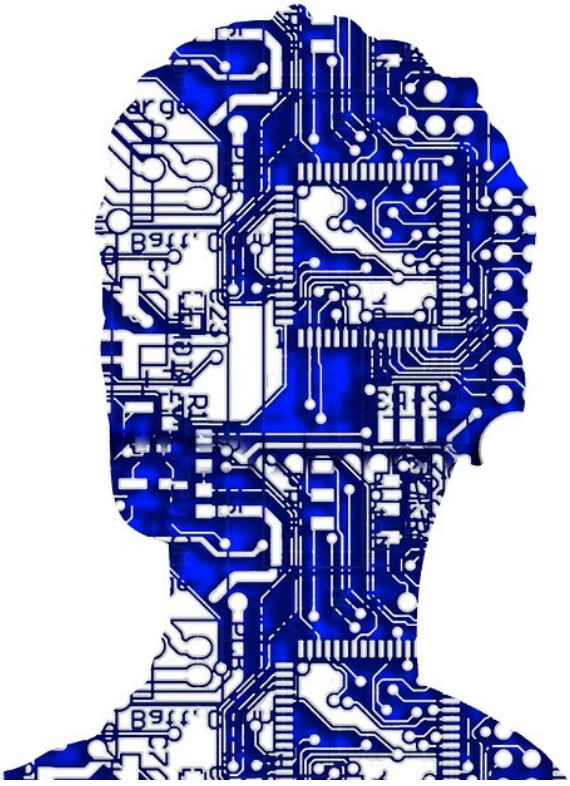
A Level Computer Science
Component 01

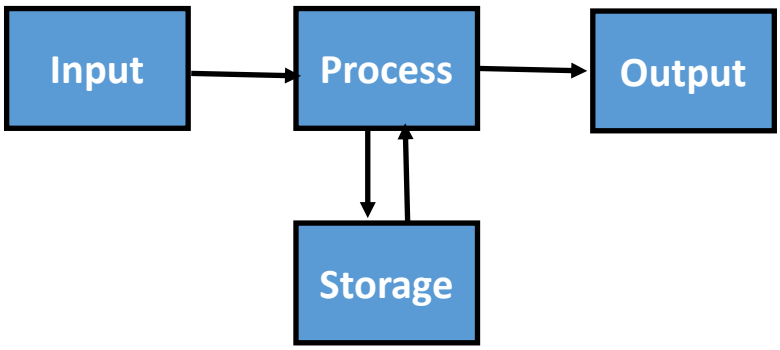


Revision

learncomputing

Hardware

Computer Systems all feature these four components



Input Devices : Allow a user to take data from the real world and input it into the computer system.

Storage Devices : Allow data to be saved permanently for retrieval at a later date.

Output Devices : Display the results of processing data back to the user.

RAM	ROM
Volatile	Embedded in the motherboard and isn't normally removed.
Constantly interacting with the CPU	No need to refresh the data as it is permanent
Holds data while it is being used by the CPU	Very small capacity
Can be easily removed from the computer and upgraded	Only interacts with the CPU under certain conditions
Needs constant refreshing to stay active	Non-volatile
In modern computers the size is measured in GB	Contains data that the computer needs to boot up

Technology	How it works	Examples
Magnetic Storage	Uses a read/write head to form magnetic particles on a disk. Either magnetising a segment or not magnetising a segment refers to a binary 0 or 1.	Hard disk drive, Floppy drive, tape drive
Optical Storage	Uses a laser to burn indentations into a reflective disc. 0s and 1s can be produced by either burning an indentation or not.	CD-ROM, DVD-ROM, Blu-Ray ROM
Flash Storage	"Flashes" electricity through a chip causing logic gates to either open or close. An open gate refers to a 1, a closed gate is a 0.	Solid state drive, SD memory card, USB memory stick

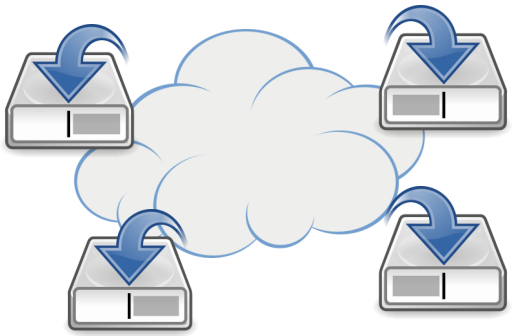
Virtual Storage

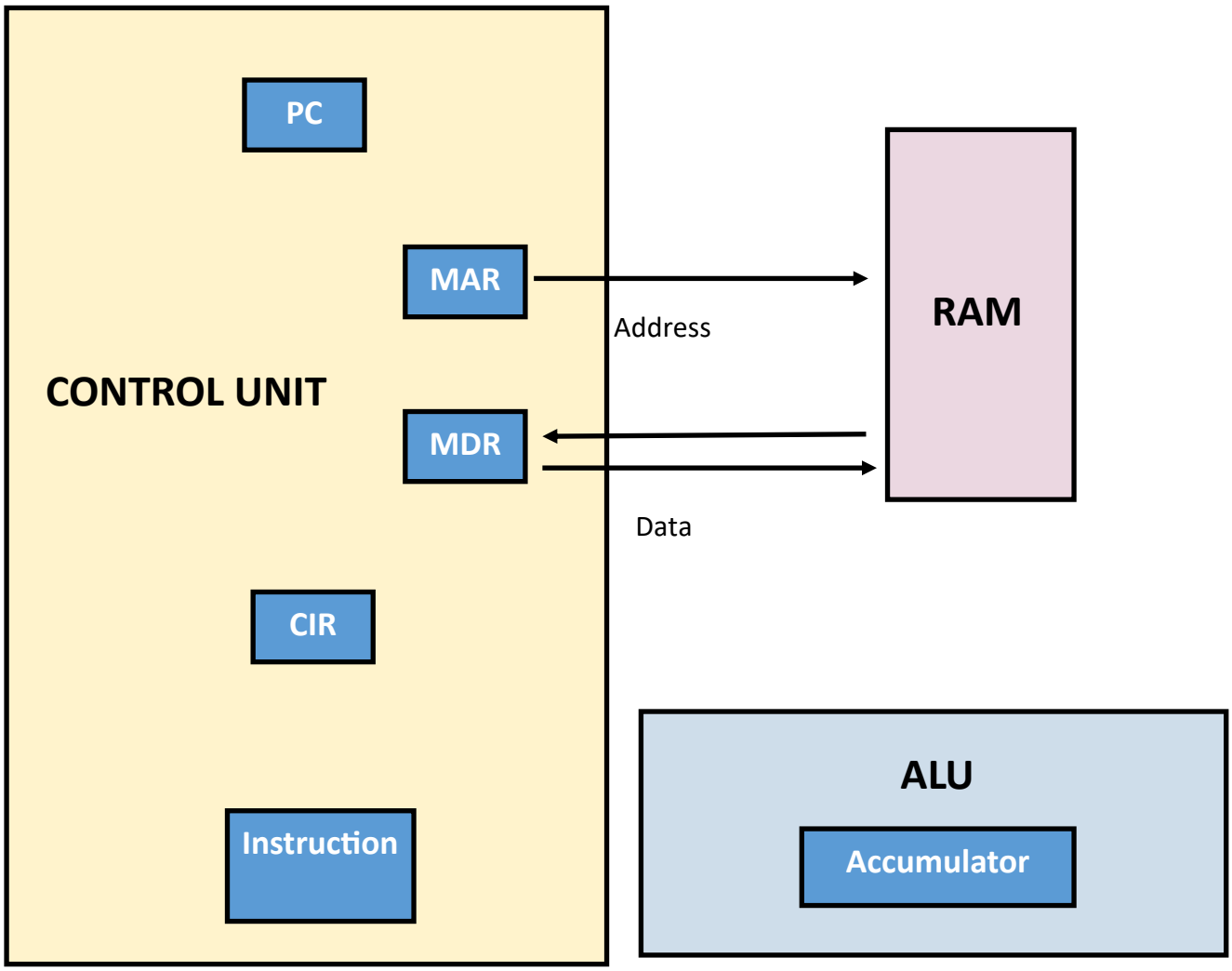
An operating system can give a user the illusion of having a storage drive.

This means that segments of physical drives can be combined together under one drive letter and the user won't see any difference. They will operate as if they are just saving to a drive like they normally would but in fact the files can be spread across any number of drives and locations.

It can also be used to create a partition in a physical drive so that it appears to the user there are in fact two drives.

- Useful for organising network resources e.g. providing users with a place to store their documents but not allowing saving to important sections of the network
- Useful in cloud computing where files can be spread across a range of servers but can provide simplicity to the user





The Fetch, Decode, Execute Cycle

1. The program counter copies its' contents into the Memory Address Register. This indicates the next memory location to look for the next instruction.
2. The MAR sends the memory location along the address bus to the main memory.
3. The PC is incremented by 1 so that it is pointing to the next instruction.
4. The instruction and data at the memory location returns along the data bus and is stored in the Memory Data Register (aka Memory Buffer Register).
5. The value is then split up into its opcode and operand and stored in the Current Instruction Register.
6. The opcode is decoded using the instruction decoder. This uses the instruction set for the CPU to work out what to do with the operand.
7. The instruction is then executed. The resulting value is placed in the accumulator.
8. If there is a part of the program that needs the code to go down another path the CIR changes the value in the PC so that a new memory location is used next. This is called a jump instruction.
9. If the instruction needs to be stored the accumulator sends the data and a new memory location back to the MDR and the MDR sends both the memory location and the data across the data bus so that the value can be stored in main memory.

Control Unit

Manages the process of fetching and executing data/ instruction sets from memory.

ALU

Carries out calculations on data. Add, multiply, divide, subtract, greater than, less than, equal to

Clock Speed

The number of cycles per second. Measured in Hz. 2GHz is 2 billion cycles per second.

Cache

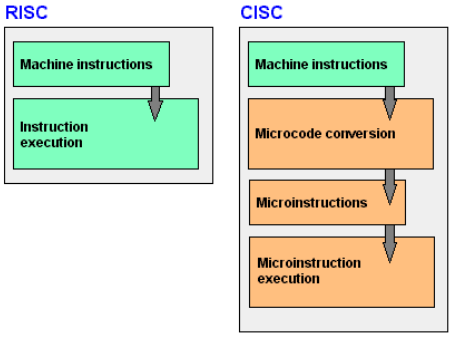
Super fast memory kept on the processor. The most commonly used instructions are stored here to speed up retrieval time.

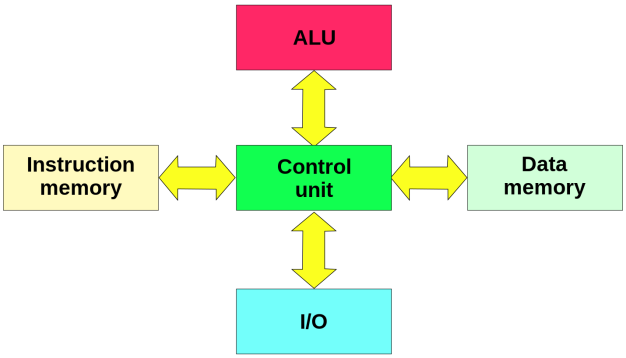
CISC (Complex Instruction Set Computer)

- Gives the programmer many complex and useful instructions to work with
- There are lots of instructions available and lots of ways to address memory
- Single register set
- There is less storing of intermediate results so less RAM is used
- The hardware is more complex
- Complex instructions may take multiple cycles to complete
- Software generally runs more slowly due to the more complex hardware

RISC (Reduced Instruction Set Computer)

- Very fast execution of simple instructions
- Each instruction takes only a single cycle
- Only a few instructions available to program
- Each instruction can only perform a very simple task e.g. ADD, LOAD
- Each full job can take many cycles to process





Harvard Architecture

Separate data and instruction buses

Can read an instruction and access memory at the same time

Von Neumann Bottleneck

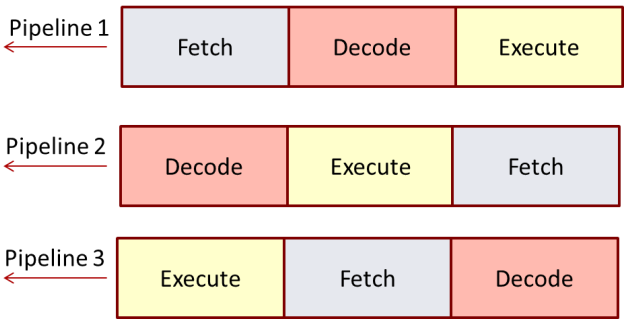
The time it takes to send data and instructions along the address and data bus is unavoidable. This causes latency.

CPU Architecture

Single Instruction Single Data (SISD)

A type of processing where instructions are brought in one at a time and each instruction matches up with one piece of data.

This can be made more efficient using **pipelining**



Fetching an instruction (pipeline 1) occupies the address and data buses but whilst that is happening an instruction can be decoded in the control unit and an instruction can be executed elsewhere.

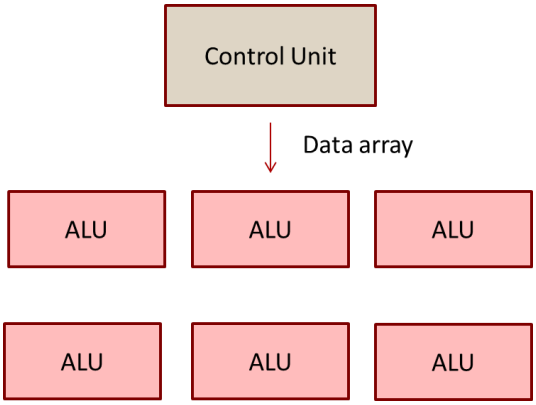
Pipelining makes a more efficient use of CPU time.

