higher resolution**, it has **more pixels** to store data about and is therefore **larger** in size.

If the image is **lower resolution**, this means that there are **less pixels** per inch, which means less pixels to store data about and is therefore **smaller** in size.

Using binary to represent sound

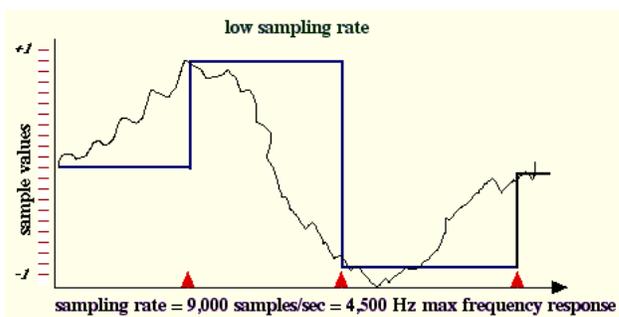
You need to be able to:

- (a) explain how sound can be sampled and stored in digital form
- (b) explain how sampling intervals and other considerations affect the size of a sound file and the quality of its playback.

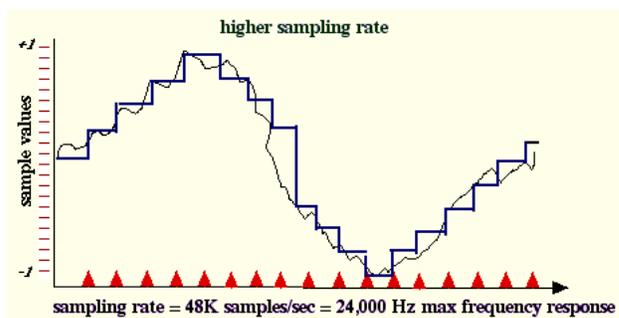
Sound is stored in the computer by taking samples of the sound wave at regular intervals and converting those into binary numbers (using an analogue-to-digital converter). The computer can then re-create the sound by reversing the process.

The file size and quality of the sound recording is affected by the sample rate and bit rate.

The **sample rate** refers to the number of samples taken each second. It is measured in Hertz (like the CPU clock speed).



If the sound wave is not sampled at a higher enough rate, this can **lower** the **quality** of the sound recording, but the **file size** would be **smaller**.



If the sound wave is sampled at a higher sampling rate, this will **increase** the **quality** of the recording, but the **file size** will be much **larger**.

The **bit rate** refers to the amount of bits used to store the different levels of sound at each sampling interval. The more bits, the greater range of levels that can be distinguished. This also has the affect of **increasing** the **file size** if the bit rate is high (e.g. 16-bit instead of 8-bit).

Computer Communications and Networks

Networks

You need to be able to:

- (a) explain the advantages of networking stand-alone computers into a local area network
- (b) describe the hardware needed to connect stand-alone computers into a local area network, including hub/switches, wireless access points
- (c) explain the different roles of computers in a client-server and a peer-to-peer network
- (d) describe, using diagrams or otherwise, the ring, bus and star network topologies
- (e) describe the differences between a local area network and a wide area network such as the Internet
- (f) explain the terms IP addressing, MAC addressing, packet and protocols
- (g) explain the need for security measures in networks, such as user access levels, suitable passwords and encryption techniques
- (h) describe and justify network policies such as acceptable use, disaster recovery, failover, back up, archiving.

