N5

National 5

Computer Systems

Learning Intentions

By the end of this unit:

You should be able to:

Data Representation
Page 3 - 23

- Describe and exemplify the use of binary to represent positive integers.
- Describe floating point representation of positive real numbers, using the terms mantissa and exponent.
- Convert from binary to denary and vice-versa.
- Describe extended ASCII code (8-bit) used to represent characters.
- Describe the vector graphics method of graphic representation for common objects:
    - rectangle
    - ellipse
    - line
    - polygon
- with attributes:
    - co-ordinates
    - fill colour
    - line colour
- Describe the bit-mapped method of graphics representation.
- Describe and compare the various media types used for storing text, graphic, audio and video
- Describe what is meant by compression and factors affecting file size.

Computer Structure
Page 24 - 33

- Describe the purpose of the basic computer architecture components and how they are linked together:
    - processor (registers, ALU, control unit)
    - memory locations with unique addresses
    - buses (data and address)
    - Explain the need for interpreters and compilers to translate high-level program code to binary (machine code instructions).

Environmental Impact
Page 34 - 35

- Describe the energy use of computer systems, the implications on the environment and how these could be reduced through:
    - settings on monitors
    - power down setting
    - leaving computers on standby

Security Precautions
Page 36 - 39

- Describe the role of firewalls.
- Describe the use made of encryption in electronic communications.

Data Representation

What is a computer?

Have you ever wondered what the word "computer" means? If you look it up in a dictionary, you will find something like:

"an electronic device which runs a program to process data at great speed"

We will come to the words **program** and **data** soon; however, the word "computer" means "something that computes".

So what does "compute" mean? Well, it means to **calculate** or **work out**.

The very first computers were actually **people**. They did sums – all day, every day!

From around the 1700s until the 1950s, teams of these "computers" used to carry out complex calculations (and check each other's work).

Representing Information

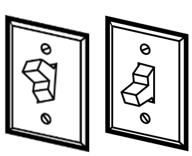
A computer is a machine that processes **information (data)** using **instructions** given to it by a human. Computers are electronic machines, so the instructions and information they work with is represented electronically – by switches.

Switched on

No matter what kind of computer you are using – a desktop PC, smartphone, games console or embedded computer – it is doing one thing: **processing information (data)**. Since a computer is made up of switches, then the information it processes is represented by the positions of switches.

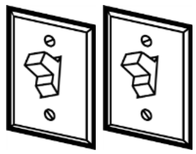
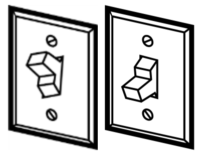
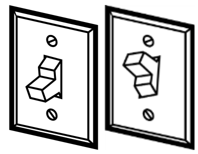
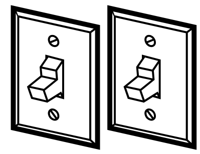
How is this possible? Well, think about an ordinary switch. It has **two** positions – OFF and ON – but we could also call these positions (or make them stand for):

- YES or NO
- BLACK or WHITE
- 0 or 1



If we had **two** switches, there would be **four** possible combinations of these.

Look at the table below that shows the four combinations of two switches and examples of the way that you could represent different kinds of information by these.

Switch positions	 **OFF / OFF**	 **OFF / ON**	 **ON / OFF**	 **ON / ON**
Numbers	0	1	2	3
Colours	Black	Blue	Red	White
Letters	A	B	C	D

Now consider if there were **millions** of switches – you can see how there could be almost endless combinations of OFF and ON for all of these switches.

A modern computer **does** contain millions (even billions) of tiny switches and they are used to represent all such information.

The silicon chip

As you have seen, a computer is basically a collection of electronic switches.

Over the years, the technology used for this has changed from old electronic valves used in Colossus to the **silicon chip**.

Incredibly, a modern silicon chip can contain over a *billion* switches (called **transistors**) on a wafer of silicon smaller than your fingernail!

Binary: The language of computers

No matter what you are using your computer for – watching a movie, listening to music or surfing the web – the **data** (information) that the computer is processing is represented by the positions of millions of switches. Computing scientists like to call the positions 0 and 1 instead of OFF and ON. This is known as **binary**. It may seem strange, but it is easier than you might think.

Bits & Bytes

Computers can sense when a signal being sent is on or off. This is represented by a '1' (on) or a '0' (off). Each individual 1 or 0 is called a **binary digit** or **bit** and it is the smallest piece of data a computer system can work with.

Eight bits are grouped together to make one **byte**.

One byte provides enough codes (256) to represent all of the characters that appear on a standard keyboard. A byte is the basic unit used to measure computer memory size.

The Denary & Binary Systems

In our everyday lives, we use the decimal(denary) or base 10 system. **This means we have 10 digits : 0-9**

Computers use the binary or base 2 system.

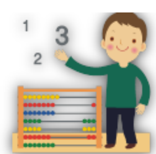
- **There are only two digits: 0 and 1**
- **Each figure is known as a bit**
- **Binary digit**

Denary Number system

Let's look at how a number is made up: 173

Technically this is:

Hundreds	Tens	Units
1	7	3



10^2	10^1	10^0
Hundreds	Tens	Units
1	7	3

Binary Numbers

- Let's look at how the same number is stored in binary:

1010 1101

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

This number is constructed as shown above.

These values come from:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	1	0	1

So…. 128 + 32 + 8 + 4 + 1 = 173

How to convert Binary to Denary

1. Create your table with the values in the top

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

2. Insert your binary value into the table
3. Add the place values that have a binary 1 in them:

64 + 32 + 16 + 4 + 2 = 118

How to Convert Denary to Binary – Method 1

We will only be using 8 bits – there are two methods for converting decimal to binary

1. Create your table with the values in the top

128	64	32	16	8	4	2	1

2. Starting at the left hand side (128), fill in if that value should be switched on (1) or switched off (0)

3. Example – convert 100

- **128 is too big so 0**
- **64 fits so 1 leaving 36**
- **32 fits so 1 leaving 4**
- **And so on…….**

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

How to convert a Denary to Binary – Method 2

This is the "Divide by 2" Method.
This should be used when working with larger numbers.

Convert 180 into Binary

Divide by 2	Remainder
180 / 2 = 90	R 0
90 / 2 = 45	R 0
45 / 2 = 22	R 1
22 / 2 = 11	R 0
11 / 2 = 5	R 1
5 / 2 = 2	R 1
2 / 2 = 1	R 0
1 / 2 = 0	R 1

Write down the 1s and 0s from bottom to top

1 0 1 1 0 1 0 0

Check

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

128 + 32 + 16 + 4 = 180

The largest positive integer that can be stored....

An integer is a whole number

The largest positive integer that can be stored in 8 bits = 255 (All 1's)

But what's the largest that can be stored in 5 bits or 14 bits for example?

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Largest in 5 bits? 5 columns all 1 16+8+4+2+1 = 31

But what about larger numbers, there must be a rule. You may have noticed that 31 is one less than the 6^{th} column. (2^5 = 32)

Therefore the rule is **2^n + 1** **Try it out for any number**

Advantages of the Binary system

Summary

- Binary only has **two** values– 0 & 1
- A single 0 or 1 is known as a **bit** (**Bi**nary Digi**t**)
- The place values in binary start on the right at 1 and double every time going to the left.