However pipelined CPUs are more expensive to make. Also sometimes certain instructions need to be run first as their outcome is needed for the next instruction to be run. On such occasions pipelining breaks down and its benefit is lost.

Single Instruction Multiple Data (SIMD)

A type of processor known as an **array** or **vector** processor. A single instruction is carried out on multiple sets of data i.e. there is one control unit and multiple ALUs.

This is useful for large scale calculations such as predicting weather patterns or working out the slipstream of a jet in a wind tunnel.

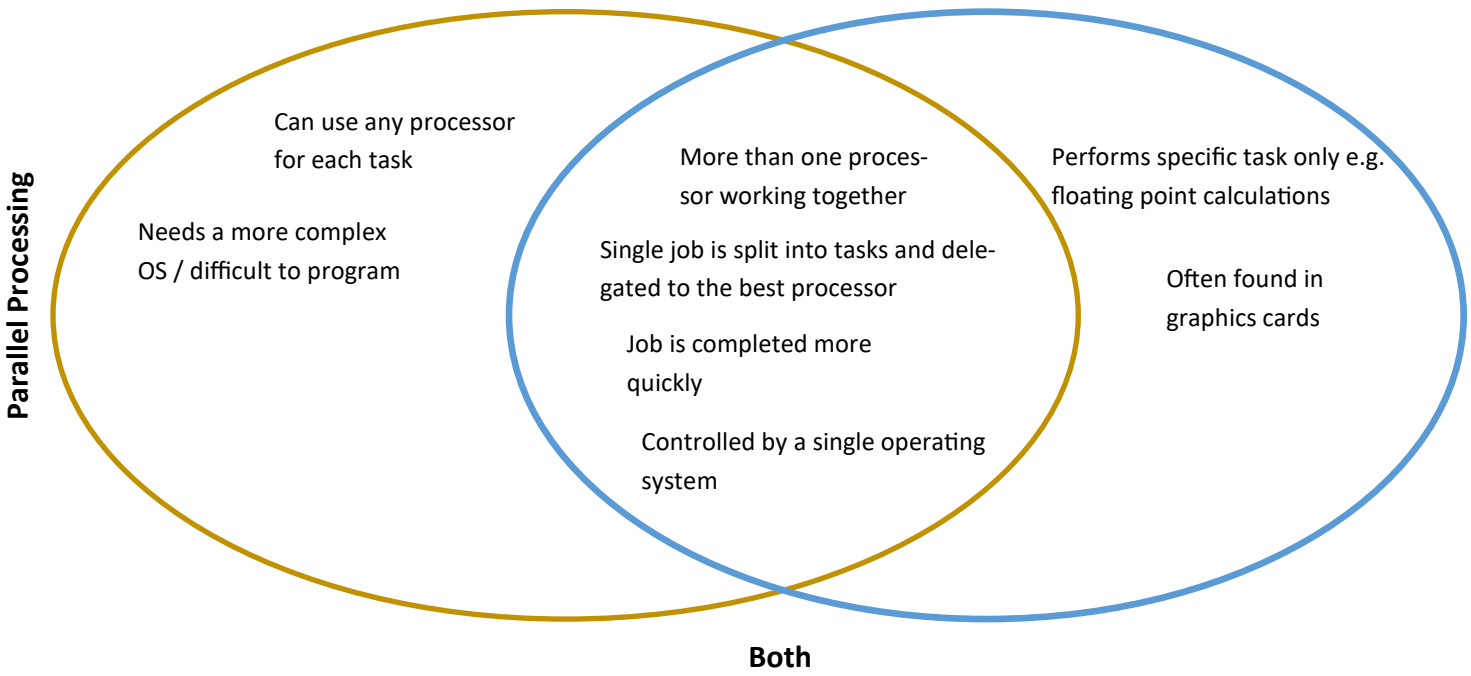


SIMD allows the same instruction to be carried out on multiple sets of data. This can be very fast.

However you are limited to one instruction at a time.

It also suffers the same drawback as pipelining - if an instruction relies on the outcome of a previous instruction it needs to wait until that instruction resolves before proceeding.

Parallel Processing vs Co-Processor

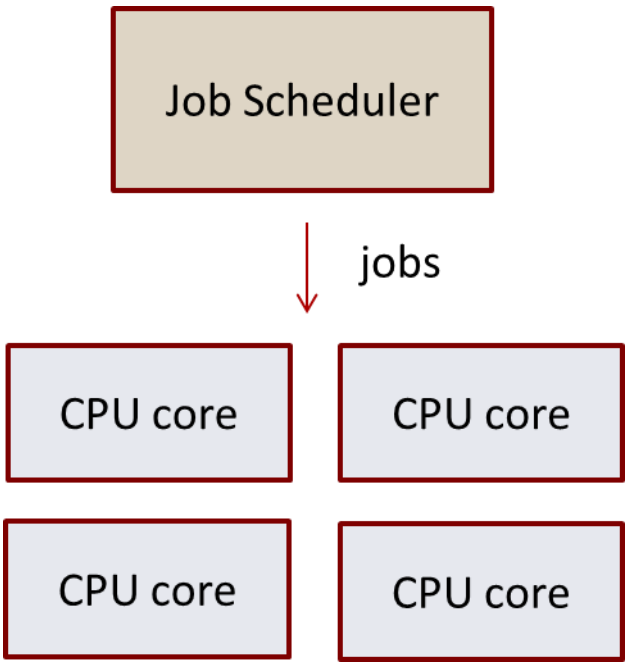


Multiple Instruction Multiple Data (MIMD)

This type of processing involves having multiple control units that act on different data at once.

A job is sent to the processor, it is split up into different tasks and can be delegated to the best place to be processed.

There are multiple CPU cores i.e. multiple control units and multiple ALUs.



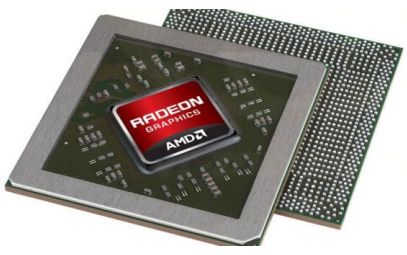
Instructions can be processed in parallel and the Von Neumann bottleneck does not apply.

However it needs a more complicated OS and it is more difficult to program. It is more expensive and many existing programs aren't build to take advantage of the architecture.

Co-Processor

Parallel processing is used in a multi-core CPU.

A co-processor is a separate processor used to support some complex function such as calculating floating point number calculations.



GPUs

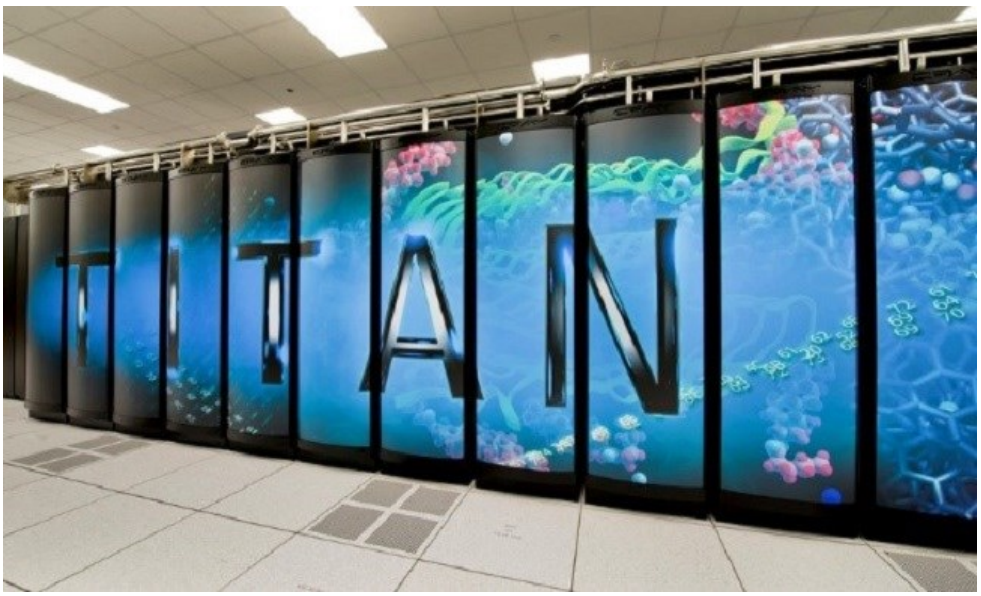
GPUs are processors that specialise in processing vast data sets.

They use Single Instruction Multiple Data architecture.

This is great for processing graphics and this is the use for which they are commonly known. GPU = Graphic Processing Unit.

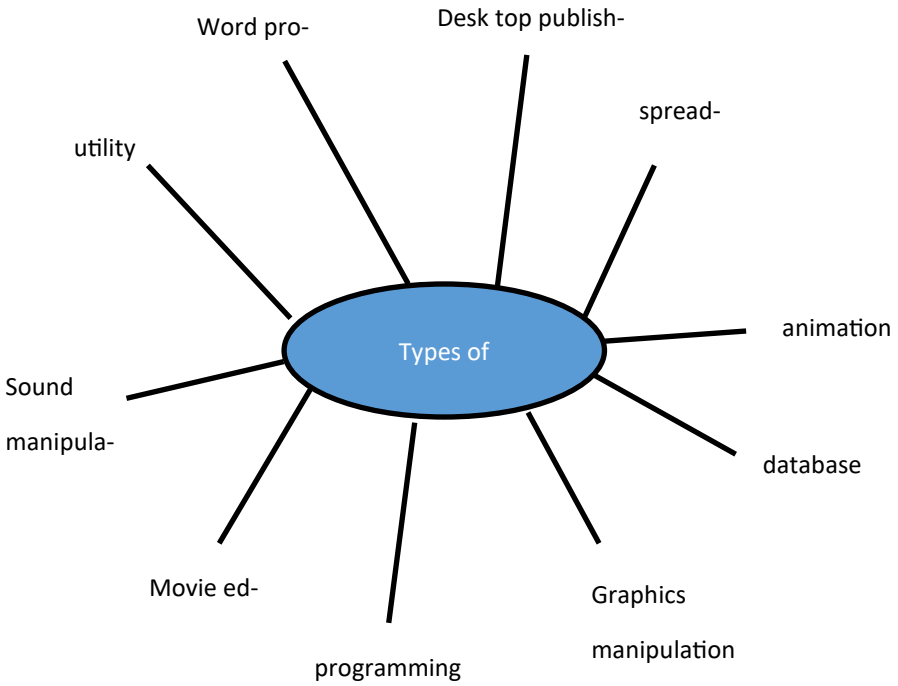
However, their use is not limited to graphics. Any application that requires large data sets to be processed can make use of the fast speeds a GPU can generate. An exam question might ask you for an example—predicting weather cycles is a good one or calculating the wind resistance of a car.

GPUs are very common in supercomputers. Most supercomputers will contain thousands of CPUs and also thousands of GPUs to work alongside them. Companies can book out a supercomputer for a period of time when they need vast data processing to be completed.



The Titan supercomputer has over 250,000 GPUs inside.

Software Basics



And many more!!!

Utility Software

Computer Security

Anti Virus

- ◊ Attempts to protect your computer from malware
- ◊ Stays resident in memory and scans memory for suspicious activity
- ◊ Regularly scans the hard drive and compares results against a known malware database

Spyware Protection

- ⇒ Protects against spyware such as key loggers
- ⇒ Like anti virus it stays resident in memory and scans all processes
- ⇒ Protects against adware that attempts to sell your browsing habits to companies

Firewall

- * Acts as a barrier between your network and the internet
- * Incoming traffic is scanned and the results compared against a database of known threats

System Maintenance

System Information

- ⇒ Gives you detailed information about your system
- ⇒ Information could include CPU manufacturer / clock speed, device driver versions, amount of RAM and many more.

System clean up

- * Scans your hard drive for files that could be unnecessary and deletes them to free up space
- * Files could include—temporary files created by programs, cookies, old emails, unused registry entries
- * Can be set to run on a regular timescale e.g. daily / weekly

Automatic updating

- ◊ Automatically and regularly logs into a software manufacturer's website to see if there is an available update. If there is it will download and install it.

Disk Organisation

Defragmenter

- ◊ Scans your hard drive and looks for files that have become fragmented / scattered across the disk
- ◊ Attempts to put all fragmented files together so that the disk read head has less time to travel thus increasing speed and freeing up space

File Transfer

- ⇒ Used to transfer files across the internet using ftp (file transfer protocol)
- ⇒ Stores the username / password for a web server and allows you to connect remotely

Formatter

- * Prepares a disk for use with an operating system
- * Writes the File Allocation Table and defines the way files are stored
- * The process also erases all files on a disk so be careful!

Open Source Software

Software where the source code is given away when distributed.

Usually free (licenced under GNU or MIT license).

Means you can alter the code yourself if you have the knowledge to do so.

Software can be developed collaboratively by many developers across the world.

A wide range of software is available from word processors to graphics to operating systems.

How closed source software can compete

Quality—paid for software companies can afford to get the best developers to make the highest quality software

Best testing—one development company can fully test all components

Support—help files, documentation, online support, telephone support, training

Industry standard—some software is industry standard e.g. Word, Photoshop and is so well used many won't accept anything else.

Device Driver

Acts as a layer between a piece of hardware and the operating system.

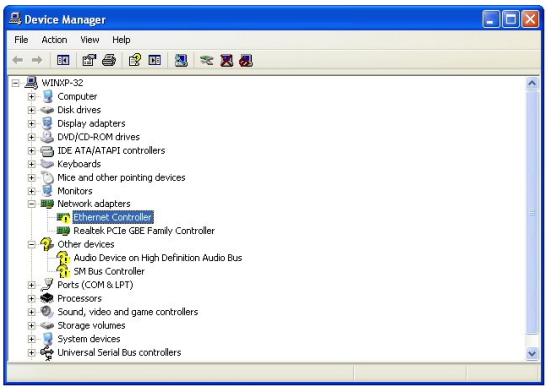
Could be made by the manufacturer of the OS for common parts but is more often made by the hardware manufacturer.