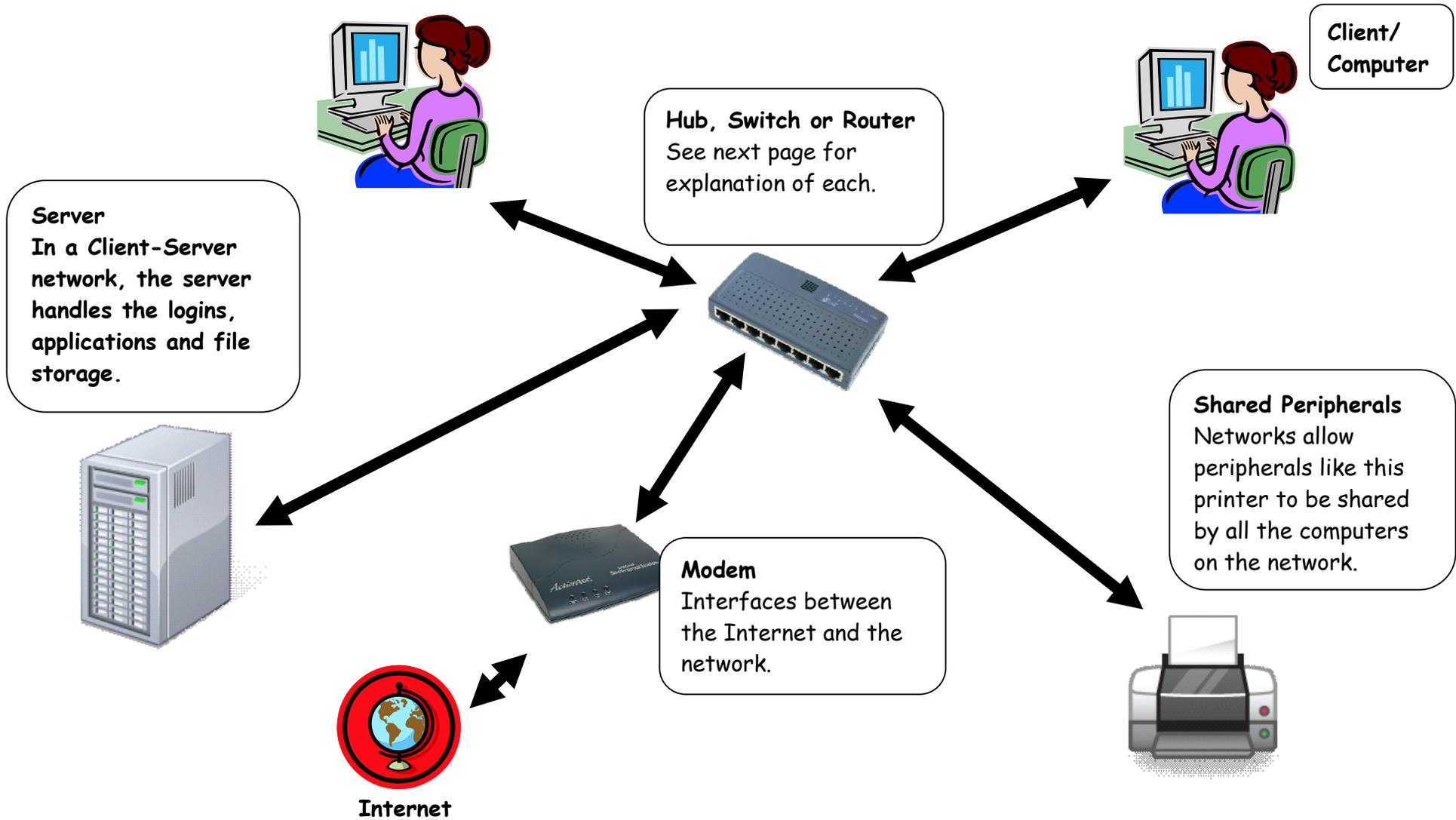
Why network?

Allows computers to communicate with each other and share resources, e.g. peripherals (printers, scanners), storage, internet access.

Data Packets

Information is sent around a network in data packets. Data is broken down into packets to be sent over the network and often may take different routes to get there. It contains info on where it needs to go (the address) and the packet number and an error-checking facility to ensure the packet arrives intact and can be put back together in the right order.

Basic components of a local area network (LAN)



Hub, Switch or Router?

These are not the same. You need to know the difference between them.