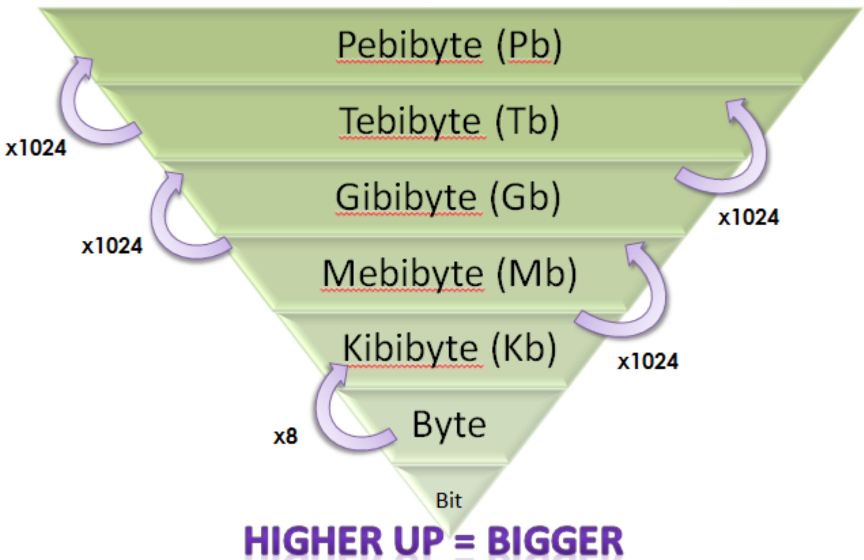
Storage Terms

Which is bigger, pennies or pounds?

It is important to be able to sort terms into the correct order.
Working with computer storage terms is no different………

If buying a phone would you take a 512Mb version or a 4Gb version? Why?

Storage Terms



Converting Between Terms

If going from a **smaller unit** to a **larger unit** you divide.
If you wanted to know how many Mebibytes were in 2048Kb then you would **divide** by 1024

If going from a **larger unit** to a **smaller unit** you multiply.
If you wanted to know how many Gibibytes were in 5 TB then you would multiply by 1024.

Storing REAL numbers – Floating point representation

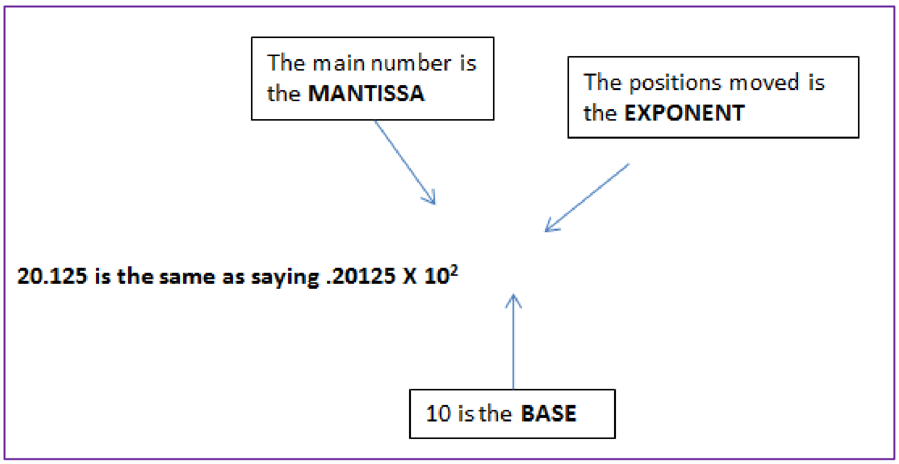
Numbers with a decimal point are known as **real** numbers.

15.215 is an example of a real number.

In computers, real numbers are represented by storing two parts of the number:

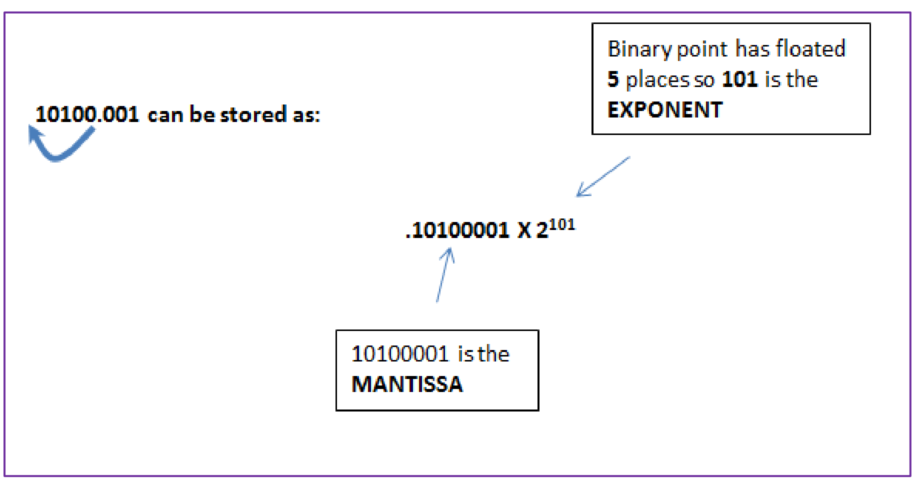
- **Mantissa**
- **Exponent**

In the denary system that we use:



The decimal point is fixed and the numbers have moved **two places to the right**.

In the binary system that computers use:



Since the **BASE** is always 2, the computer only has to store the **MANTISSA** and **EXPONENT**

Fully worked example

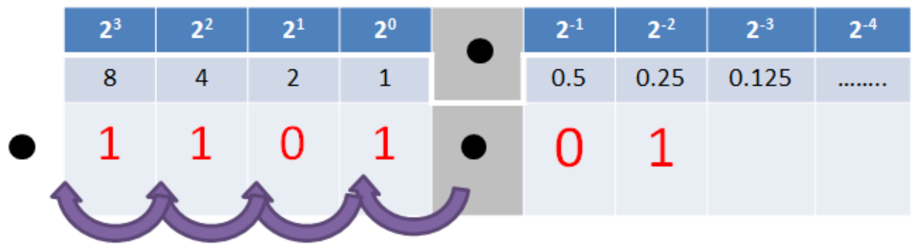
From previous lessons, you should be aware of the placeholders

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	●	2^{-1}	2^{-2}	2^{-3}	2^{-4}
128	64	32	16	8	4	2	1		0.5	0.25	0.125	....

We have now added the columns after the decimal point.

For example:

13.25



mantissa - <u>110101</u> exponent – 4 places to left so.... <u>100</u>

Example Questions

Q: Show how 10100.1111 would be represented in binary.

A: mantissa - <u>101001111</u> exponent – 5 places to left so.... <u>101</u>

Q: Show how 11.1101 would be represented in binary.

A: mantissa - <u>111101</u> exponent – 2 places to the left so ..... <u>10</u>

Storing TEXT

Computers can only use and understand binary digits which are 0's and 1's.
We have already looked at how binary can be used to store numbers.

How can we store text?
By storing them as binary numbers using the ASCII code

ASCII Codes

American Standard Code for Information Interchange

Each character on a keyboard has its own ASCII code. This is a binary value that represents each character that can be seen on the screen.

An extract of the ASCII Table is shown below:

- **The ASCII table has 128 values**
  - **52 just for text. 10 for numbers**
  - **Spacebar and tab key have codes too**
  - **What about the rest?**

Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol
48	0	78	N	64	@	97	a
49	1	79	O	65	A	98	b
50	2	80	P	66	B	99	c
51	3	81	Q	67	C	100	d
52	4	82	R	68	D	101	e
53	5	83	S	69	E	102	f

For example A = 0100 0001 (65)
Each ASCII value takes up **1 byte** of storage.