The manufacturers need to make a driver for each piece of hardware they make and for each operating system they want it to work on.

In Window the Device Manager can be used to see the current version.

Drivers can be updated to the latest version by accessing the manufacturer's web site. Often the ability to "roll back" to a previous version is allowed in case the update causes problems.

Drivers are usually written in a low level language (machine code or assembly) or a powerful high level language like C. This is because the driver is extremely machine specific.



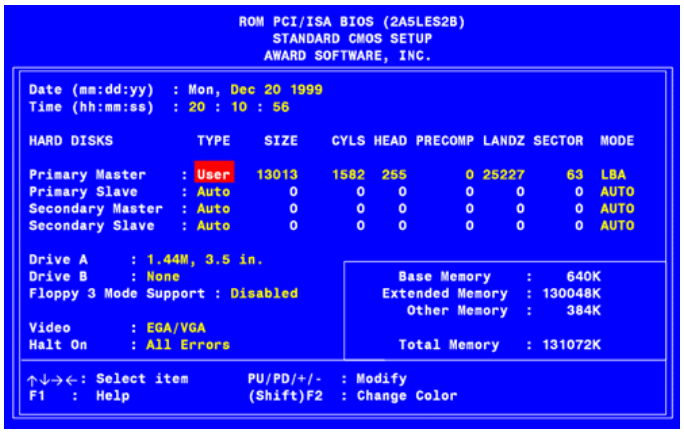
BIOS, CMOS and POST

When the CPU is first switched on a jump instruction is triggered that points to the first address of the BIOS—this is stored in ROM not RAM. The BIOS contains information to allow keyboard input and screen display.

The BIOS then starts the Power On self Test (POST) that runs through a series of checks to make sure the hardware is operating ok.

The BIOS then looks for an Operating System. Normally it is set to check the hard drive (but it can be set to check other drives). The BIOS looks in the MBR (Master Boot Record) of the hard drive for the OS and if found the CPU loads the OS by executing the code in the boot sector. This is called the **boot loader** or **bootstrap loader** and loads the OS into main memory.

The CMOS setup lets you change the time and date and settings for how devices are loaded at start up, like hard drives, disc drives, and floppy drives. The CMOS setup lets you enable and disable various hardware devices, including USB ports, the onboard video card and sound card (if present), parallel and serial ports, and other devices.



Types of Code

HIGH LEVEL CODE

Closer to natural language than assembly

Needs to be translated before it can be run.

Two types of translator for high level code:

- Compiler
- Interpreter

```
Function GetList() As XElement
    Dim c As Category = CType(ViewData.Model, Category)
    Return <ul><%= From product In c.Products _
                   Select _
                   <li id=<%= "prod" & product.ProductID %>>
                       <span class="editlink">
                           <a href=<%= "/Products/Edit/" & product.ProductID %>>
                               <%= product.ProductName %>
                           </a>
                       </span>
                   </li> %>
           </ul>
End Function
```

ASSEMBLY LANGUAGE

A low level language that is related to the computer it is being programmed for

Machine specific

Uses descriptive names for data stores

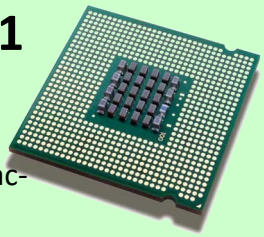
Uses mnemonics for instructions

Uses labels to allow selection

Each instruction is usually translated into one corresponding machine code instructions

```
01 DATA SEGMENT
02       MESSAGE DB "HELLO WORLD!!!$"
03 ENDS
04
05 CODE SEGMENT
06     ASSUME DS:DATA CS:CODE
07 START:
08       MOV AX,DATA
09       MOV DS,AX
10       LEA DX,MESSAGE
11       MOV AH,9
12       INT 21H
13       MOV AH,4CH
14       INT 21H
15 ENDS
16 END START
17
```

MACHINE CODE—01010100101110010010001001



Machine code is in binary notation

It includes all the available instructions that have been designed by the CPU manufacturer

Each CPU has a different instruction set depending on how the manufacturer designed it

Instructions operate on bytes of data

SOURCE CODE vs OBJECT CODE

Source code is the original code that a programmer writes

It is translated into object code

Object code can either be:

1. Machine code—to be run directly by the CPU
2. Intermediate code—for use in a virtual machine. Intermediate code will then have to be compiled again by the VM. This is slower but allows code to be ported to multiple devices. E.g. Java

COMPILER

Translates the whole code as a unit

Creates an executable program or intermediate program

May report a number of errors all at once

Code is optimised

INTERPRETER

Translates one line / statement at a time

Allows each line to be run before the next line executes

Reports one error at a time

Stop when it encounters an error

ASSEMBLER

Converts assembly language into machine code

Replaces mnemonics with corresponding machine code opcode

It replaces symbolic addresses / address identifiers with numerical addresses

It checks the symbol table to match labels to addresses

It reserves storage in the machine for the instructions and data that are produced

Checks syntax and provides error reporting

VIRTUAL MACHINES

Used by languages like Java. Code is compiled into intermediate code and can then be interpreted by the virtual machine. The advantage is that the Java VM works on any device making your programs fully portable.

Translators

Stages of Compilation

HIGH LEVEL CODE is sent to the compiler

```
Function GetList() As XElement
    Dim c As Category = CType(ViewData.Model, Category)
    Return <ul><%= From product In c.Products _
                   Select _
                   <li id=<%= "prod" & product.ProductID %>>
                       <span class="editlink">
                           <a href=<%= "/Products/Edit/" & product.ProductID %>>
                               <%= product.ProductName %>
                           </a>
                       </span>
                   </li> %>
           </ul>
End Function
```

STAGE ONE—LEXICAL ANALYSIS

```
Function GetList() As XElement
    Dim c As Category = CType(ViewData.Model, Category)
```

Keyword Lookup Table

Keyword	Token
Function	101101101
Dim	11001100
CType	11011000

Symbol Table

Name	Data Type	Memory Location
myVariable	integer	011010110
nameOfCow	string	011010010

LEXICAL ANALYSIS:

The source code is inputted into the compiler—the first stage of compilation is lexical analysis.

The compiler passes over / scans the code one piece at a time. This is called a **parse**.

Any language keywords are identified and converted into a **token** by looking them up in a table.

A token is a fixed length binary string.

Variables are identified as well and stored in another table called a symbol table.

Operations are identified as well.

Any spaces are taken out.

Any comments are taken out.

Some error diagnostics may be given.

The code is prepared and outputted for the next stage of compilation – syntax analysis.

STAGE TWO—SYNTAX ANALYSIS

Takes the output from lexical analysis and checks the syntax

Statements and expressions are checked to ensure they follow the rules of the language

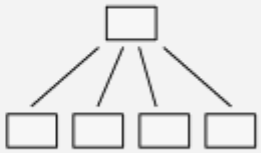
If there are errors a report is generated that is given at the end of the parse

Error diagnostics may be given

Further details may be added to the symbol table e.g. data types, scopes, memory addresses

STAGE THREE—CODE GENERATION

During the code generation stage the output from the syntax analysis stage is taken and turned into low-level or intermediate code so that is can be run.



Each high-level language statement might end up becoming dozens of low level statements due to the relative nature of programming at each level. The final program will

During the code generation stage code is **optimised**.

This means it is simplified to its most simple form possible.

The advantages of this are :

- Shorter program – takes up less storage space and memory space
- Faster execution

LIBRARIES

Libraries are externally written code chunks that you can import into your project.

Static linking – puts all the library code and compiled code into one big executable file. This might mean that library files end up being repeated if two installed programs need the same libraries.

Dynamic linking – keeps the library separate. The OS loads the library when it is needed using a loader program. This is more efficient but if you accidentally delete the library your programs won't run. Windows saves dynamic link libraries in .dll files.

Functions of the Operating System

Provides and manages hardware resources such as memory and job management

See sections on scheduling and memory management.

Provide a user interface

CLI—Command Line Interface

Very fast and powerful but not suited to beginner users as the commands can be daunting and difficult to learn.

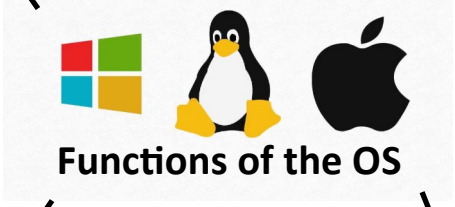
GUI—Graphical User Interface

Accessible by most users. Simple to learn. Not as fast and powerful as CLI.

Provide security and manage users

Firewall and anti-virus are common bundled applications.

User access levels, accounts and passwords, administrator privileges.



Provide an interface between software applications and the machine

Devices drivers—software that interfaces between the OS and the hardware. Many device drivers are pre-installed but often manufacturers create their own to ensure their device is optimised.

Provide utility software to allow maintenance to be done

See utility software section

Virtual Memory

The CPU can only process data and instructions stored in RAM.

If there is not enough physical memory to store all the required processes a section of the hard disk can be used.

Pages / segments need to be swapped to and from the hard drive when needed.

The process of swapping the files is very slow and takes a lot of CPU time. See problem of disk threshing on the back of this sheet.

Scheduling

This is used to:

- Maximise the number of jobs processed in a specific time (process jobs as quickly as possible).
- Obtain efficient use of processor time depending on job priorities.
- Ensure all jobs get a fair share of processor time and one job doesn't monopolise it.
- Maximise the number of users in a network system so the system runs without apparent delay.

Scheduling Algorithms

FCFS – ***First Come First Served***. Jobs are completed in the order they arrive.

RR – ***Round Robin***. Jobs are split up into time slices (normally just fractions of a second). Once a CPU has processed a time slice the job moves to the back of the queue. This continues for all jobs until all complete.

SJF – ***Shortest Job First***. When new jobs are added they are placed in the queue according to their length.

SRT – ***Shortest Remaining Time***. As processing goes on jobs get shorter. Similar to SJF but is a pre-emptive version – i.e. if a shorter job is added then the current job gets interrupted.

MFQ – ***Multi-Level Feedback Queues***. Complex algorithms that set up a league table system and jobs rated according to time left, resources required and importance.

Interrupts

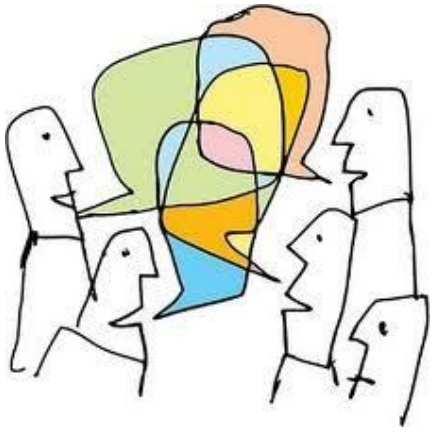
- An interrupts is a signal sent to the CPU to attempt to obtain processor time. It is stored in the **interrupt register**.
- The interrupt register is checked every time the fetch, decode, execute cycle is finished.
- The priority of the current task is compared to the priority of the task in the interrupt register.
- This is so higher priority tasks can be done first, delays are avoided and data loss is avoided.
- Interrupts are often triggered by devices indicating they need to be serviced.
- Each interrupt triggers the execution of an Interrupt Service Routine (ISR). This is a set of commands that are designed to handle the interrupt. Once a cycle is finished, the contents of the registers are copied to stack memory. The new request is loaded into the CPU, executed then the previous task is copied back in from the stack.

Sources of interrupt

Includes (but not limited to):

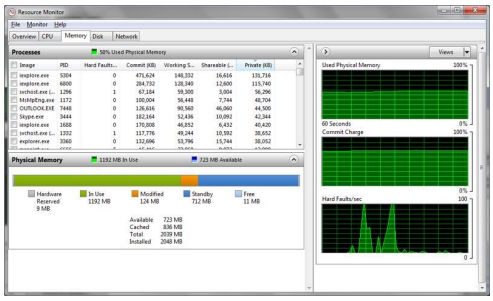
- Imminent power failure (HIGH priority!)
- System clock interrupt
- User interrupt e.g. new log on request, mouse, keyboard
- Software interrupt caused by a process in an application
- Peripheral—e.g. printer our of paper (LOW priority)

PRIORITY



Memory Management

- Allocates memory to allow separate processes to run at the same time.
- Protects processes / data form each other i.e. stops memory being overwritten / data loss.
- Protects the OS and provides security from malware.
- Enables memory to be shared.
- Reallocates memory where necessary.
- Allows paging and segmentation (see below).