Hub	A hub receives all data sent over the network and sends it to all the other devices on the network. Only the device who the data packet is for will acknowledge it; the other devices will ignore it.
Switch	A switch is bit more intelligent than a hub and it learns which devices are on the network (and their port number/address) and only sends the data packet to the computer who is the intended recipient . This is more efficient than a hub as it doesn't result in lots of unnecessary network traffic.
Router	A router is cleverer still than both hubs and switches. It can re-direct network traffic and deal with different packets in different ways. It can protect the computers on the network by hiding them from the outside world (i.e. a firewall). Some routers have a modem built-in and wireless capabilities.



Most homes have a 4-port wireless broadband router to access the Internet. This combines the router and modem functions.

Wireless access points don't require a physical connection and therefore are suited for a variety of different devices and provide access to network over an area, e.g. iPad or smartphone.



IP address

The **IP address (Internet Protocol)** is a computer's unique address on the network, e.g. 192.168.0.1 is usually your home router's address. On the Internet, each website has its own or several IP addresses.

MAC address

The **MAC address (Media Access Control)** is a hexadecimal number that is unique to that particular device.



Hey! Did you know that there are 281,474,976,710,656 possible MAC addresses in the world? That's a lot of MACs baby... huh huh!



Network Protocols

Protocols are a set of standards and guidelines that tell computers how they should handle the sending and receiving of data over the network.

TCP/IP (Transmission Control Protocol and Internet Protocol)	Used to exchanged data between computers on a network and route packets between networks and over the Internet.
HTTP (HyperText transfer Protocol)	Used on WWW to transfer webpages and web content from the website host server to the computer requesting the page.
FTP (File Transfer protocol)	Used to transfer files between computers over a network.
SMTP (Simple Mail Transport Protocol)	Used to transport emails.

LAN or WAN?

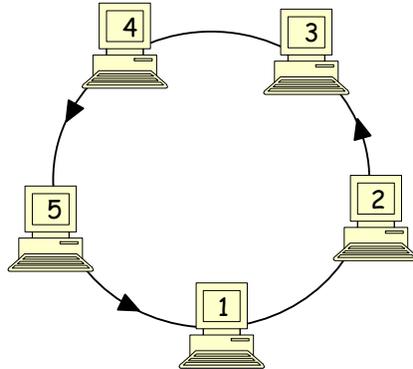
A LAN is a Local Area Network which is usually set-up within a certain area, e.g. a building or group of buildings - hence the term "local".

A WAN is a Wide Area Network which is usually on a nationwide or worldwide scale. The Internet is an example of a WAN - a series of interconnected LANs.

Network Topologies

A network can be set up in three main ways:

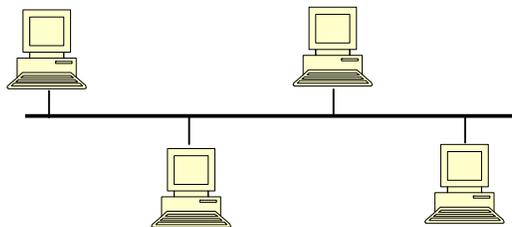
Ring



All the computers are linked together in a ring and data packets are sent one way round. For example in the diagram above to get from PC 1 to PC 5 the data needs to go through PCs 2, 3 and 4. It can't go anti-clockwise back to PC 5.

Drawbacks: If there is one break in the ring then the network breaks down. Also not very efficient as a lot of network traffic doesn't take the most direct route.

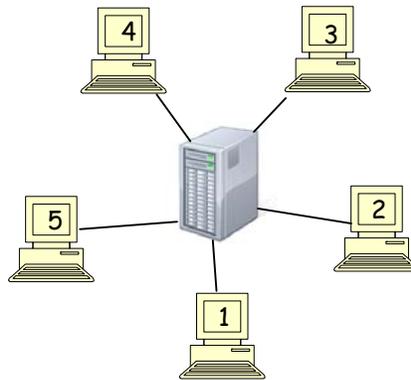
Bus



All the computers are connected to a central bus. You can add and take away computers easily in a bus network as the computers are connected to the central bus, rather than between one another.