Control Characters

The rest of the ASCII code are reserved for Control characters

These are non-printable characters that have an effect such as

- **Escape key**
- **Enter key**
- **Delete key**
- **Shift key**
- **and so on……**

Character Set

The character set is the name given to the complete set of characters that the computer can represent. Different character sets are used to represent different languages

The character set can alter the layout of the keyboard
For example on American layout keyboards the @ sign is above the number 2 not the " mark.

Summary

- Text is stored in the computer using ASCII values
- 1 ASCII Value = 1 byte of memory
- Control characters are the non-printable characters which have an effect on the screen, such as the Enter Key
- The character set is the name given to the entire set of characters that the keyboard can produce

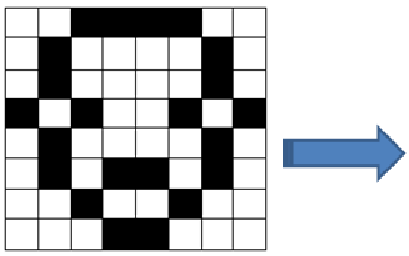
Storing Graphics

There are two types of graphic you need to be familiar with:

1. **Bitmap**
2. **Vector**

Bitmapped Graphics

A bitmap image stores each individual pixel.
First you need to understand black and white images. These are constructed as follows:



0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
1	0	1	0	0	1	0	1
0	1	0	0	0	0	1	0
0	1	0	1	1	0	1	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0

0 is stored for a white pixel, 1 is stored for a black pixel
Each black and white pixel takes up 1 bit of storage

Resolution **is the term given to the amount of pixels that the image is made of**

- Also used for the amount of pixels displayed on the screen
- Higher resolution = clearer and more detailed picture

Coloured Bitmaps

The colour bitmap method is exactly the same as for black and white with one difference.

- Each pixel is not black and white but can represent a variety of colours.
- Each pixel has a binary value representing the colour. The amount of colours is known as the **bit depth**.
- So an image with 8 bit colour depth could have 256 Colours
- True Colour is defined as an image with 24bit colour depth. 16,777,216 colours!

Colour depth

When we have been looking at black and white images, each pixel requires 1 bit of storage but what happens when we factor in some colour?

8 bit colour depth - each pixel = 8 bits and can be one of 256 colours

16 bit colour depth - each pixel = 16 bits (2^16) = 65536 colours

24 bit colour depth - each pixel = 24 bits (2^24) = 16777216 colours

True Colour

True Colour is defined as an image with **24bit** bit depth. This means that 16,777,216 colours can be represented.

The colour code for each pixel is constructed of a single 8 bit number for each of the main 3 additive colours.

Increasing Bit Depth

Vector Graphics

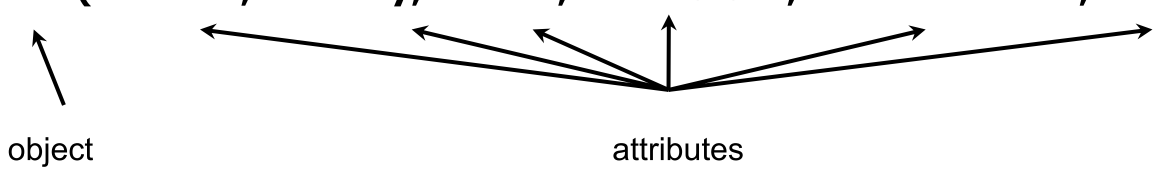
This image is an example of a vector image. A vector image stores the graphic by storing the attributes or information about the shapes that are used.

For example attributes for the eyes(ellipses) may be

- Centre X/Y-co-ordinates
- Fill Colour
- Border Style
- Border Colour etc…

Example

circle(centrex, centrey, radius, line colour, line thickness, fill colour)



i.e. it doesn't store the picture itself but the instructions of how to draw the picture.

A line might be stored as

line(startx, starty, end x, endy, line colour, fill colour).

Obviously, when stored in the computer each attribute would be stored as a binary number.

There are several common objects that you need to know how they are represented. These are:

- **Rectangle**
- **Ellipse**
- **Line**
- **polygon.**

For each of these you need to know how the attributes of:

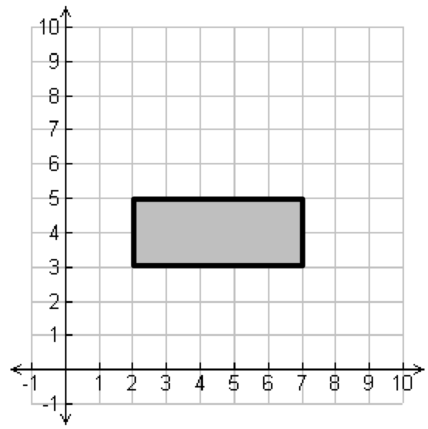
- **co-ordinates**
- **line colour**
- **fill colour.**

Rectangle:

Rectangle (height, width, x, y, fill colour, line colour)

For example:
rectangle (2, 5, 2, 3, grey, black) would draw:

Height	the height of the rectangle
Width	the width of the rectangle
X,Y	the starting coordinates of the rectangle
Fill colour	the colour the rectangle will be filled with
Line colour	the colour of the line the rectangle is drawn with

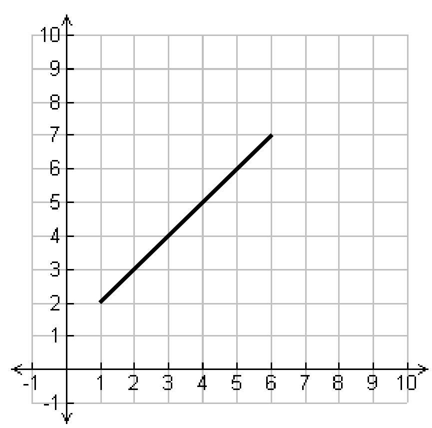


Line:

line(x1, y1, x2, y2, line colour)

For example: line(1, 2, 6, 7, black) would draw:

x1, y1	Coordinates of beginning of line
x2, y2	Coordinates of end of line
line colour	the colour of the line



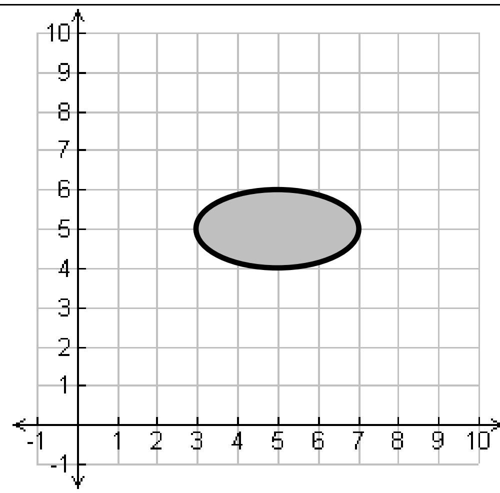
Ellipse:

ellipse(cx, cy, rx, ry, fill colour, line colour)

For example:

ellipse(5, 5, 4, 2, grey, black) would draw:

cx, cy	Centre point of the ellipse
rx	Horizontal radius of ellipse
ry	Vertical radius of ellipse
Fill colour	the colour the ellipse will be filled with
Line colour	the colour of the line the ellipse is drawn with