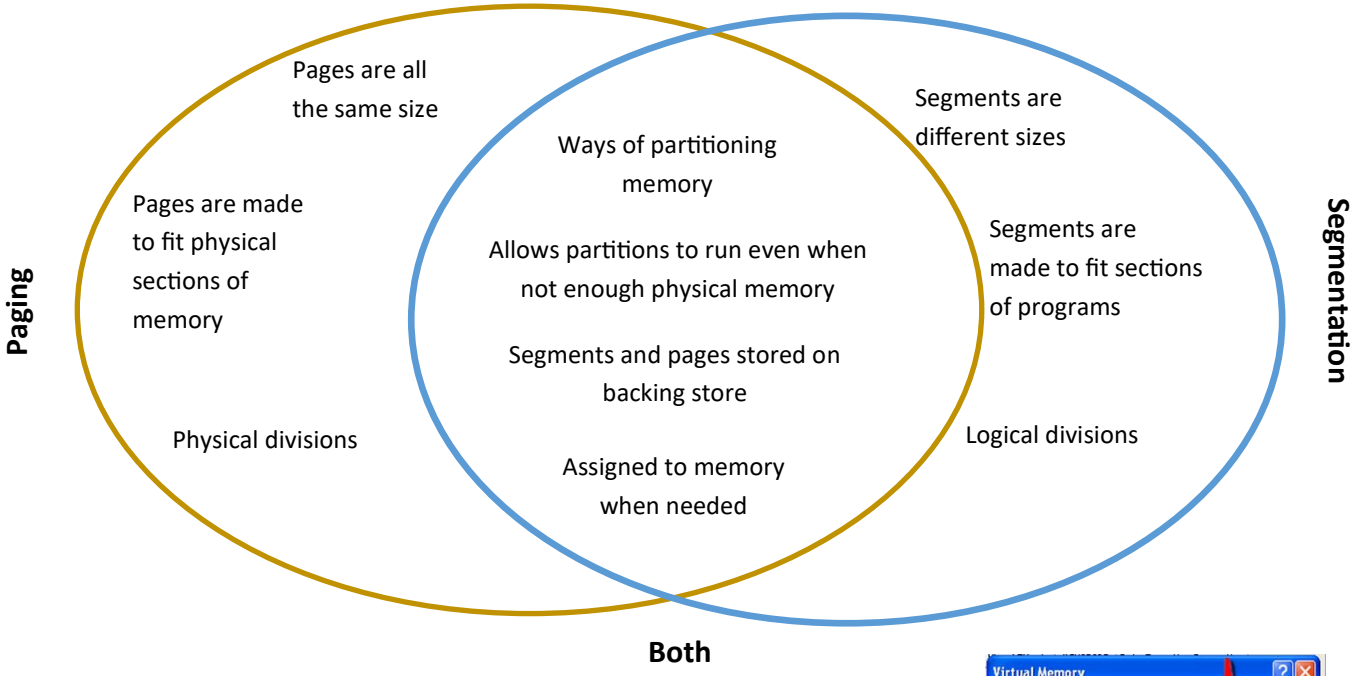
Paging and Segmentation

So that memory space can be maximised it is partitioned into small chunks. There are two methods called paging and segmentation.

A large file on the hard drive is created and partitions are stored there and swapped into memory as needed. Remember a partition needs to be in main memory before it can be run.

Pages are simply the total of main memory split into small, equal sized chunks. Sizes vary but a typically size is 4k.

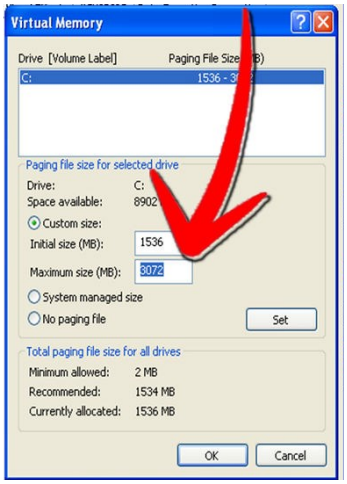
Segments are dynamically sized and will fit the job, task or function that needs to fit into memory. This is more complex to code.



Disk Thrashing

A problem of using paging and segmentation is disk thrashing.

This is where the main memory isn't big enough to fit in all currently active processes and the CPU has to constantly swap files to and from the backing store. When this happens more time is spent swapping files than actually executing the instructions and the system can hang or crash.



Types of Operating System

Single User

- Allows one user at a time to use the system
- Each user can be allocated with rights
- The user's files and preferences are separate
- Windows 10 and PlayStation 4 OS are examples

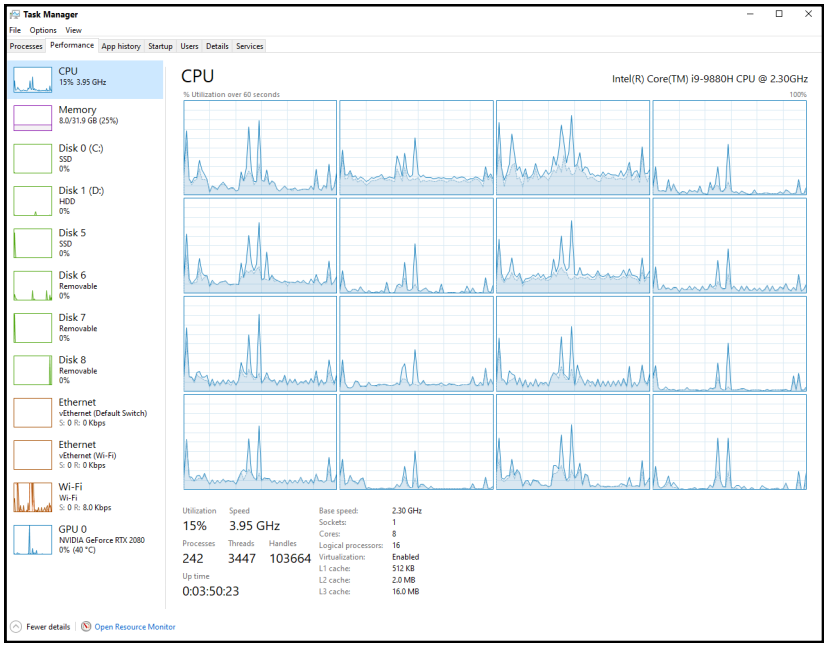
Multi User

- Allows multiple users to use a system at once. It is therefore suitable for networks.
- Each user can have rights and privileges.
- The accounts are controlled by an administrator like a network manager.
- Windows NT is an example.

MICROSOFT WINDOWSNT

Multi Tasking

- Most modern operating systems are multi-tasking.
- This type of OS gives the illusion of performing multiple jobs at the same time e.g. having a range of programs open at once.
- In actuality the tasks are processed one at a time and all jobs are scheduled using one of a variety of scheduling algorithms.



Real Time

- A real time OS is one which **guarantees a response within a certain timeframe.**
- This is useful for sensitive equipment like medical equipment, robotics plants and heavy machinery.

Embedded OS

- An embedded OS is a specialized OS for use in computers built into larger systems.
- Examples include computers in cars, traffic lights, digital televisions, ATMs, airplane controls, point of sale (POS) terminals, digital cameras, GPS navigation systems, elevators, digital media receivers and smart meters, among many other possibilities.
- The operating system will be largely simplified to streamline the processes the those few required by the hardware.

Distributed OS

- A distributed operating system is a software over a collection of independent, networked, computers.
- Each processor in the computers contributes computing power to the overall OS.

HTML

You need to memorise the following tags. Any others will be explained in the question.

HTML Tag / Element	Explanation	Example
<html>	Goes at the top of the page to let the browser know it is an html file. The closing tag ends the file.	<html> <!— your entire website here —> </html>
<link>	Links a CSS file so your style can be saved on another sheet. This is a good idea as the same style sheet can be used for multiple HTML pages.	<link rel="stylesheet" href="style.css" />
<title>	Puts a title in the browser window. NOT on the page.	<title>My Website Title</title>
<body>	A special tag used to show where the main content of the site is i.e. not links, style, metadata or scripts.	<body> <!— all the content for your site here </body>
<h1><h2><h3>	Heading tags. H1 is the biggest, h3 is the smallest.	My main title Subheading Really small heading
<img>	Used to import a picture	
<a>	A hyperlink to another page	Click to contact us!
	A generic division in the page. Often you give it an ID so it can be referenced in JavaScript.	Here is a div
<form>	Used to start an input form	<form onsubmit="login.php"> Enter password: <input name="password" /> <input type="submit"/> </form>
<input>	A textbox. Can be changed to be a checkbox or radio button as well if needed. Used for the submit button as well.	
	A paragraph element—used for text.	This is a paragraph of text and can be any length.
	An ordered list. This is a list with numbers.	 - Fish - Cabbage - Duck
	An unordered list. This is a bullet point list.	
	List item—each individual bullet in a list.	
<script>	Used to designate some JavaScript will be next.	<script> alert("Hi this is JavaScript!"); </script>

CSS

Most of the CSS you have to know is fairly self-explanatory but you do need to know the different ways that CSS can be selected:

Inline

This is when the CSS is written directly into the HTML code:

```
<div style="border: 1px solid black; font-size: 24px; color: #34ee34"></div>
```

In a style element

This is when the CSS is included inside an HTML <style> tag

In a separate style sheet

The CSS is stored in a separate file which is linked using a <link> element.

The # symbol means you are selecting an ID. The full-stop . character is used for a class.

```
#myDiv { color: blue }
```

```
.blueClass { color: blue }
```

CSS Property	Explanation	Example
background-color	Sets the background colour of an element. Notice "colour" is always spelled as color. You can use some predefined colour names or hex values.	
border-color	Sets the colour of a border.	

border-style	Changes the style of a border. Acceptable options are "solid", "dashed" and "dotted"	#dottedDiv {
		border-style: dotted;
		}

border-width	Sets the width of a border. Usually quoted in px (pixels) but there are other measurements (see height below)	.wideBorder {
		border-width: 3px;
		}

color	Sets the colour of the **text**.	This is a white heading
font-family	Sets the font. Often you see a couple of fonts listed—this is so that if the first choice isn't installed on the user's computer the fonts cascade down the list.	
font-size	Sets the size of the text.	
height	Sets the height of an element. Can be in pixels (px), percentage of the parent (%), relative to the viewport (vw), and a few others.	<div style-"height: 20%">
width	As height but for width.	

JavaScript

JavaScript is a programming language used to add interactivity to web pages. It is interpreted rather than being compiled. This is because JavaScript runs in a web browser on many different platforms with many different CPU architectures. An interpreted language will interpret one line of code at a time and stop if there is an error and report it.

You need to be able to follow the constructs of JavaScript code in the same way that you know any other language you have studied. However, you will not be asked questions about file handling or object oriented code involving JavaScript. You will also not be asked any question where passing parameters via value or reference is important.

Inputs

Inputs will be by reading values from an HTML form. You don't need to memorise the exact way of doing this but practicing your skills won't hurt.

Outputs

Will either be:

- Changing the inner HTML of an element.

  e.g. document.getElementById('myElement').innerHTML = "This is the output";
- Writing directly to the page.

  e.g. document.write("This is the output");
- Using an alert box:

  e.g. alert("This is the output");

LMC

You need to be able to read, follow and write programs for the Little Man Computer as a way to learn basic assembly language.

The instruction set is provided by the exam board and you need to memorise all the instructions and how they work.

This is a significant topic and the advise is that you practice your LMC code until fluent by completing all the exercises in the Learn Computing resources.

Mnemonic	Instruction	Alternatives
ADD	Add	
SUB	Subtract	
STA	Store	STO
LDA	Load	LOAD
BRA	Branch always	BR
BRZ	Branch if zero	BZ
BRP	Branch if positive	BP
INP	Input	IN, INPUT
OUT	Output	
HLT	Stop the program	COB, END
DAT	Data location	

Assembly Addressing Modes

You may be given some LMC code and be expected to work out the type of memory addressing being used.

Immediate Addressing

This is the simplest and fastest form of addressing.

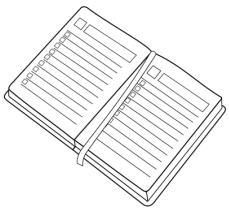
In fact – it's a bit like not having any addressing at all as it doesn't need to access memory once the operand has been called.

An example might be: ADD 45

The data in the operand is the data to be used by the operator.

This means that the 45 simply means the number 45 – it doesn't refer to any section in memory. This instruction would add 45 to the accumulator.

You could use this to add a constant to the accumulator e.g. pi.



Direct Addressing

This is a simple method – the operand refers to a memory location in which the number to be used is stored.

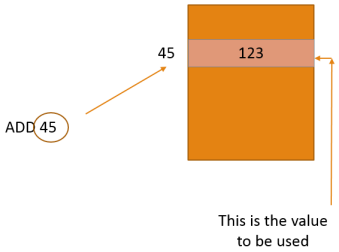
Using ADD 45 as the example again....

This time 45 refers to memory location 45. Let's say that memory location contains the number 342.

342 is added to the accumulator.

This method is fast but it depends on the same locations being available every time the program is run. This normally isn't possible as you probably want to relocate your program to different computers.

Also it means that the number of addresses is limited by the size of the address field.



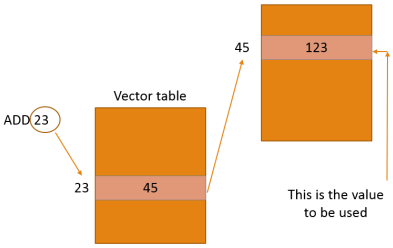
Indirect Addressing

Uses address field as a vector/pointer to address to be used.

e.g. in ADD 23 – we look first in memory address 23. Let's say stored there is the value 45. Address 45 holds the value to be used.

Used to access library routines.

Increases the size of the address that can be used allowing a wider range of memory locations to be accessed.



Relative Addressing

Allows real address to be calculated by adding a base address to the operand.

Relative address is an offset.

This is useful as we often don't know where in memory the program will be loaded. If we make all the memory locations in the program an offset from the base address it doesn't matter where the base address is.

Can be used for arrays or branching.

Indexed Addressing

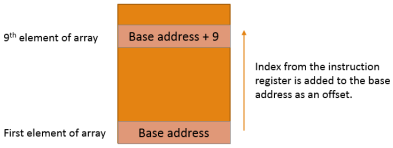
Modifies the address given by adding the number from the index register to the address in the instruction.

Often code is stored in chunks. An array is a good example.

This style of addressing needs a base address which is the start of the chunk.

An offset is then added to it. This is a number stored in a special register called the index register.

An advantage of this is if the array is relocated only the base address needs to change and any reference in your program to the index numbers can stay the same.



Symbolic Addressing

The use of characters to represent the address of a stored location.

We can reference memory using a number – e.g. 4005. However when we come to run the program there may already be another program using that address.