Drawbacks: Only one computer can use the bus at once which means only one computer can communicate at a time - the others have to wait until the bus is free before sending their data; this makes it slow. If a computer connected to the bus breaks down, then the network still runs but if the central bus breaks down then so does the whole network.

Star



All the computers are connected to central server. The server can handle multiple communications at once and re-directs traffic over the network. More computers can be added or taken away without affecting the network.

Drawbacks: If the main server fails then the network goes down.

Client-server and peer-to-peer networks

In a Client-server network, the server acts as the main computer handling login requests, user storage, etc. The client could be any computer. The user can log on to any client and still access their stuff on the server.

In a Peer-to-peer network, each computer shares information equally and plays an equal role in the network.

Network Security

Part of Data Protection Act 1998 requires companies who store data electronically to keep it safe and secure.



Without security then anyone could hack into the network and intercept data packets. Confidential data packets should be encrypted so if they are intercepted they are unreadable.

There may be a need to prevent unauthorised access to particular areas of the system; this could be controlled by passwords and different user levels, e.g. in the school a student can only access the student shared drive, whereas teachers can access both the student shared drive and the staff shared drive.

Network Management Policies

Disaster recovery

In the event of the system failing there needs to be plans in place to get the network up and running again as quickly as possible. Many organisations, such as banks, have a disaster recovery plan which may mean relocating to a different location (if the building is no longer accessible).



Failover and backups

If there is a power cut, many networks have UPS (uninterrupted power supplies) which are large batteries with enough charge to keep the network going long enough to shut it down correctly.

If a hard drive fails, there is usually a backup which can take the strain. Networks are often backed up at least daily and these backups are stored off-site or in a fire-proof safe so in the event of a disaster in the building, the data is still intact.



Maintenance

Sometimes data needs to be kept but is not accessed regularly. This data can be archived off onto tape to help free up space on the network.

Acceptable Use

It is important for the network to have an acceptable usage policy to provide users with guidelines about what they can and can't do using it. For example, it would use up vital network bandwidth if some users decided to play a network game or download the latest torrent from the Internet. If users start file sharing copyrighted materials, like music, using the network then there is a legal issue to consider as well.



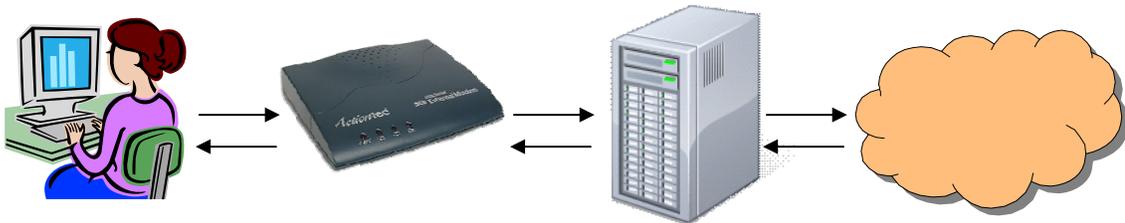
The Internet

You need to be able to:

- (a) describe the nature of the Internet as a worldwide collection of computer networks
- (b) describe the hardware needed to connect to the Internet including modems, routers etc
- (c) explain the need for IP addressing of resources on the Internet and how this can be facilitated by the role of DNS servers
- (d) explain the importance of HTML and its derivatives as a standard for the creation of web pages
- (e) describe common file standards associated with the Internet such as JPG, GIF, PDF, MP3, MPEG
- (f) explain the importance of compressing files that are transmitted via the Internet
- (g) describe the differences between lossy and lossless compression.

What happens when you type in a web address in your browser?