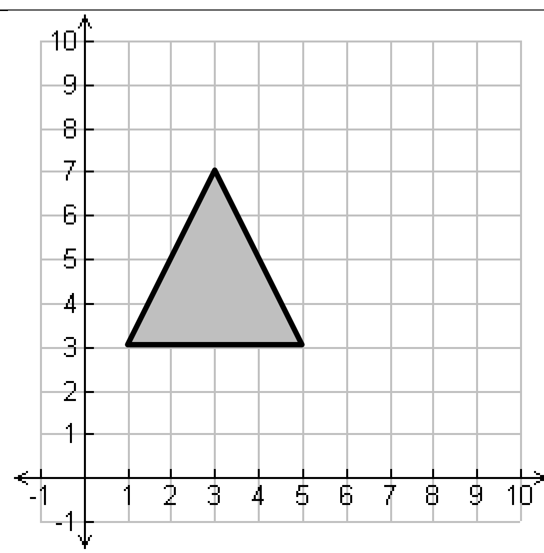


Polygon:

polygon(x1, y1, x2, y2, x3, y3, fill colour, line colour)

For example: polygon(1, 3, 3, 7, 5, 3, grey, black) would draw:

x1, y1	Coordinates of first point of polygon
x2, y2	Coordinates of second point of polygon
x?, y?	Coordinates of next point of poygon. E.g. a pentagon would have 5 points
Fill colour	the colour the ellipse will be filled with
Line colour	the colour of the line the polygon is drawn with



Summary

Bitmap Graphics – Pros and Cons

Advantages

- ✓ Can be manipulated at pixel level
- ✓ Can create a wide array of graphic effects
- ✓ Can represent photo-realistic images

Disadvantages

- ✗ Requires large storage space
- ✗ Image becomes jagged when scaled

Vector Graphics – Pros and Cons

Advantages

- ✓ Do not lose quality when scaled
- ✓ Require less storage space
- ✓ Objects are easily moved/manipulated
- ✓ Resolution independent

Disadvantages

- ✗ Cannot be edited at pixel level
- ✗ Cannot show photo realistic scenes
- ✗ Will usually require particular applications to open

Standard File Formats

- A standard file format is a type of file that can be recognised by **all computers**.
- They don't need any special software in order to open them which allow them to be open on any computer with a variety of software. They are **portable**.
- Portable files can be easily transferred from one computer to another.

Text

Text (txt) files store only the characters contained in the document. Any formatting is **ignored** and not saved.

This is a document about the history of computers

This is a *document* about the **history** of <u>computers</u>.

Rich Text Format (RTF) stores the characters as well as **some formatting** information.

This makes the file size of RTF larger than plain text files.

Standard File Format	Use	File Size
TXT (Text)	Plain Text Only	Very small (1 byte per character)
RTF (Rich Text Format)	Formatted Text	Larger than Txt (due to formatting)

Audio

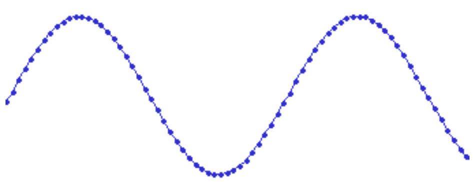
WAV is a standard for storing uncompressed sound that has been sampled by a computer. This makes wav files very large in size.

MP3 is a sound format that compresses sound files by removing parts of the sound that our ears cannot hear.

MP3 files have a much smaller file size than wav with very little reduction in sound quality

Sampling Rate

The sampling rate is **how often** a sound is "listened to", converted into digital and stored **each second**.



Each dot here represents a sample (and must be stored).

The more times per second a sound is sampled, the greater the quality will be (and the larger the file size).

Standard File Format	Use	File Size
WAV	Uncompressed Sound	Very Large (depending on sample rate)
MP3	Compressed Sound	Very Small

Increased Sample Rate = Increased File Size

Decreased Sample Rate = Decreased File Size

Compression = Reduced File Size

Video

AVI is a standard for storing video that has been captured by a computer. AVI uses little compression making file sizes very large.

MP4 is a video format that compresses video files by encoding only the **changes** between frames.

MP4 files have a much smaller file size than AVI with very little reduction in quality.

Standard File Format	Use	File Size
AVI	Video (with some compression)	Very Large
MP4	Compressed Video	Small

Greater Compression = Reduced File Size

Graphics

JPEG (Joint Photographic Experts Group)

JPEG is a graphic file format that uses **compression** to reduce file size.

JPEG compression involves removing parts of the image that our eyes normally ignore.

JPEG files have a much **smaller file size** than BMP with very little reduction in quality.

This is why JPEG files are commonly used for digital cameras, websites and when emailing images.

GIF (Graphics Interchange Format)

GIF is a graphic file format that also uses **compression** to reduce file size.

GIF images only allow 256 colours so are unsuitable for high quality photographs.

GIF images are commonly used for cartoons, logos and especially **animations.**

PNG (Portable Network Graphics)

PNG is another graphic file format that uses **compression** to reduce file size.

PNG files achieve better compression than GIF but allow many more colours so file sizes will still be larger.

PNG allows control over **transparency** in images.

Summary

Standard File Format	Use	File Size
JPEG	Compressed. High Quality Photographs	Small
GIF	Compressed. Animations, cartoons	Small
PNG	Compressed. Transparency	Small

Compression is the method of reducing the file size of any text, graphic, audio or sound files. Compressing allows for more files to be stored on a computer/portable storage device because file sizes are reduced therefore taking up less storage space.
It also reduces the upload, download or transfer time of files as they are smaller in size.
However, compressing larger files can take time and if compression is used repeatedly on a file the quality of the file may become damaged.

Factors Affecting File Size / Quality

Resolution (graphics)

Increase = Increased quality = Increased file size

Colour Depth (graphics)

Increase = More colours = Increased file size

Sampling Rate (audio)

Increase = Increased quality = Increased file size

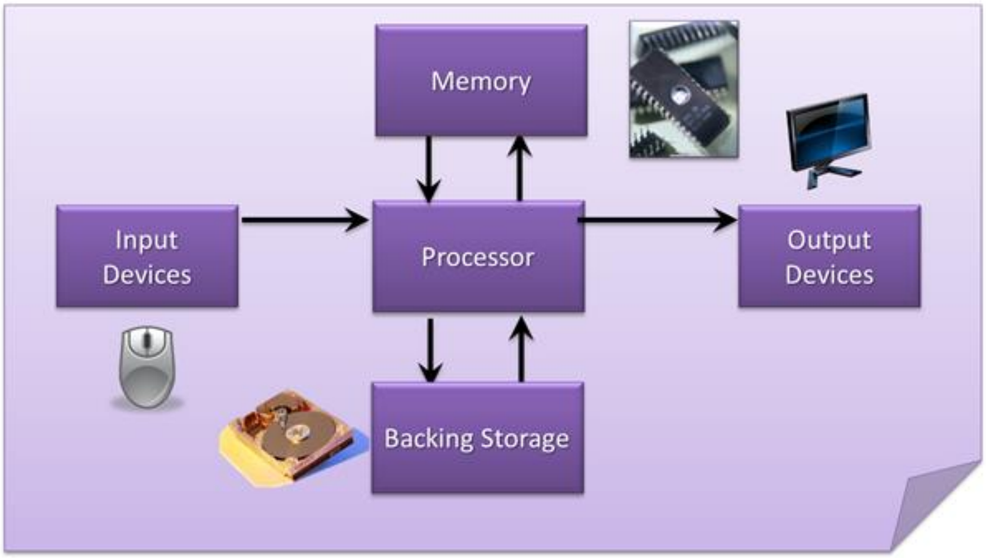
Compression(all)