Symbolic addressing solves this. Instead of using an absolute location it lets you use a "symbol" instead e.g. VarA. This is exactly like using a variable in high level code.

The assembler then resolves the variable to the correct addresses during assembly by creating a symbol table.

All this means the software is relocatable in memory and it becomes much more readable.

Symbol	Data Type	Memory Location
VarA	Byte	4005
VarB	Byte	5321
VarC	Word	7894

A symbol table is created and used to place the software in memory.

The Index Register

Used in indexed addressing

Stores a number used to modify an address which is given in an instruction

Allow efficient access to a range of memory locations by incrementing the value of the IR

e.g. used to access an array

Programming Techniques

Program Flow

Sequence—instructions are executed in order inexorably from the top of the program to the bottom.

Selection / Branching—the flow of the program changes dependent on the result of a logical decision. Refers to if statements and switch statements.

Iteration—the program loops either until a certain condition is met or a specified number of times. Involves using while loops (condition controlled) and for loops (count controlled)

Variables and Constants

A variable is:

- A pointer to a storage location in main memory.
- An identifier is used to name the variable.
- It has a certain data type to define it's size.
- Can be declared at any time, assigned a value any time and can change during the program.

A constant is:

- A pointer to a storage location in main memory.
- It has an identifier and data type.
- Is declared at the start of the program and has to be assigned a value immediately.
- The value cannot change during the running of the program.

Procedures and Functions

They are both:

- Named blocks of code that perform a specific purpose.
- Are called from the main program and can be reused many times.
- Can have parameters passed into them as an input.
- Can have their own local variables declared within them.

The only difference is: **a function returns a value where a procedure does not.**

String Handling

You will be expected to be expert in handling strings. Ensure you go over your programming exercises practicing method such as getting substrings, concatenating strings, reversing strings etc.

File Handling

You will be expected to be expert in file handling. Ensure you know how to open a file in write / read / append mode, write to a file, read a line from a file, read all lines from a file into an array, loop through lines in a file, update a file and close a file.

Data Structures

You are expected to know how to declare, use and problem solve with the following data structures:

- **Array**—up to 3 dimensions.
- **Record** (in C# this is a struct)
- **List** (not the same as a linked list)
- **Tuple**

Operators

Operator	Name	Purpose
+	Addition	Adds numbers together
+	Concatenation	Puts strings together
-	Subtraction	Subtracts
/	Divide	Divides
*	Multiply	Multiplies
=	Assignment	Assigns a value to a variable
==	Comparison	Compares two values
===	Strict comparison (JavaScript)	Compares two values and checks they are the same data type
!=	Not equal to	Checks two values are not equal.
!==	Strict not equal to (JavaScript)	Checks two values are not equal or not the same data type.
<	Less than	Compares two values to see if the first is less than the second.
>	Greater than	Compares two values to see if the first is greater than the second.
<=	Less than or equal to	Compares two values to see if the first is less than or equal to the second.
>=	Greater than or equal to	Compares two values to see if the first is greater than or equal to the second.
||	Or	Performs logical OR
&&	And	Performs logical AND
%	Modulo	Gets the remainder from a division calculation
DIV	Integer division	Gets the integer part only of a division calculation

Database

A database is a persistent, organised store of data.

- **Persistent—**means that the database is stored on permanent storage like a hard drive and is not a memory-only data structure like an array, queue or list.
- **Organised—**the data isn't just a data file (like a simple text file). It is catalogued in a certain standardised way that is discussed on this sheet.
- **Store of data—**the raw facts and figures that an application needs are stored here. Databases are often fundamental to an applications design and it is rare to make an application without one.

Flat File Database

A flat file database has one table only.

It is made up of a number of fields (also called columns). Each column stores one data attribute—examples from the table below are FirstName and Surname.

At least one field must be unique. This is to ensure there are no duplicate fields. This field is called a **primary key**.

Data can then be stored—each data entry session is called a record or sometimes, more simply, a row.

Flat file databases are useful for simple systems such as storing details of customers (like the flat file database below).

Sometimes you may wish to index a field so that is can be sorted or searched quickly. If this happens this is known as a **secondary key**.

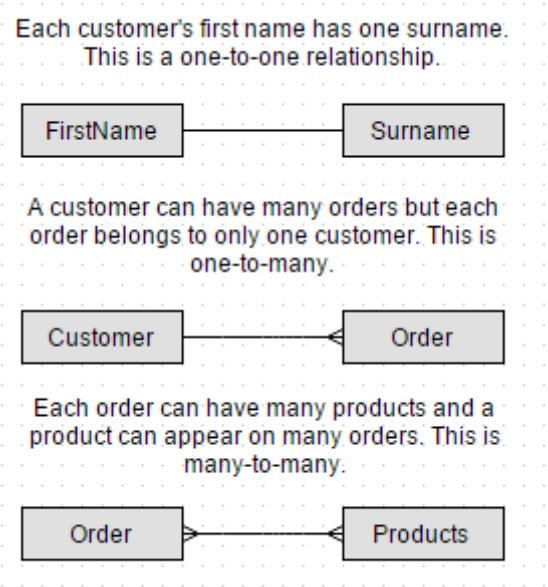
CustomerID	FirstName	Surname	Email	Password	NINumber
1	Jonny	Bairstow	jonny@ecb.com	Ginger123	JJ 50 94 84 D
2	Stuart	Broad	stewie@ecb.com	Broady123	JK 34 12 54 D
3	Moeen	Ali	fearthebeard@ecb.com	Beard	JI 65 76 45 R
4	James	Anderson	jimmy@ecb.com	jimmy123	JJ 90 55 12 A

Relational Database

A relational database has more than one table.

There are lots of advantages to having this approach:

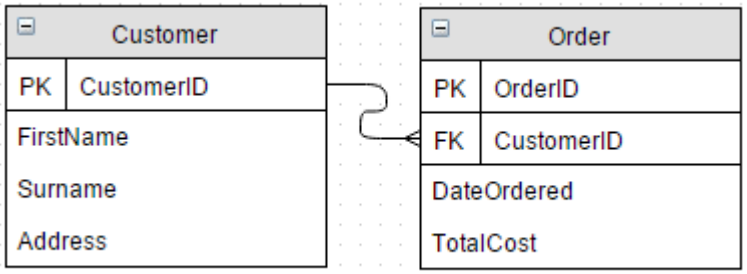
- Reduces data duplication / saves storage
- Improves data consistency
- Easier to change data or the format of data
- Easier to add data
- Improves data integrity and security
- Allows different access levels / security levels



When thinking about how the entities (tables) in your database relate just think about how they relate in real life. Remember a computer system needs to model real life and there is no technical trick to.

Foreign Key

To link tables together we put the primary key from one table inside another table. This is called a foreign key. It means that we never have to repeat data as, if we have the key to another table, we can look up anything at that table whenever we want.



Simple Database Construction

A quick and easy way to create a database is to use a top-down approach:

Abstraction / Problem Decomposition

Get the client's problem and pick out the main nouns to store data about. These become the entities in your database.

"I need a system so that my customers can log in, order some products and have them delivered."

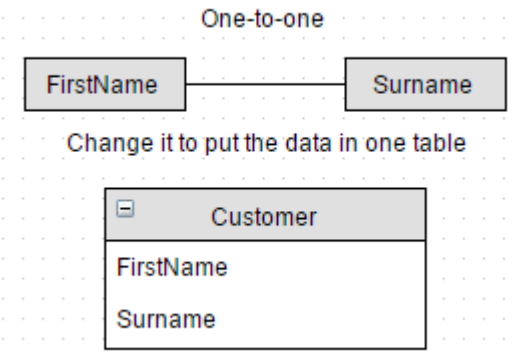
The entities here could include: Customer, Order and Product

Fix the Relationships

As a rule of thumb the only relationship that is allowed in 3rd Normal Form is one-to-many.

One-to-one relationship

If you have a one-to-one relationship then you should have included the data all in one table e.g. FirstName and Surname could be put into one general customer details table.



One-to-Many Relationship

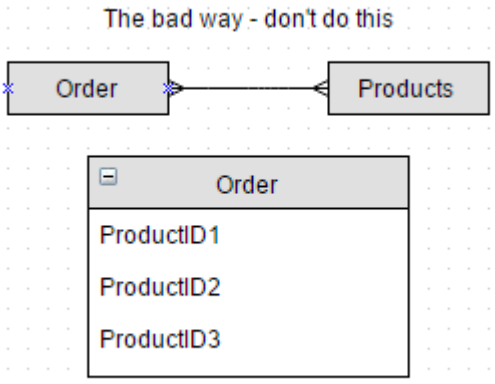
This is the only relationship that is allowed.

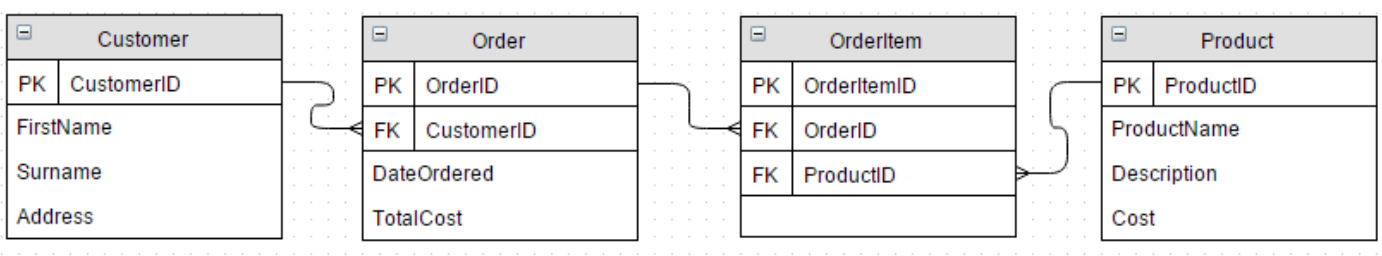
Many-to-Many Relationship

Two ways to fix this.

First the bad way is to limit data entry to a certain number of values. In this example we have a many-to-many relationship between order and products. You could link the together by allowing a set number of product IDs in the order table. This is bad don't do this.

The best way to fix a many-to-many is to put a table in the middle. This mean you can have unlimited products in an order whilst still maintaining data integrity and avoiding duplication.





Normalisation

Relational databases need to be normalised. This is to ensure that the system will operate properly, data integrity will be maintained and unnecessary duplicates are avoided.

There are different stages to normalisation—1st Normal Form (1NF), 2nd Normal Form (2NF) and 3rd Normal Form (3NF).

UNF (unormalised form)

Data appears in any format or configuration

FullName	Address	Telephone	Telephone
Jon Smith	6 Turner Drive, Whiby, YO12 3RD	0121234567	0121234567

1NF (first normal form)

- No columns have repeating / similar data.
- Each data item cannot be broken down any further (it is atomic).
- Each row has a unique field (primary key).
- Each field has a unique name.

CustomerID	FirstName	Surname	House	Address	City	Post Code	Telephone
1012324	Jon	Smith	6	Turner Dr	Whitby	YO12 3RD	0121234567

2NF (second normal form)

All non-key attributes must depend on every part of the primary key.

Venue	Artist	Attendance	Revenue	Style
Wembley	Metallica	75,000	£19mil	Heavy Metal
NEC	Adele	12,000	£4mil	Soloist

The table above is **not** in second normal form as the style does not depend on the venue.

3NF (third normal form)

No column should have transitive dependence. A column has transitive dependence when it relies on another column (other than the primary key).

e.g. a table about cars with the fields Model and Manufacturer would not be in 3rd normal form as the manufacturer can be inferred from the model name.

A table with city in as well as postcode is not in 3rd normal form as the city can be worked out by the postcode.

Often, stages of normalisation can become too complex and there is a trade off between having a fully normalised database and having one that is easy to work with.

SQL—Structured Query Language

You need to be able to work with basic SQL.

An SQL statement ends with a semi-colon (;).

The layout is unimportant so you can space your statements out in the most readable way.

You don't have to use the capital letters for key words but it is a nice way to distinguish between key words and entity names.

For row and column names case is important.

Creating a Table

```
CREATE TABLE Customer
(
    CustomerID int,
    LastName varchar,
    FirstName varchar,
    Email varchar
);
```

Inserting Data

```
INSERT INTO Customer (LastName, FirstName, Email) VALUES
("Smith", "John", "john@smith.com");
```

You can use the wild card character * to mean "everything" e.g.

SELECT * FROM Users; //gets everything from the users table.

The % character means "any letter"

Using LIKE means it will search for a specific pattern e.g.

SELECT email FROM Users WHERE email LIKE 'd%' returns all emails starting with d.

Selecting Data / Querying Data

```
SELECT Email FROM Customer WHERE CustomerID = 1;
```

Selecting Data with a Relationship

```
SELECT Customer.Email, Order.OrderDate FROM Customer WHERE CustomerID = 1
JOIN Order
ON
Customer.CustomerID = Order.CustomerID;
```

SQL Injection is a malicious technique where a user attempts to input an SQL statement into a website input field. Entering something like

John; DROP TABLE Customer

into a name field could be catastrophic for a company.

Good web developers always ensure their inputs are sanitised and protected against this type of attack.

Updating Existing Data

```
UPDATE Customer SET Email = "john@smiths.co.uk"
WHERE CustomerID = 1;
```

Deleting Data

```
DELETE FROM Customer WHERE CustomerID = 1;
```

Dropping a table

```
DROP TABLE Customer;
```

SQL is very widely used and is an essential skill for any budding IT professional. Many programming languages (especially web server languages like PHP, Python and Perl) have pre-built library routines for connecting with databases and using SQL.

Database Key Terms

Transaction processing attempts to give a response to the user within a short time frame. It is not as time critical as a real time system. An example of transaction processing is a user logging on to a website.