When you enter a web address in your browser, it heads off via your modem to your ISP's (Internet Service Provider) domain name servers (DNS). These computers store the IP address(es) for each website and directs your request to the relevant server hosting the webpage you want to view. The information is then sent back to your ISP and then back to your computer.



Why HTML?

HTML (HyperText Markup Language) provides a standard format which any computer with web browser can interpret and display the pages. It allowed the World Wide Web to get going as there was a universal standard that everyone could now use to share information.

Common file formats used on the Internet

.html	HTML Web page	Used to store webpages
.jpg	JPEG image	Used to store photos as can show up to 16.7 million colours.
.gif	Graphic Interchange format image	Used for simple graphics and short animations - can show up to 256 colours.
.pdf	Adobe Portable Document format	Used to store non-editable documents for viewing on the Internet using the free Adobe viewer software.
.mp3	MP3 Music format	Used to store music tracks. Compressed to make downloading faster.
.mpg	MPEG video format	Used to store videos. Compressed to make downloading faster.
.swf	Shockwave Flash format	Used for Adobe Flash files to add interaction and/or animations to websites.
.mov	Apple Quicktime format	Used to store video files playable using free Apple viewer software.

Why compress on the Internet?

The Internet can be slow at times, especially in it's early days relying on early 56Kbps modems as opposed to the 3-4 Mbps average. Therefore if you could compress the files that were being sent on the Internet, then you sent smaller files, and smaller file sizes meant faster downloads. It also frees up the network and avoids clogging up the bandwidth.

Lossless versus Lossy Compression

Lossless compression	Compression used means the original file stays completely intact and unchanged.	Used for text documents, data files and executable programs where it would not be desirable for parts to be lost.
Lossy compression	Compression which involves discarding parts of the file which won't be noticed if they are not present.	Used for multimedia files, i.e. sound, images, video.

For example, it is possible to remove some data within an image and your brain fills in the gaps.

Take a look at the images of the dog below:



Source: wikipedia.org

The image has more compression applied to it as you go from left to right. Even though the image on the right is heavily compressed, we can still see it is a dog. Our brain fills in the gaps.

Databases

What is a database?

A database is a persistent organised store of data which can be easily and quickly searched.

They are most commonly used for Marketing, e.g. loyalty cards. Huge databases are searched for patterns in customer's buying habits and then targeted coupons are mail shot to customers.



The school has several databases to store everything from your *behaviour logs* and *Brunels*, to your address, date of birth and even how much money you have on your lunch card!

Data handling software, such as Microsoft Access, allow users to:

- add tables, define fields and validation (create)
- add/edit/delete data (maintain)
- run queries to extract information from the database (interrogate)

DBMS (Database Management System)

Many large databases use a database management system (DBMS). This system separates the data stored from the data handling applications used to maintain and interrogate the database.

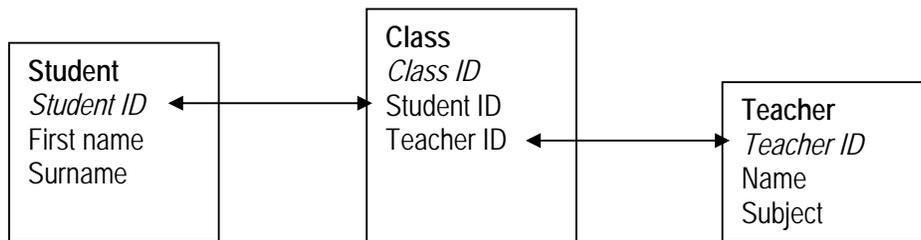
Feature	Why is this good?
Concurrent usage	Allows multiple users to use the database at the same time
Security	Provides ability to allow/restrict data access to individuals or applications
Recovery & Backup	Provides backup solution and recovery tools to ensure data is not lost in event of failure
Data integrity	Enforcing data integrity by ensuring data is consistent and adheres to the rules