Increase = Reduced Quality= Decreased file size

Decreasing all of the above has the opposite effect

Computer Structure

A Basic Computer System



The 5 Block Diagram

Input Devices
These are devices that enter data **INTO** the computer. i.e. Keyboard and mouse

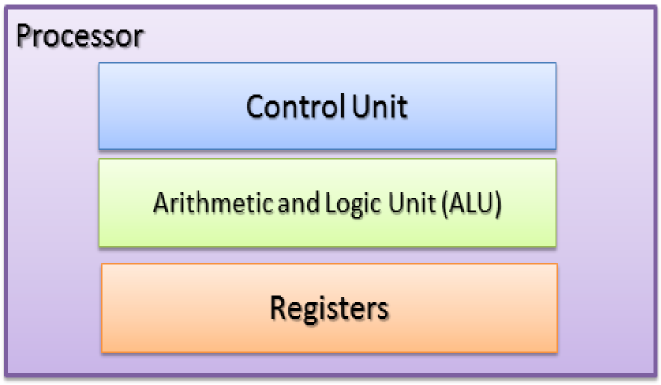
Output Devices
These are devices that generate output **FROM** the computer i.e. printer.

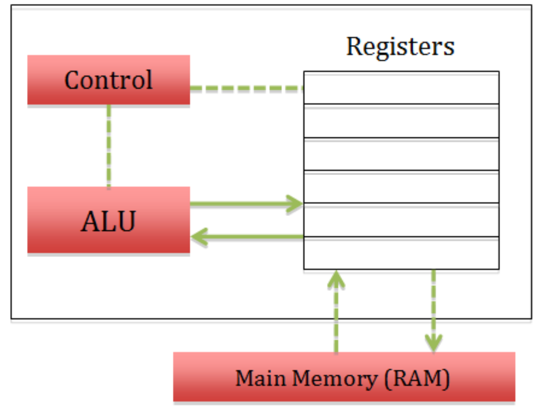
Memory
There are two types of memory. RAM and ROM (more on this later)

Backing Storage
This is required to store **permanent** copies of our files.

The processor
The processor is the 'brain' of the computer.
The CPU is the main component within a computer where instructions are processed and computations carried out.

The processor has 3 parts shown below:





Control Unit

- Makes sure program instructions are carried out/executed in the correct order.
- Makes sure all operations are carried out at the correct time.
- Sends out Control Signals

Arithmetic Logic Unit (ALU)

- Carries out calculations + - x /
- Performs logical operations (AND, NOT, OR)
- Deals with comparisons (< > => =<)

Registers

- Small and fast, temporary storage locations within the processor.

Processor Speed (clock speed)

The speed of a processor is measured by the speed of its internal **clock**. Control bus sends out a constant, steady clock pulse

Frequency is measured in Hertz (Hz) , meaning per second

Typical computer is around **3GHz**.

That's **3000 million** pulses per second

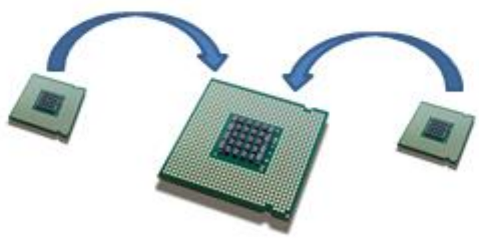
One action is carried out by the processor on **each** clock pulse.

The faster the clock speed, the more instructions can be carried out in a fixed time.

Processor Type

In the past, each CPU chip had a **single core**. This means that it contains one processor and can do **one** thing at a time.

Nowadays, CPU chips contain **dual core** (two cores) **or more** meaning they can do **more than one** thing simultaneously.

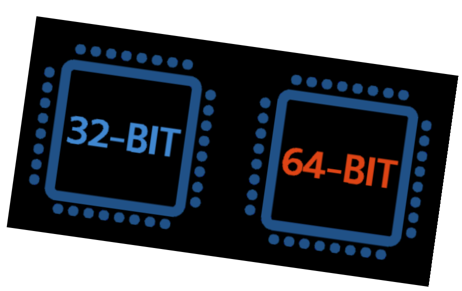


The more cores a CPU has, the more tasks it **can be** programmed to carry out at the same time, improving the performance of the computer.

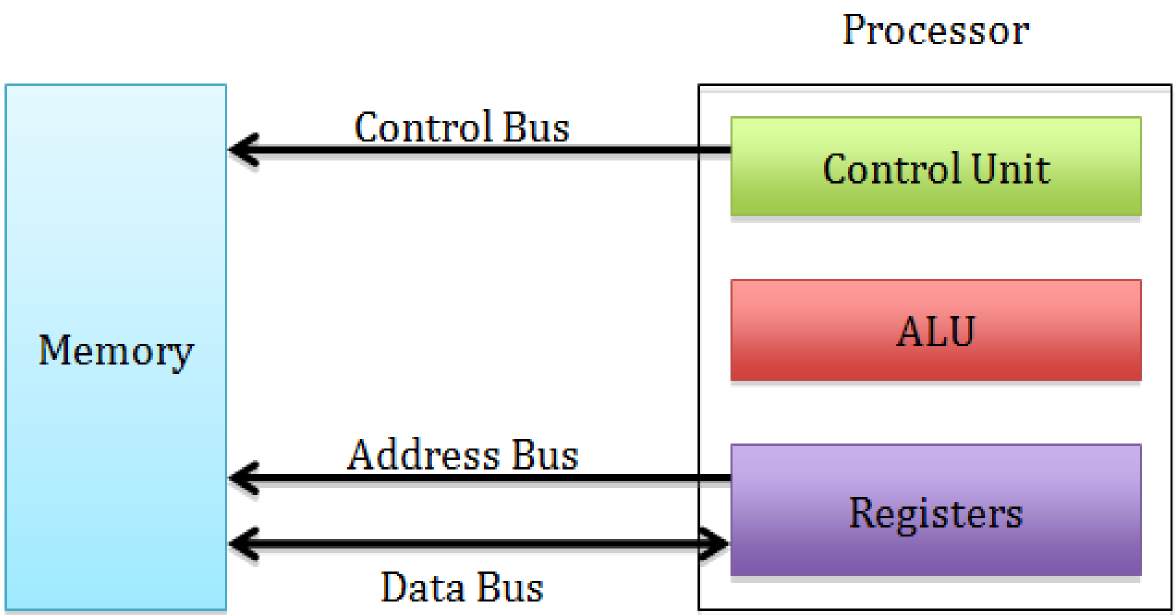
If a piece of software isn't programmed to make use of these extra cores then they will not make any difference to the system performance.

32 bit and 64 bit computing?

You often hear a processor being described as 32 bit or 64 bit. This is usually referring to the '**word length**' of the computer. This is the largest binary number that the processor can manipulate in one operation.



System Buses



A **bus** is a collection of wires which can carry data. The Processor (CPU) has buses. These are multiple lines that connect the processor and main memory and used to transfer data and send signals between them.

Address Bus

Address Bus is used to specify the address of the memory location that is to be read from or written to. The bus is uni-directional (one way). The address bus is made up of parallel wires each carrying a single bit. The size of the address bus will determine how many memory locations can be directly accessed,

2 width of address = Number of Unique addresses possible

Modern computers will typically have an address bus 32 lines wide although 64-bit address buses are now becoming normal in everyday computers.