All databases should be able to handle CRUD operations (Create, Read, Update and Delete). These are the results of transactions. These link to the SQL commands CREATE, SELECT, UPDATE and DELETE.

The DBMS needs to ensure that a transaction does not damage the database in any way. For example if a user decided to update their details and the database handled this by deleting their first set and adding their second set a power failure at the wrong time could mean that data was lost. The DBMS is always aiming to avoid this.

Data integrity means that data should be protected by stopping inconsistent or incomplete transactions from causing **data corruption**.

Referential integrity is an aspect of data integrity that ensures data is not lost when using relationships. A good example is cascading delete—if you delete a customer for example then all their orders are also deleted.

ACID

Atomicity – means that a transaction is "all or nothing" i.e. it is either completed in full or it is not completed at all.

Consistency – means that any transaction must move the database from one valid state to another i.e. after the transaction all defined rules are intact including any constraints and triggers.

Isolation – means that transactions occur on their own and cannot be influenced by other transactions happening at the same time i.e. if five transactions were happening at the same time the result should be the same as if the five transactions had happened one after another. Record locking is used for this.

Durability – means that once a transaction has taken place the results are stored permanently (e.g. on a hard drive). If there is a power failure after a transaction then it won't matter.

Record Locking—Part of the isolation section of ACID. It means that if one user is accessing a record or making changes to it then nobody else can access it at the same time.

Redundancy— Redundancy is a word used in IT to mean a back up or copy of something. In databases data redundancy is often seen as a negative. It usually means that your data is copied somewhere and this is wasting space. A fully normalised database aims to reduce or eliminate data redundancy.

Methods for Collecting and Transferring Data

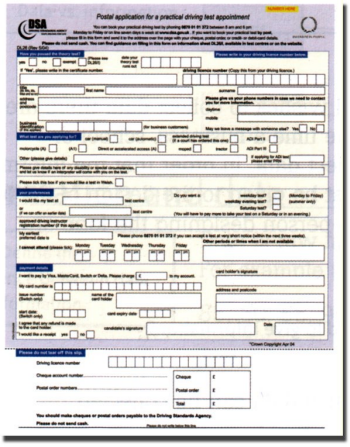
Data Capture

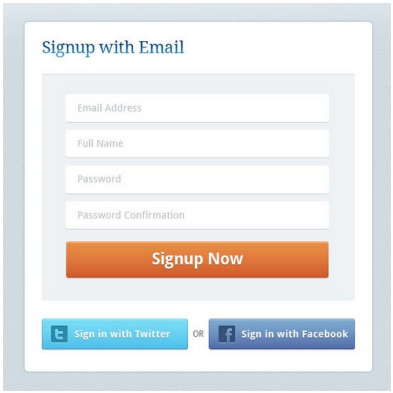
Paper forms—can then be manually typed in.

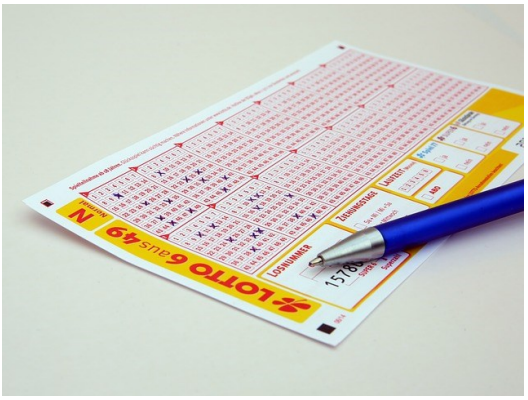
Digital forms—e.g. website sign up page

OCR—optical character recognition—attempts to read printed text and converts it to a digital string.

OMR—optical mark recognition—used it multiple choice questions and lottery tickets.







Data File Formats

CSV—Comma Separated Value. Stored in a text file the data is just in a list with a comma between each one. Often used for Excel documents.

JSON—JavaScript Object Notation. Objects are written in text like this:

{ name: “Dan Jones”; address: “20 Peel Street”; shoeSize: 10 }

JSON strings can easily be converted into an object and back in most modern languages.

Indexing a database

Indexing a database makes it quicker to search through and find the required items.

When a field is indexed it becomes a **secondary key**. E.g. if a customer table had Customer_ID as it’s primary key, then it is likely that users will want to search by surname as well. This means that the Surname field could be indexed to allow faster searching—it then becomes a secondary key.

Network Basics

A network is two or more computers (or other digital devices) connected together.

The major benefit of this is to share files and resources such as printers. Users, software installation, security and permissions can all be managed centrally.

Computers can be connected with cables of varying kinds such as coaxial, or Ethernet.

They can also be connected wirelessly using Wi-Fi, Bluetooth, microwaves or mobile phone signals.

The internet is a network of networks spread around the globe.

LAN	WAN
Local Area Network	Wide Area Network
Small geographical area	Unlimited area—could be the whole world
Cheap to set up	High set up costs
Created using Ethernet cable or WiFi	Fibre optic cables, under sea cabling, satellites
Cheaper to maintain	Much more expensive to maintain
Hub, switch	Modem, router
e.g. school network, office network	e.g. large international company, the internet

DNS

Domain Name System or sometimes Domain Name Server. The technology and infrastructure that manages global internet domains.

The process of a DNS lookup:

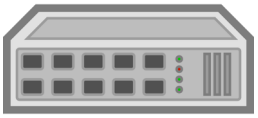
1. A user types in a domain name e.g. www.amazon.co.uk
2. First the browser will check its cache to see if the domain has been visited before.
3. If not then it must try to locate the amazon.co.uk server by performing a DNS search. First it will check the root servers, then the top-level domain servers, then the authoritative name servers until the registry for amazon.co.uk is found.
4. The DNS system will return the IP address for that domain. The browser can then create an HTTP request to get data from that IP address' web server.

DNS record type	Explanation
A	Finds the IP address for a website.
MX	Mail exchange used to find email servers.
NS	Used to state the domain registrar's name server
CNAME	Canonical name—used for domain aliases

Compression

Lossy Compression—significantly reduces the size of a file at the cost of some data loss. Suitable for images, video and sound where loss of quality is an acceptable trade off for smaller files.

Lossless Compression—uses an algorithm to reduce the size of a file but no data is lost. Useful for text compression. See later sections on dictionary and run length encoding.



Hub

Connects and sends data to multiple devices on a network

Modem

Modulator / demodulator. Turns digital data into analogue sound and vice versa so it can be transmitted down a telephone line.

Switch

Similar to a hub but acts intelligently to send the data to the individual device that needs it rather than broadcasting it out to all devices.

Ethernet Cable

Used to connect devices together in a wired network.

Router

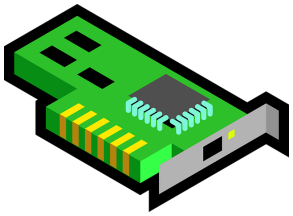
Connects two networks together. Most often used to connect the internet to your home WLAN.

Wireless Access Point

Beams out a Wi-Fi signal to allow devices to connect to the network wirelessly.

Network Interface Card

Plugs into the motherboard of a PC and allows the computer to connect to a network via Ethernet.

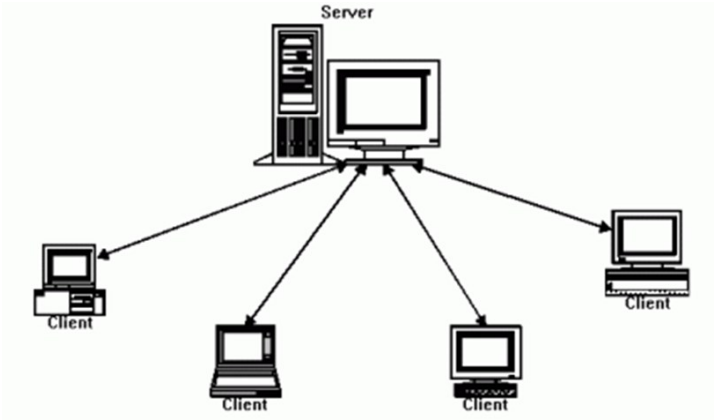


Threat	What it Is	How to Stay Safe
Hackers	A hacker is a person who attempts to gain	Use strong passwords. Keep passwords safe. Use a
Viruses	A malicious program that harms your computer and replicates itself.	Use up to date anti-malware software and a firewall. Avoid clicking on suspicious links or downloading
Unauthorised Access	The process of accessing a computer system	See hackers above.
Denial of Service	An attack on a computer system where a malicious person attempts to overload the system by	Use your firewall to block suspicious IP addresses. Configure your web server to block HTTP attacks.
Spyware	Malicious software that attempts to watch your keystrokes and gain access to passwords or	Use up to date anti-malware software and a firewall.
SQL Injection	Exploits vulnerabilities in SQL code to gain access to a database or destroy data.	Carefully sanitise any input from a user. Use prepared statements so all inputs are treated as
Phishing	An email that attempts to scam you into giving aware personal details by pretending to be from a legitimate source.	Always log onto your internet banking directly. Avoid emails that do not use your full name or with suspicious URLs.
Pharming	Directs you to a bogus website that mimics the appearance of a legitimate one, in order to obtain	As phishing.

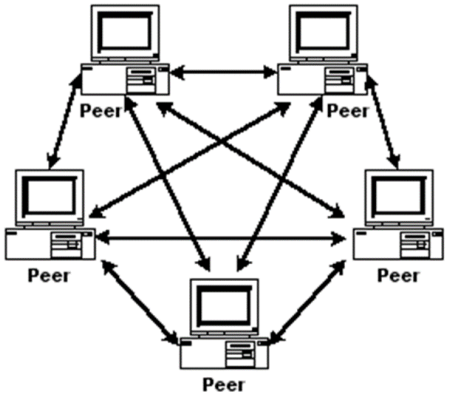
Data Transmission

A server is just another computer – often it has powerful hardware but it is essentially the same.
It is designed to serve information to other computers (clients).
There are different types:

- Web server – serves web pages to computers that connect and parses scripting languages like php and asp.
- Mail server – stores users email accounts and sends and receives emails.
- File servers – stores files in a centralised areas, controls user access via passwords and log ons.



The client server model involves having a central computer that does most of the processing. It serves data out to client computers.



In the peer to peer model there is no server and all computers have the same importance. Some file downloads like BBC's iPlayer are in a peer to peer model to reduce the potential load on the server.

HTTP Protocol

Hyper Text Transfer Protocol – used for sending web pages across the internet.

Four main methods:

- **GET method**—requests data from the server by sending a query string starting with a ? in the URL. This URL can be saved and reused as a hyperlink. This does mean that the search criteria is public.

  e.g. http://www.example.com?id=12345
- **POST method**—some data is sent to the web server. This time the data is secret. Used for submitting form data like entering email and password.
- **PUT**—similar to POST but used for updating data
- **DELETE**—does what it says on the tin

FTP Protocol

Used for transferring files across the internet. Not necessarily related to the world wide web.

The protocol is simple: you need the IP address, ftp username and password. A connection is opened, data is streamed then the connection is closed. Utilities like FileZilla perform this operation.

Standards of the Internet

As the internet is a global network of networks it makes sense to agree standard practices and ways of doing things. The protocols are examples of this.

The W3C organisation is in control of maintaining and updating the standards used. For example, HTML, CSS and JavaScript are defined in standards produced by this organisation. If everyone is using the same specification of code all around the world it makes building technology much easier.

The TCP/IP Stack

The is the protocol used for sending all data on the internet. It can wrap up other protocols like HTTP in layers. You need to know the four sections of the stack:

Layer	What is does
Application	Encodes the data being sent.
Transport	Splits the data into manageable chunks. Adds port information.
Internet	Adds IP addresses stating where the data is from and where it is going
Link	Adds MAC address information to specify which hardware device the message came from,

---

Circuit Switching

In a circuit switching system a pathway is established before data transmission. The nodes are locked off and are kept solely for the data transmission until it is complete. The data is transmitted in one long stream down the pathway from one node to another.

Three simple stages:

- Establish connection
- Transfer data
- Close connection

Packet Switching

In a packet switching system data is broken down into packets before being sent.

Each packet is marked with the destination and its source (think TCP/IP stack) and it finds it's own way to the destination. The packets are then reassembled to form the original message. This is more efficient than circuit switching.

Advantages of Circuit Switching	Advantages of Packet Switching
Simple system—easy to set up.	Packets can find the most efficient path.
One long stream can sometimes be fast if	Does not rely on only one connection path.
Does not require data to be broken up.	Packets are marked with the sender and receiver

Dictionary Coding

A lossless compression algorithm. Words / code are searched for repeated sections. These sections are placed into a table or "dictionary". Then when the repeated part occurs again we can simply refer to the dictionary to save space.

This can easily be implemented using an array.

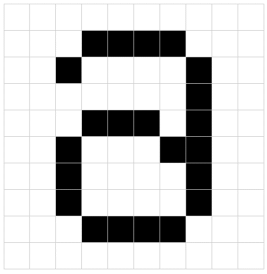
ask	not	what	you	can	do	for	your	country
0	1	2	3	4	5	6	7	8

What would this representation say?: 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 3, 9, 6, 7, 8, 4, 5

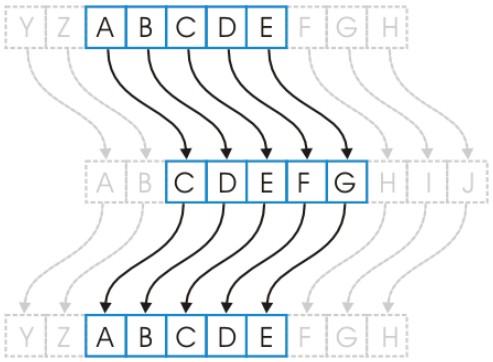
Run Length Encoding

A lossless compression algorithm used to compress images. Sections of a bitmap are scanned for repeating pixels. Whenever colours are repeated there is no reason to store the colour code multiple times. Instead we can store the colour code and the number of times in a row it appears. e.g.