Relational databases

A flat-file database is when all of the data and fields are stored in one table, e.g. an excel spreadsheet full of names and addresses - see below:

	A	B	C	D	E	F	G
1	ID	First name	Surname	House No	Street	Town	Postcode
2		1 Frank	Smith	13	Swindon Road	Reading	RG115TG
3		2 Bob	Field	76	High Street	Swindon	SN29GD
4		3 Mark	Grady	3	Fordingbridge Close	Salisbury	SP11QW
5		4 Sarah	Phillips	100	Cricklade Road	Cricklade	SN241RT
6		5 Nick	Truman	16	Lawn Close	Swindon	SN16HU

A relational database is when that data is separated into individual tables (entities) that are linked by a primary key (which appears in each table).



In the example above, the **STUDENT**, **CLASS** and **TEACHER** are all tables (entities) in the database. The **id** is the unique piece of information (the **primary key**) that identifies each instance of a student, class or teacher. We couldn't use *First Name* as the primary key in **STUDENT** as many people share the same first name - it isn't unique enough!

The advantage of separating the data into different entities is to **avoid data redundancy** - when the same data is stored in several places in the database. Let's imagine if we stored the information above in a flat-file database (all in one table) - it would look something like this:

	A	B	C	D	E	F	G
1	Student ID	First name	Surname	Class ID	Teacher ID	Name	Subject
2		1 Frank	Smith	7AN	JXP	Miss Parsons	ICT
3		2 Peter	Truman	7AN	JXP	Miss Parsons	ICT
4		3 Harry	Grimes	7AN	JXP	Miss Parsons	ICT

Duplicated data

Can you see the duplicated information? Not only does this use up space but it is easier to make a mistake (and lose **data integrity**) if we only update one of the records. E.g. if Miss Parsons gets married and her name changes to Mrs Smith then we need to make this change for every record in the database. If we are using a relational database, because the *Teacher's name* is stored in a separate table and linked to the rest of the entities, we only need to update it in the **TEACHER** table once.

In the **CLASS** entity, both Teacher ID and Student ID are **foreign keys** (primary keys that when linked to another table become foreign keys in that table).

Components of a relational database

Tables	Represents the entity in the database. Fields and primary keys are defined and data stored in the table. (Create)
Forms	Provide a customisable user interface to add/edit and delete data (Maintain)
Queries	Allow data that meets specific criteria to be extracted and displayed (Interrogate)
Reports	Displays the results of queries in a customised format designed to be printed

Querying a database

When interrogating a database, you need to specify the fields you wish to display and the criterion you will use to search. E.g. if you were searching for a car on a car supermarket website you might search using the make of the car you wanted:

Make = "Ford"

You could also use logical operators such as <> (NOT equal to), < (less than), > (greater than), <= (less than or equal to) and >= (greater than or equal to). E.g.

Mileage <= 50000

If using more than one criterion to interrogate your database, then you can choose whether to make it AND or OR. E.g.

Make = "Ford" AND Mileage <= 50000

[This pulls back Ford cars with 50,000 miles or less on the milometer]

Make = "Ford" OR Mileage <= 50000

*[This pulls back all Ford cars - regardless of the mileage;
and all cars with 50,000 or less mileage - regardless of the make]*

Data Validation

We need to prevent incorrect or inaccurate data from entering our database in the first place otherwise incorrect data will be extracted from the database later on. This is the *GIGO* principle (*Garbage In, Garbage Out*) which basically means if you put rubbish data into the system, you will get nonsense out again, e.g. allowing someone to enter "carrots" as their telephone number in the database and then printing that on subsequent reports, etc.

We can restrict the type and range of data that is entered in a database.

Data Types	These tell the database what format of data to expect, e.g. text, integers, decimals, Boolean, list.
Data Range	Rules can be set up to only allow certain values, e.g. between 0 and 10, >1000 or ensuring the text entered starts with a "0" and has a particular length (useful for telephone number validation)