Data Bus

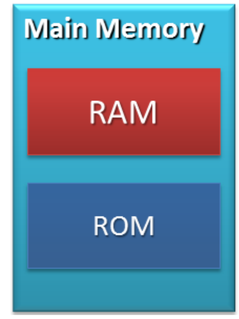
This bus is used to transfer data between main memory and the processor. It is bi-directional (two way) since data can be transferred from a memory location and vice versa.

Control Bus

The control bus is made up of individual lines with specific functions giving instruction to the rest of the system from the control unit:

- Read: used to initiate a memory read operation which reads the contents of a memory location into the processor
- Write: used to initiate a memory write operation which writes an item of data from the processor into a memory location
- Clock: sends a series of pulses into the processor to synchronize events. The time interval between pulses is called a clock cycle.
- Reset: causes the computer to stop the current program and then reboot
- Interrupt: peripheral devices such as printers can send a signal on the interrupt line into the processor when they require attention.
- NMI (Non-Maskable Interrupt): requires serious attention such as a power failure and cannot be ignored

Main Memory



Main memory falls into two types:

1. **RAM -** Random Access Memory
2. **ROM-** Read Only Memory

RAM – Random Access Memory

RAM is the computers short term memory.
If a machine is quoted as having 8GB of Ram then it has 8GB into which programs and the operating system is loaded into.
RAM is **volatile**. This means that it requires power to store data. When power is lost the contents of RAM are lost.

- The processor **can write to and read from RAM** at high speed
- Data held in RAM can be changed
- **All data in RAM is lost when the power is switched off**

ROM – Read Only Memory

There is a problem with just having **RAM**. If the contents of **RAM** are lost then the power is lost how does the computer know what to do when it switches on? **ROM** (**R**ead **O**nly **M**emory) is memory that will always retain its contents even when power is lost. This concept can be used by mobile phones to store their operating systems.

- Data is **stored permanently in ROM**,
- Data is **not lost when the power goes off**
- Data in ROM cannot be changed

Difference between RAM and ROM

Feature	RAM	ROM
Permanent storage for operating or control programs		✓
Temporary storage for programs and data	✓	
Data can be **read** from /**written to** memory	✓	
Data can only be **read** from memory		✓

Addressability

Your house has a unique address - No two houses have the same address!
This is the same for memory. Main memory consists of a number of storage locations, each of which is identified by a unique address. The ability of the CPU to identify each location is known as its addressability.

Each location stores a word i.e. the number of bits that can be processed by the CPU in a single operation. Word length may be typically 16, 24, 32 or as many as 64 bits.

A 32 bit address bus has 32 **parallel** wires each switched on (1) or off (0) that can address locations starting from:

0000 0000 0000 0000 0000 0000 0000 0000 (decimal 0)

up to and including address:

1111 1111 1111 1111 1111 1111 1111 1111 (decimal $2^{32}-1$)

making a total of 2^{32} addresses

Translation

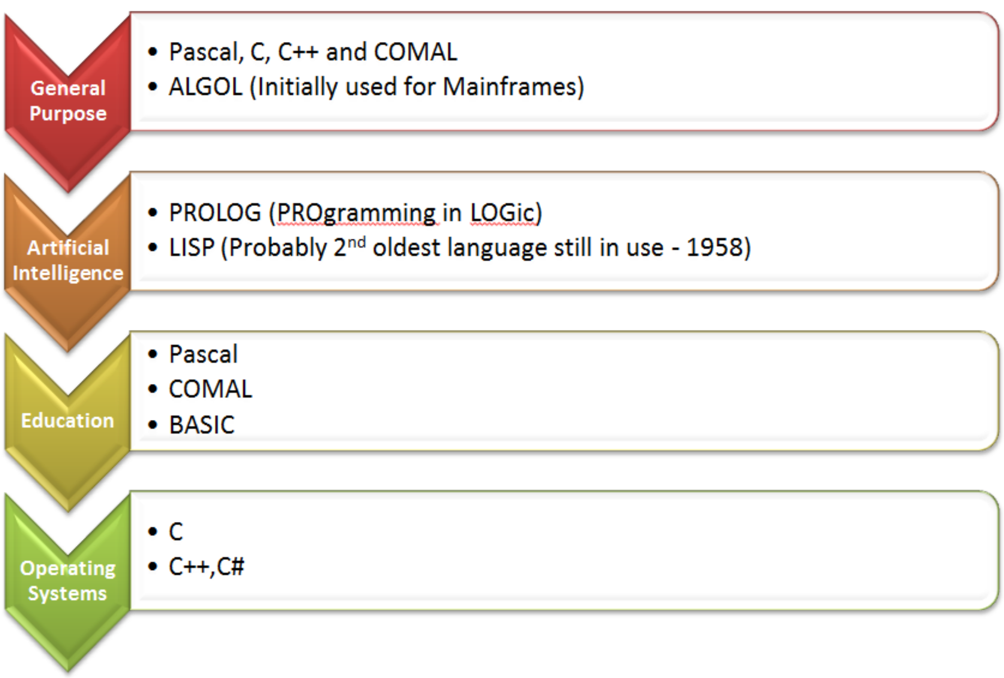
Types of Languages

Machine Code is made up of only 2 symbols.

- 1 and 0 (On and Off)
- Computers can only understand 1's and 0's (binary)

High Level Languages such as Visual Basic are written using some English words.

- They are not written IN English
- They are designed to solve problems
- They are portable
- They need to be translated into Machine Code so the computer can execute the instructions.



High Level Languages- Why do we need them?

The source code you write is instructions for the computer. They are in a High Level Language

The computer needs these instructions translated into machine code

It's the **only** language it understands!

Types of Translators

These programs convert High Level Language Source code into machine code (binary)

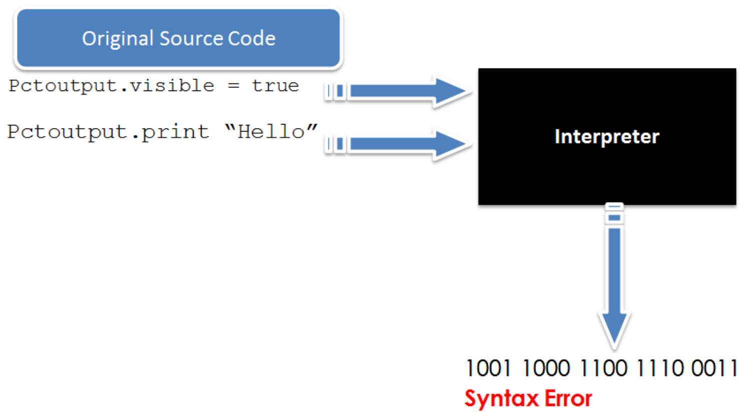
1. Interpreter

- Functions like a spoken language interpreter.
- Translates and executes a **single** line at a time
- The interpreter is **ALWAYS** loaded in memory
- Error feedback is provided **line by line**
- Interpreted programs have to be re-interpreted every single execution

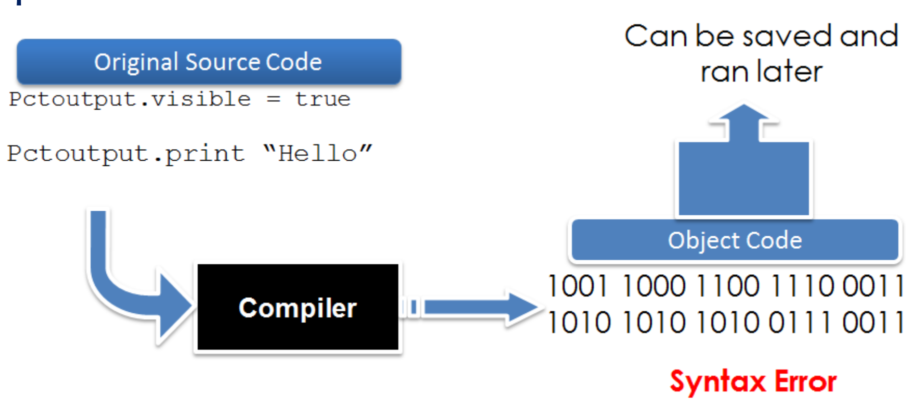
2. Compiler

- Translates and executes the **entire program** at once
- This program can then be ran repeatedly

How the interpreter works?