10W
3W 4B 3W
2W 1B 4W 1B 2W
7W 1B 2W
3W 3B 1W 1B 2W
2W 1B 3W 2B 2W
2W 1B 4W 1B 2W
2W 1B 4W 1B 2W
3W 4B 3W
10W



Symmetric Encryption



Encryption is a technique used to scramble us a message prior to transmission. If it becomes intercepted then the message can't be read unless the reader knows the encryption key to decrypt the message.

Symmetric encryption is when there is **one** key—i.e. the same key is used to encrypt the message as is used to decrypt the message. This clearly in more secure than not encrypting but isn't really suitable for internet transactions as the key would need to be transmitted along with the message! This would defeat the point.

The Caesar cipher is a classic symmetric encryption technique. Using this method letters are displaced up the alphabet a certain number of places. e.g. with a displacement value of 3 the letter A would become D.

Asymmetric Encryption

This encryption method uses **two keys** and is widely used for transactions on the internet and for securing webpages with the HTTPS protocol. Websites can create / apply for an SSL certificate which then means data can be transmitted over HTTPS and browsers will mark the site as "secure" - often showing a padlock or similar symbol.

A public key is given out freely and users wishing to communicate with the site encrypt their data on the client machine. Data is then sent. The server then has a separate, private key, which is used to decrypt the data. The algorithm is such that only the private key can decrypt the data so it doesn't matter if the message is intercepted prior to reaching the server.

Hashing

A hash function takes an input string and scrambles it up based on a certain algorithm. This is useful for:

- Hash tables
- Generating disk addresses on a hard drive
- Storing passwords securely on a server

ID	Email	Password	Salt
1	jim.jones@gmail.com	!ER£$SD	$%£”F
2	bigstacey@yahoo.co.uk	E£r5%%6d	12£dsf
3	t.sanderson@email.com	BN*&^”D	£sdf
4	kev@ntlworld.com	FJ$!FD:L$	gFQtc!!2
5	rr@hwe.org	LK:SFEK£!	BvfGr^!

In web development passwords are hashed. The hashing algorithm is one way so that the owners of a website will never know the customer's password—only the customer themselves will know. The alternative would be very insecure and quite unethical as often customers use the same password for multiple sites.

Even a good hashing algorithm is vulnerable to brute force attacks so we can also add a salt value to increase security. This is a randomly generated string that is added to the password before hashing and stored alongside the password in the database.

Advanced Web Development

Search Engine Indexing

Search engines like Google **crawl** webpages looking for **keywords.** The programs that do this are called spiders or robots and search companies dedicate huge amounts of processing power to constantly crawling all the millions of sites on the world wide web.

Companies have their own algorithms but generally keywords on web pages can be found:

- Inside the site's meta data in the head section of the HTML.
- Inside heading tags e.g. <h1>, <h2>
- Inside image alt properties.

A website needs to have a file called robots.txt that instructs the search robots which pages to index and which to ignore.

There is a whole industry of companies who aim to improve the rankings of websites in different search engines. This is called SEO (Search Engine Optimisation). SEO companies use two methods:

- ***On-page optimisation***—the code of your site is optimised to have the most effective keywords in the best places so search engines pick them up better.
- ***Off-page optimisation***—involves boosting your ranking by making blog posts that link to your site, writing articles and getting links from other reputable sites.

Google have a set of rules that SEO companies need to follow called the Penguin algorithm. If a site is deemed to be spamming links in an attempt to boost ranking then Google ban them from their search and they are never seen again.

The PageRank Algorithm

An early algorithm for ranking pages in search engines. Made by Larry Page from Google.

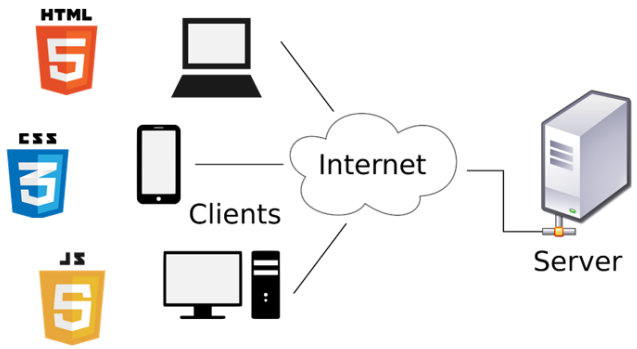
$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

The page rank of page P_i is equal to 1 minus the damping modifier divided by the number of pages in consideration plus the damping modifier multiplied by the sum of all the pages ranks divided by the number of outbound links.

The damping modifier is normally 0.85.

You should learn this off by heart and you will likely be asked questions where you have to calculate the ranking of a page.

Client Side Processing

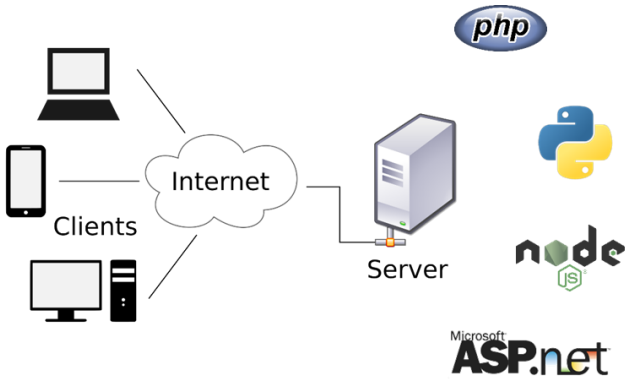


Involves the client computer processing data. The client computer is normally the computer of a user who has connected to a webpage. Client side technologies are HTML, CSS and JavaScript.

Jobs might include—drawing a webpage on screen and JavaScript interactivity (e.g. form validations).

Using JavaScript to update the screen and validate forms is fast and is good for the user experience but is insecure as it can easily be altered.

Server Side Processing



The processing is completed by the web server before data is transmitted.

Server side tasks might include accessing a database to check log in details or to find a specific resource that has been requested. Lots of websites are created dynamically i.e. they change depending on the needs of the user.

Server side languages include PHP, Perl, Python, Node.js and ASP.Net.

It is important that validations are done server side as well as client side. Server side code cannot be altered by the user so this is vital for security reasons. Server side processing should protect against threats such as SQL injection.

Character Sets

A character set is the symbols stored and used by a computer. You need to know two ASCII and Unicode although questions may ask you to apply your understanding to other sets.

ASCII—American Standard Code for Information Interchange

- Uses 7 bits to store characters. This means only 127 characters can be stored.
- Extended ASCII has an extra bit so can store up to 256 characters.
- Only English characters supported so not suitable for international use. Not enough bits to represent all the characters in the world.

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

Unicode

More modern system that attempts to have one character set for the entire world. Uses many more bits which means it can hold many more characters—useful for international languages and medical / science symbols. Three types:

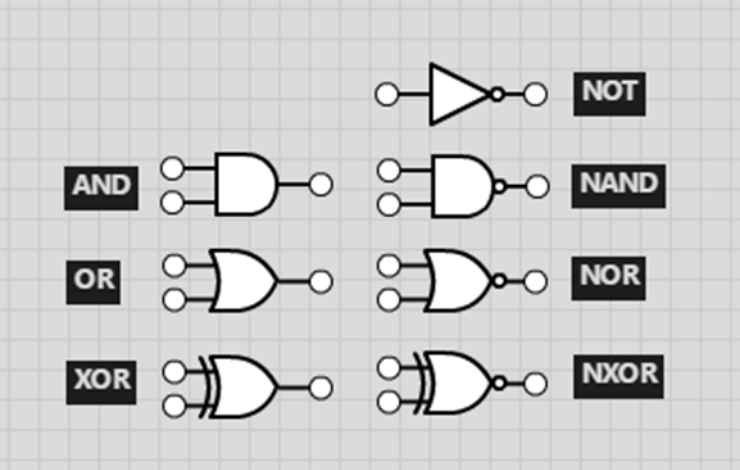
UTF 32—the most comprehensive using 32 bits to represent each character. Good because there are easily enough characters available for all used on earth commonly plus some to spare but bad because it uses a large amount of bits for each character.

UTF 16—enough bits to store the character sets of even complex languages like Chinese. The most commonly used in the far east.

UTF 8— the first 7 bits are identical to ASCII so it is backwards compatible. Not an 8-bit representation it is in fact variable length. The first bits say how many additional bytes are chained on so common symbols will use only 8 bits whereas more esoteric maths symbols will use many more. Most common encoding in the western world.

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
7	U+0000	U+007F	1	0xxxxxxx					
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx				
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx			
21	U+10000	U+1FFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx		
26	U+200000	U+3FFFFFF	5	111110xx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	
31	U+4000000	U+7FFFFFFF	6	1111110x	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx

Logic Gates



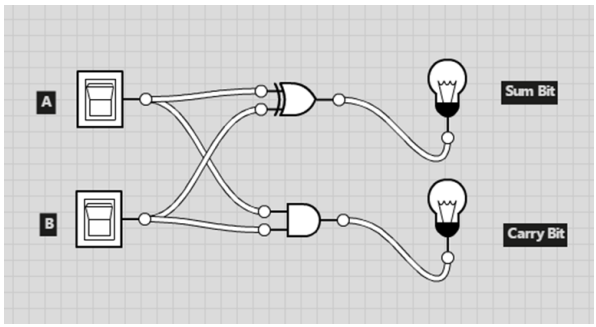
Here are the logic gates used.

Notice that any "N" gate has a little circle next to it. Look out for this to avoid mixing them up.

Gate	Symbol Used
AND	∧
OR	∨
NOT	¬
XOR	⊻

Half Adder

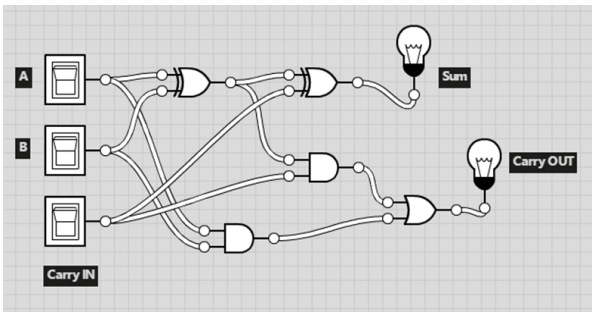
A circuit that takes two inputs and performs the equivalent of a binary addition on them outputting what would be the sum and any carry forward.



Full Adder

A continuation of the half adder—this time it can also add in the carry bit brought forward.

These circuits can be chained together to add up binary numbers of any length.



D-Type Flip Flop

The D-Type flip flop is known as a rising edge circuit.

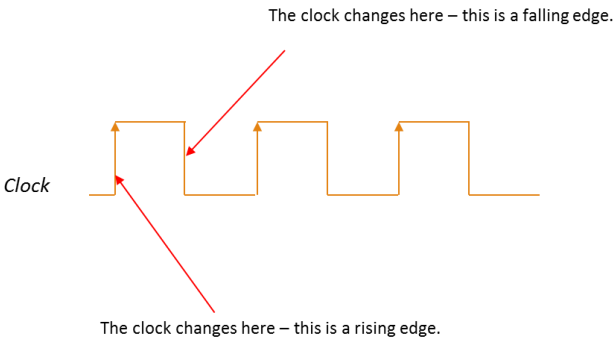
This means the input will only change when the clock impulse is first triggered.

If you flick the switch at any other time it won't have any effect until the clock impulse gets triggered again.

The clock impulse can be drawn like in the diagram opposite. Look for the rising edges – this is when the clock impulse triggers.

Flip flops essentially store one bit of data. The memory in CPU registers are created using flip flops. The intermediate arithmetic storage used in the ALU is created using flip flops.

SRAM used in cache uses flip flops as well.



Simplifying Boolean Expressions

De Morgan's First Law

$\neg(A \vee B) = \neg A \wedge \neg B$

De Morgan's Second Law

$\neg(A \wedge B) = \neg A \vee \neg B$

General rules

$A \wedge 0 = 0$
$A \wedge 1 = A$
$A \wedge A = A$
$A \wedge \neg A = 0$

$A \vee 0 = A$
$A \vee 1 = 1$
$A \vee A = A$
$A \vee \neg A = 1$

Commutation

$A \wedge B = B \wedge A$
$A \vee B = B \vee A$

Association

$A \wedge (B \wedge C) = (A \wedge B) \wedge C$
$A \vee (B \vee C) = (A \vee B) \vee C$

Absorption

$A \vee (A \wedge B) = A$
$A \wedge (A \vee B) = A$

Distribution

$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
$(A \vee B) \wedge (C \vee D) = (A \wedge C) \vee (A \wedge D) \vee (B \wedge C) \vee (B \wedge D)$

Double negation

$\neg\neg A = A$

Karnaugh Maps

Expression ¬A∧ B∧ C∧D ∨ ¬A∧B∧C ∨ ¬D

		AB	AB	AB	AB
		00	01	11	10
CD	00	1	1	1	1
CD	01				
CD	11		1		
CD	10	1	1	1	1
Simplified		¬A∧B∧C ∨¬D			

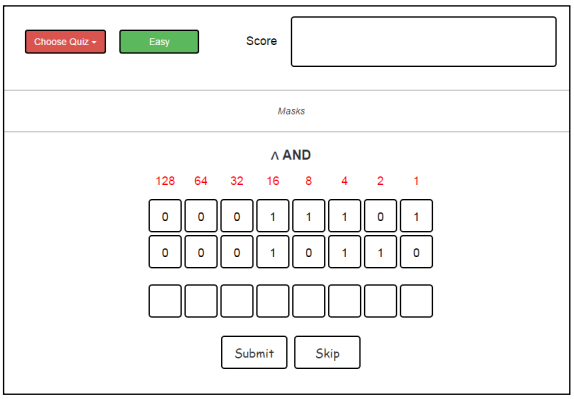
RULES TO REMEMBER

1. Make sure you get the 0s and 1s at the top correct – remember they don't go in normal binary order – you only change 1 bit at a time.
2. Only the ones in the diagram count – don't even bother writing zeros in.
3. You have to make your groups in either rectangles or squares. i.e. no diagonals or any other shape.
4. Make your groups as large as possible – that's how you simplify the expression.
5. Every 1 has to be in a group. If it ends up being on its own that's ok its just becomes a group of 1.
6. 1s can be in groups of 1, 2, 4, 8 etc. You can't have a group of 3 or 5.
7. Overlapping groups is fine.
8. Wrap around is fine.

Number Representation

For the exam you will be expected to be able to be able to:

- Convert positive integers to binary and back
- Convert positive integers to hex and back
- Convert between binary and hex
- Use sign and magnitude to represent negative numbers
- Use two's complement to represent negative numbers
- Add and subtract binary numbers
- Shift bits left and right and understand how this effects the number (i.e. doubles / halves the number)
- Add a bitwise mask of OR, AND, XOR and NOT.



You can practice all these skills using the binary quiz available at www.learncomputing.org.

Normalised Floating Point Binary Representation

8	4	2	1		1/2	1/4	1/8	1/16
0	0	1	1	•	1	1	0	0

Just like in normal decimal representation, the binary point comes to the right of the 1 column. Everything to the right of the point is a fraction. The number above is therefore 3.75

The binary point can become a **floating point** in some representations. This means it moves a certain number of places either to the left or right.

In a normalised floating point binary representation there are two ingredients. The **mantissa** and the **exponent**.

The mantissa is the part that tells you what the number is. The exponent quite simply tells you how far to move the binary point.

How to spot if a number is normalised

In a normalised number the first two digits of the mantissa **have to be different**. This means that the digits will **always** be 01 for a positive number and 10 for a negative number.

e.g.

1000 0001 - normalised: first two numbers are different (negative)
0111 0110 - normalised: first two numbers are different (positive)
0011 0001 - not normalised: first two numbers the same

Convert from normalised floating point to denary

Step 1 – check if the mantissa and / or the exponent are negative. If the first number of either is a 1 then you know it is a negative number. Remember this for later.

0100 0110 - both mantissa and exponent are positive
1000 0110 - negative mantissa, positive exponent

Step 2 - work out the value of the exponent. You need to use normal binary conversion to change it from binary to denary. Remember if it is negative the left-most column becomes negative.

0011 = 1 + 2 = exponent is 3
0111 = 1 + 2 + 4 = exponent is 7
1011 = -8 + 2 + 1 = exponent is -5

Step 3 - move the binary point by the number of the exponent. The binary point always starts to the right of the first number. If the exponent is positive you move the binary point to the right and make the final number larger. If the exponent is negative you more the binary point to the left and make the final number smaller.

0100 0110 = 0.100 (exponent 6) = 100000.
0100 1010 = 0.100 (exponent -6) = 0.0000001
1001 0011 = 1.001 (exponent 3) = 1001

Step 4 - re-write your column headings and convert it to denary. Remember if the mantissa is negative the leftmost bit is negative as well.

100000 = 32
0.0000001 = 1/256
1001 = -7

Convert from denary to normalised floating point

Step 1 - convert the number into normal binary. If it is a negative number remember to use a two's complement representation.

32 = 0100000.0
-32 = 100000.00
1.5 = 00001.100

Step 2 - count how many places to move the binary point so that the **most significant bit is to the right of the point**. The most significant bit is the highest valued 1 in a positive number. In a negative number it is the highest 0 as the bits have been flipped.

32 = 0100000.00 = 0.1000000 = exponent is 6
-32 = 100000.00 = 1.000000 = exponent is 5
1.5 = 00001.100 = 0000.1100 = exponent is 1

Step 3 - convert the exponent to binary and lose the binary point

01000000 0110
10000000 0101
00001100 0001

Remember the amount of bits for the mantissa and exponent is not fixed and can be changed.

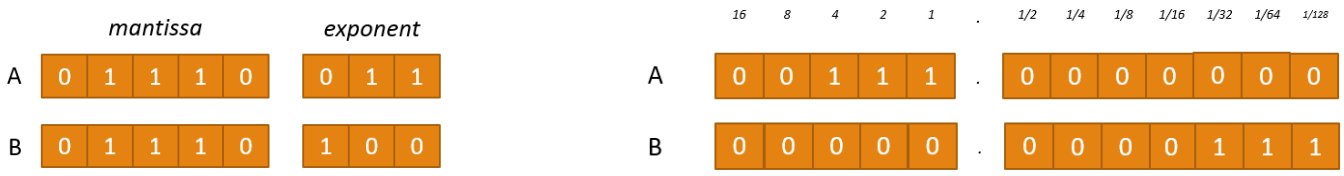
The more bits available for the mantissa the more accurate the number.
The more bits available for the exponent the greater the range of numbers.

When using a relatively small amount of bits e.g. 4-bits for each lots of numbers cannot be represented accurately.

Adding / Subtracting Normalised Floating Point Binary

The process is:

1. Un-normalise the numbers by writing them out on a full number line.
2. Add them as normal. If subtracting you will need to convert one to two's complement first so you are adding the negative.
3. Convert back to normalised format.



Once you put the numbers on a full number line the process is quite easy.

laws

The Data Protection Act 1998

When computers first became widespread the government grew concerned that people's personal data could be misused. This act attempts to keep people's data safe. Note it includes computerised data only.

If an organisation stores data about customers they need to apply for licensing from the Information Commissioner and follow the 8 principal of the Act:

1. Data must be collected and used fairly and inside the law.
2. Data must only be held and used for the reasons given to the Information Commissioner.
3. Data can only be used for those registered purposes and only be disclosed to those people mentioned in the register entry. You cannot give it away or sell it unless you said you would to begin with.
4. The data held must be adequate, relevant and not excessive when compared with the purpose stated in the register. So you must have enough detail but not too much for the job that you are doing with the data.
5. Data must be accurate and be kept up to date. There is a duty to keep it up to date, for example to change an address when people move.
6. Data must not be kept longer than is necessary for the registered purpose. It is alright to keep information for certain lengths of time but not indefinitely. This rule means that it would be wrong to keep information about past customers longer than a few years at most.
7. The data must be kept safe and secure. This includes keeping the information backed up and away from any unauthorised access. It would be wrong to leave personal data open to be viewed by just anyone.
8. The files may not be transferred outside of the European Economic Area (that's the EU plus some small European countries) unless the country that the data is being sent to has a suitable data protection law. This part of the DPA has led to some countries passing similar laws to allow computer data centres to be located in their area.

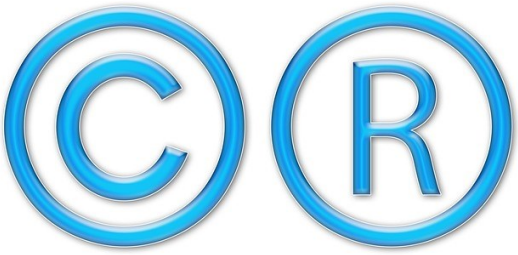
The Computer Misuse Act 1990

This law means that the following actions are illegal:

- Unauthorised access to computer material (hacking)
- Unauthorised access with intent to commit or facilitate a crime (hacking with intent)
- Unauthorised modification of computer material. (hacking where you change data)
- Making, supplying or obtaining anything which can be used in computer misuse offences (making and distributing any kind of malware)

Being found guilty of one of these offences can land you up to five years in prison and an unlimited fine.

The Copyright Design and Patents Act 1988



Everyone should have the right to be paid for the work they have done and this includes making products like books, music, TV shows, films and any other intellectual property.

The Copyright Design and Patents Act 1988 means it is illegal to copy somebody else's work without their permission. So, for example, copying a film from a friend is illegal, downloading music without paying is illegal and so on.

Modern technologies have meant that breaches of this law have been rife in recent times and have had serious negative effects on the music and film industries.

Breaking this law can land you up to 10 years in prison and up to a £50,000 fine.

The Regulation of Investigatory Powers Act 2000

This Act states what powers law enforcement organisations have when attempting to deal with threats.

It is very controversial as it gives them the power to "snoop" on citizens and many believe that we now live is a society of "extreme surveillance" with little or no privacy.

Here is a summary of the powers and when they are allowed to be used (source Wikipedia)

Type	Typical use	Reasons for use	Type of public authority permitted to use	Level of authorisation required
Interception of a communication	Wire taps and reading post	In the interests of national security, for the purpose of preventing or detecting serious crime and for the purpose of safeguarding the economic well-being of the United Kingdom	Defence Intelligence, GCHQ, HM Revenue and Customs, Secret Intelligence Service, Security Service and territorial police forces of Scotland	Warrant from Home Secretary or Cabinet Secretary for Justice
Use of communications data	Information about a communication, but not the content of that communication (phone numbers, subscriber details)	In the interests of national security, for the purpose of preventing or detecting crime or of preventing disorder, in the interests of the economic well-being of the United Kingdom, in the interests of public safety, for the purpose of protecting public health, for the purpose of assessing or collecting any tax, duty, levy or other imposition, contribution or charge payable to a government department and for the purpose, in an emergency, of preventing death or injury or any damage to a person's physical or mental health, or of mitigating any injury or damage to a person's physical or mental health.	As listed below	Senior member of that authority
Directed surveillance	Following people	In the interests of national security, for the purpose of preventing or detecting crime or of preventing disorder, in the interests of the economic well-being of the United Kingdom, in the interests of public safety, for the purpose of protecting public health and for the purpose of assessing or collecting any tax, duty, levy or other imposition, contribution or charge payable to a government department.	As listed below	Senior member of that authority
Covert human intelligence sources	Informers; undercover officers	In the interests of national security, for the purpose of preventing or detecting crime or of preventing disorder, in the interests of the economic well-being of the United Kingdom, in the interests of public safety, for the purpose of protecting public health and for the purpose of assessing or collecting any tax, duty, levy or other imposition, contribution or charge payable to a government department.	As listed above	Senior member of that authority
Intrusive surveillance	Bugging houses/vehicles	In the interests of national security, for the purpose of preventing or detecting serious crime and in the interests of the economic well-being of the United Kingdom.	GCHQ, Secret Intelligence Service, Security Service, Ministry of Defence, armed forces, Her Majesty's Prison Service or Northern Ireland Prison Service.	Authorisation from Home Secretary or Cabinet Secretary for Justice
			The territorial police forces, the Ministry of Defence Police, the British Transport Police, the Royal Navy Regulating Branch, Royal Military Police, Royal Air Force Police and HM Revenue and Customs.	Authorisation from the head of the relevant agency: chief constable of any of the territorial police forces, the Ministry of Defence Police or the British Transport Police, the Provosts Marshal of the Royal Navy Regulating Branch, Royal Military Police or the Royal Air Force Police and any customs officer designated for the purposes by the Commissioners of Revenue and Customs.

Ethics

Issue	What's the Crack?	For	Against
Computers in the workforce	Computers are replacing manual jobs more and more	• Computers can often complete the same job more quickly. • Reduces the need for humans to do repetitive tasks—more interesting work. • Can save a business salary costs. • Technology can lead to innovative working practices and improved collaboration.	• Costs money to invest in computer systems. • People can lose their jobs. • Computers can't improvise or think on their feet. • Programming years of human experience is currently not possible.
Automated decision making	Computers are being used to make the decisions that were previously left to humans	• Decisions can be made instantly. • A computer can be trusted to follow a set of rules to the letter. • Computers are unaffected by emotion. • Computers don't get tired or have bad days.	• Computers are unaffected by emotion. They can't think ethically. • A mistake in the programming could be catastrophic. • Computers won't be able to bend the rules where a human could. • Dependent on having a power source.
Artificial intelligence	We are attempting to develop computers that can think for themselves	• The future could be very bright with robot workers able to take care of us. • Complex problems could be solved advancing the scientific knowledge of mankind.	• Leading figures believe that the threat of computers "taking over" is very real indeed. • There may come a point where computers can learn faster than we can react to and they decide they don't need us.
Environmental effects	Computers have an effect on the environment	• Digital documents saves paper. • Emailing is much more environmentally friendly than using the post. • Speeding up industrial processes with computer automation can help reduce waste.	• Computers go obsolete very quickly and the old components are often left to landfill. • Computers need electricity and making electricity causes pollution. • Computers contain many noxious chemical components.
Censorship and the internet	Different countries have different levels of internet censorship	• Some believe that censoring indecent content like drugs, guns, child porn is common sense.	• Some believe freedom of the internet should be absolute. • Some countries have very restricted internet so that their governments can maintain control of the population.

Issue	What's the Crack?	For	Against
Monitor behaviour	Computers can be used to monitor the behaviour of citizens or a workforce	• Useful for bosses to make sure their workers are always working. • Good for law enforcement to catch criminals.	• Adds stress to workers who are constantly under pressure. • Leads to a "surveillance society" where privacy no longer exists.
Analyse personal information	Computers can be used to analyse personal information	• Your online habits can be analysed to provide you with useful recommendations and offers.	• Can lead to you being spammed by adverts. • If personal information falls into the wrong hands it can lead to fraud.
Piracy and offensive communications	Copying files and saying offensive things using computers	• Some argue that freedom of speech should be absolute.	• Cyberbullying can negatively affect all concerned. • Piracy takes money away from the industries which produce the content jeopardising future production
Layout, colour paradigm and character	Different cultures have different expectations of layout, colour and character	• Colours can mean different things in different cultures e.g. red is often the symbol for danger in the western world but means luck in China. • Different countries use different alphabets. • Some countries read from right to left and from bottom to top. A systems designer will need to be aware of this.	