How the compiler works?



Interpreter V Compiler – Summary

	Compiler	**Interpreter**
Run Time	Run time is fast as the code is already translated. The object code program is run so it does not need to be translated as it's al-ready in machine code.	Run time is slower as the interpreter must translate each instruction in turn, every time the program is run. Code inside a loop must be translated and executed the number of times the loop repeats. For a loop repeating 10 times each instruction inside the loop needs translated and executed 10 times.
Ease of fixing errors	Program will not compile if it has errors. Error report is produced after it tries to compile the code. The errors must be found and fixed in the HLL code and then re-compiled. More errors could result from this.	The program will run successfully up until the error is encountered. This allows partial code to be test-ed and allows the programmer to see the program improve. Syntax errors are highlighted immediately allowing them to be fixed.
Use of RAM	The compiler software can be removed from RAM once the code is compiled, this saves RAM in the future as only the object code program is needed in RAM to run the program. The resultant object code can be larger as RAM is not being used up storing the compiler software.	The interpreter software can never be removed from RAM, it must be present every time the pro-gram is run. This means the resultant HLL code must be smaller as it cannot use up the full amount of RAM.
When should I use each?	Use compiler for the final version of a pro-gram as it contains no errors and run time is fast.	Use when developing a program as it highlights error immediately making them easier to find and fix, and allows partial code to be run so you can see the result of coding up to that point in the program.

Environmental Impact

When we are using a computer, we are using electricity, not only for the computer, but also the monitor, printer, wireless router etc. This energy use can have a detrimental impact on the environment.

However, computers reduce environmental impacts.

For example, sending an e-mail to USA.

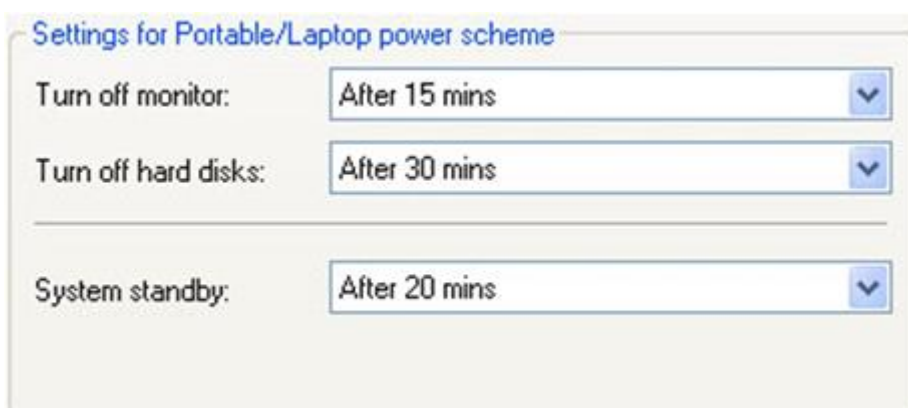
Obviously while using the computer to send the e-mail, you are using up energy and when your computer was being made a lot of energy would have been used and a lot of carbon dioxide produced.

However, consider the alternative: trees would be cut down to produce the paper to write on. The letter would be picked up by a post office van burning fuel. It would be taken to the sorting office, then driven by van to the airport, more fuel. The plane would then burn lots of fuel taking it to USA where once again it would be loaded onto vans, driven to their sorting offices, then driven to homes using an awful lot of fuel.

Reducing Computer's Environmental Impact

Some of the ways we can reduce the environmental impact is by:

1. ***Leaving computers on stand-by***
   If you are going to be using the computer intermittently, then set the stand-by mode to come on after 20 minutes of inactivity. This will reduce the power consumption quite considerably.

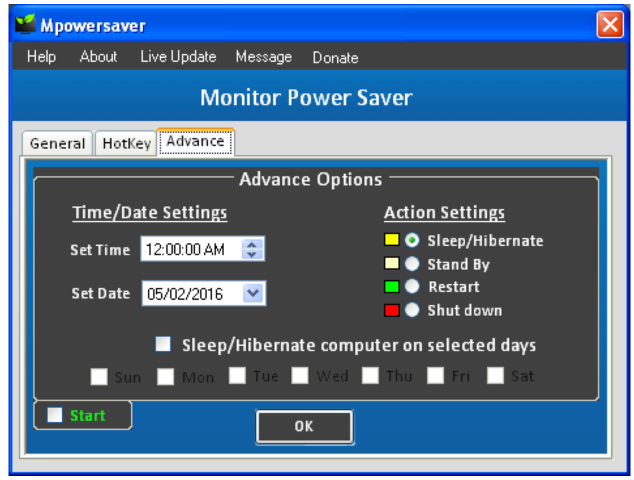


2. ***Choose the settings on your monitor carefully***

   The monitor settings can also save quite a lot of energy. To reduce the energy impact:

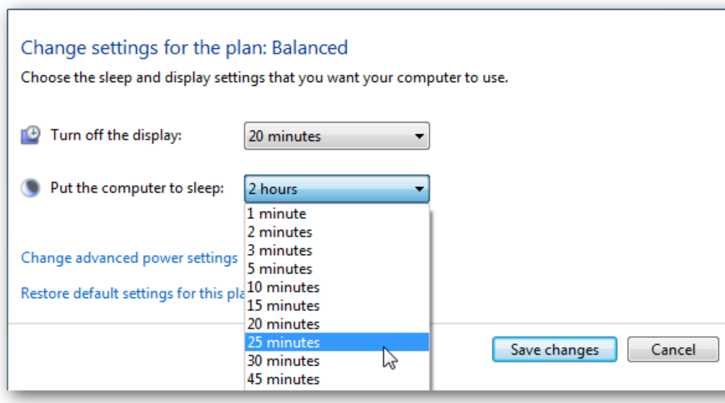
   a. Set the monitor to sleep/stand-by after 20 minutes of inactivity will save a lot of energy.

   b. Setting the brightness of the monitor to a dimmer setting will save power. (Having the monitor set to very bright will use more power than having it set to a dimmer setting).



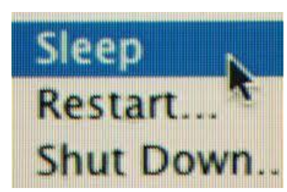
3. ***Power Down Settings***

   You are able to choose the settings that will power down the computer.



   a. Changing the settings to a shorter time of inactivity should save energy (but may become a nuisance if you set it for too short a time).

   b. It was suggested that 50% of the power used by a computer is when the user is not using it! For example, leaving it on all night. You can set a time for the computer to shut down and switch on. (The school computers switch off at 18.30 and switch back on at 07.00).



Using all three of these methods should reduce your power usage and hence the environmental impact of using your computers.

Security Precautions

There are a large number of threats to your computer and the information stored on it. For example: viruses, hacking, phishing etc.

To try and protect against threats like these there are some basic precautions you should take:

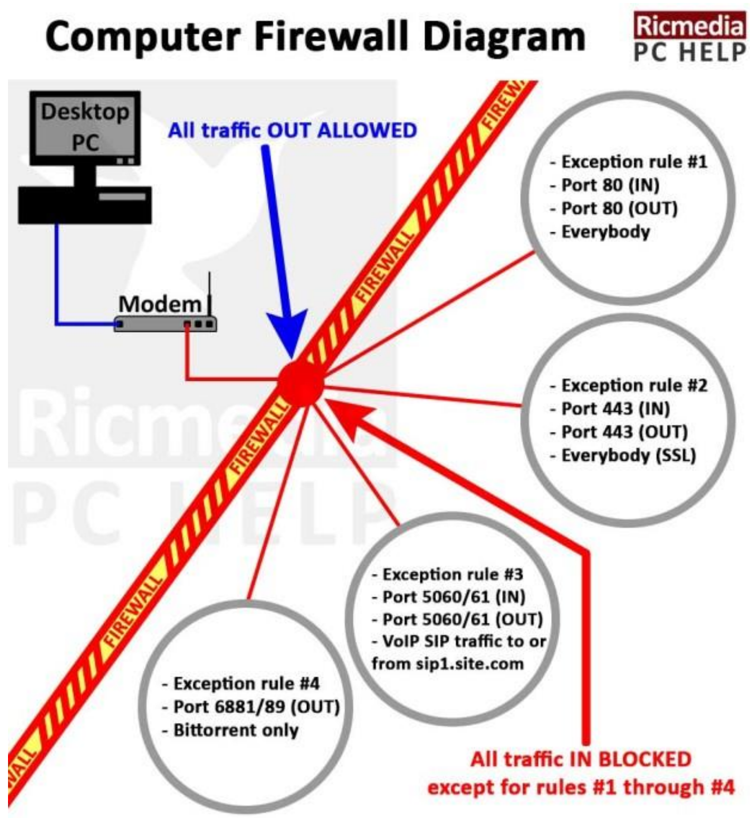
Firewall

A firewall protects against people trying to hack into your computer. It does this by:

When an external computer tries to access your computer, the firewall examines whether it is allowed to access your computer. If it is, then access is given. If not then it is blocked.

Ways in which a firewall can block access

- ***Packet filtering***
  When a data packet is received by the firewall, it checks it against its filters. If it is acceptable it is allowed through otherwise it is rejected.
- ***IP Blocking***
  When a specific IP address tries to communicate with your computer, the firewall can compare the IP address that it came from against a list of blocked IP addresses. If it is in the blocked list then it is rejected (alternatively it can be checked against a list of IP addresses that are allowed to access the computer).

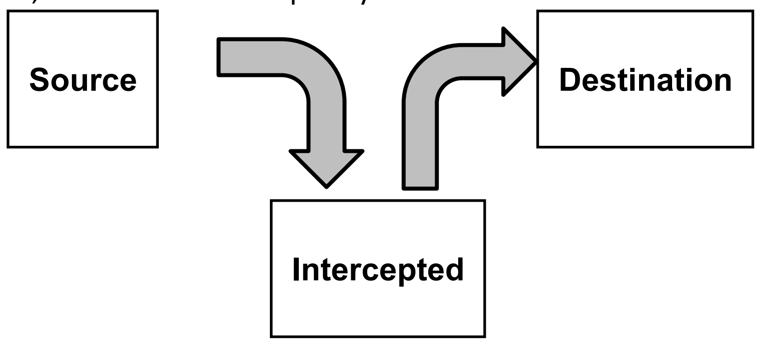


Encryption

More and more information is being transferred electronically. Much of this information is private and confidential and if found out by criminals could be used against us.

For example, when shopping on the Internet, if someone intercepted your credit card details then they would be able to buy goods using your credit card details.

One solution to this problem is to encrypt the information before it is sent electronically.



Encryption is the process of changing data so that it cannot be understood by a third party. **Decryption** is the reverse.

Data is **scrambled** using a mathematical process which turns it into something that looks like nonsense. This means that if anyone steals the information it will be meaningless to them.

Encrypted data is known as **ciphertext**. Ciphertext cannot be read without first being decrypted.

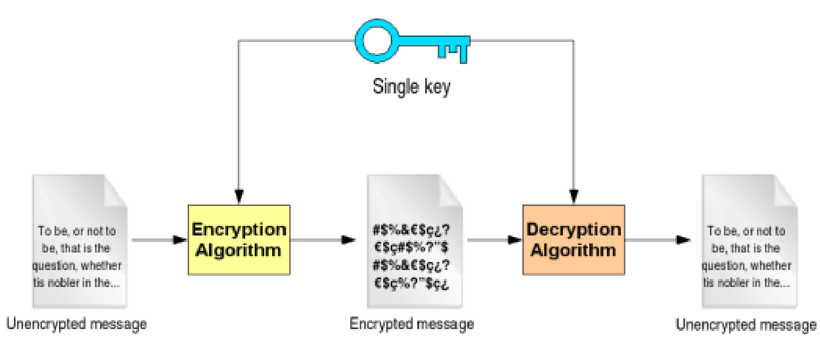
<u>**How encryption works**</u>

Data (plain text) is encrypted using a secret key and encryption algorithm. Both parties must have a copy of the secret key which must also be kept secure.

Encryption Key

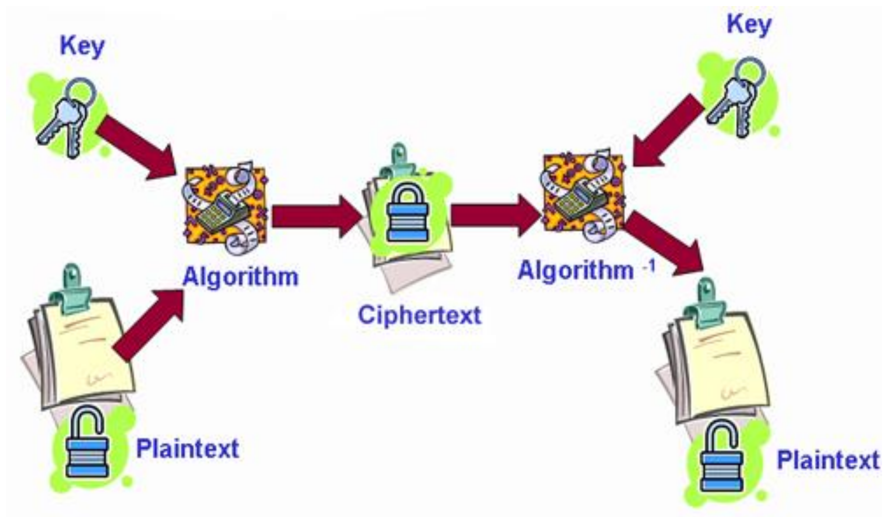
A **key** is a long sequence of bits used by encryption / decryption algorithms.

To crack some ciphertext encrypted with a 64-bit key by trying every combination of keys possible means you have 2^64 possible combinations (18 followed by 18 naughts).



If you have a computer that can carry out one encryption operation every millisecond, it will take about 292 million years to find the correct value.

Plain text is combined with the secret key and encryption algorithm to produce ciphertext.



Ciphertext is then combined with the secret key and the decryption algorithm to produce plain text.