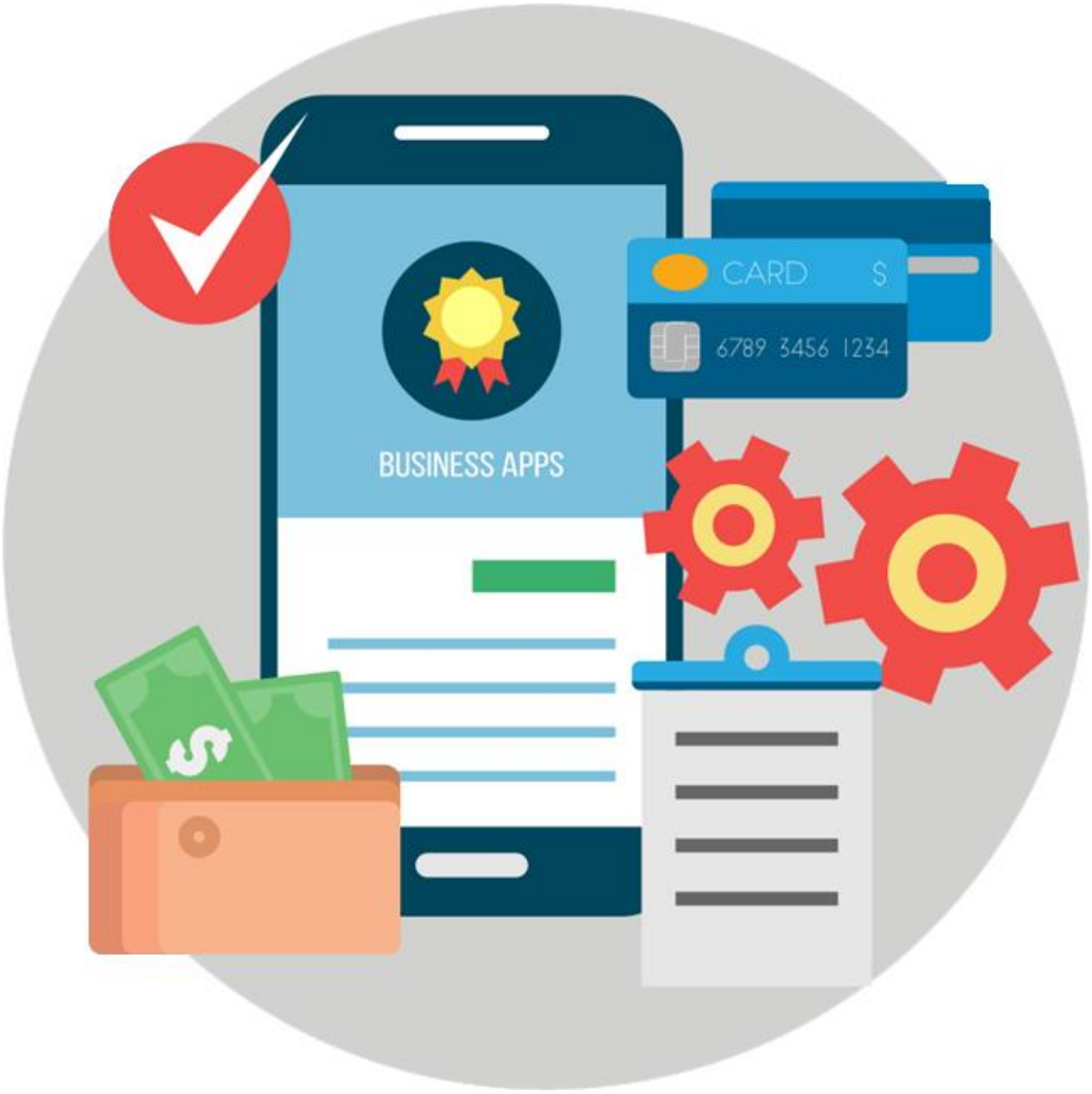
# AQA GCSE Computer Science (8525)

## Paper 2: Computing Concepts




# Revision Booklet

# Topic 3: Fundamentals of Data Representation

## Number Bases

| Decimal (Base 10) | Binary (Base 2) | Hexadecimal (Base 16) |
|---|---|---|
| This is the number system we are most used to. | Used by computers to represent all data. | Used to represent larger numbers. |
| 10 characters, 0-9 | 2 characters, 0 and 1 | 16 characters, 0-9 and A - F |
| 1, 2, 20, 40, 999 | 0, 000, 01010, 00101010 | 1, 6, A, AF, 1A, AB1E8 |

**Binary**

Used by computers due to the transistors in the CPU, which can be either on or off.
Data stored in binary can represent many things such as text, numbers, pictures or sound.

**Hexadecimal**

Uses the characters 0-9 and A – F.
Allows large numbers to be more easily represented than in binary.
Used for colour values and MAC Addresses.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| B | 1 | 10 | 11 | 100 | 101 | 110 | 111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 | 10000 |
| H | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 |

## Converting Binary, Denary and Hexadecimal

### Binary to Denary

1) Draw your conversion table.
2) Write the binary number in the conversion table.
3) Add together all numbers with a 1 beneath them

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

128
64
\+ 8
4
**204**

**11001100 in binary is 204 in denary**


### Denary to Binary

1) Draw your conversion table.
2) Is the number higher than the first column in the table?
   a) If so, put a 1 in that column and work out the difference.
   b) If not, put a 0 in that column.
3) Repeat the step above with the difference.
4) Keep going until the difference is 0, put a 0 in any empty columns.
5) Read the number from the bottom row of the table.

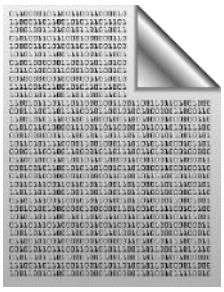
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

200 – 128 = **72**

72 – 64 = **8**

8 – 8 = **0**

**200 in denary is 11001000 in binary**



### Denary to Hexadecimal

1) Divide the denary number by 16 and write down both the answer and the remainder.
2) Divide the answer by 16 again. Write down both the answer and the remainder.
3) Keep going until you reach an answer of 0.
4) Read the remainders from bottom to top.
5) Convert each remainder to hex.

$62 \div 16 = 3 \text{ R } 14$

$3 \div 16 = 0 \text{ R } 3$

3 14

3E

**62 in binary is 3E in hexadecimal**

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | A |
| 11 | B |
| 12 | C |
| 13 | D |
| 14 | E |
| 15 | F |

### Binary to Hexadecimal

1) Draw two separate conversion tables.
2) Write the binary number across both tables.
3) For each table, add up the numbers which have a 1 beneath them.
4) Convert each number to hexadecimal.

| 8 | 4 | 2 | 1 |
|---|---|---|---|
| 0 | 1 | 1 | 0 |

| 8 | 4 | 2 | 1 |
|---|---|---|---|
| 1 | 1 | 0 | 1 |

$4 + 2 = 6$

$8 + 4 + 1 = 13$

6

D

**01101101 in binary is 6D in hexadecimal**

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | A |
| 11 | B |
| 12 | C |
| 13 | D |
| 14 | E |
| 15 | F |

## Units for Measuring of Data

A bit is the most basic unit of data measurement.
A bit can be 0 or 1.
A byte is 8 bits.
B represents a byte, and b represents a bit

| Bytes | | Bits | |
|---|---|---|---|
| **Size** | **Unit** | **Size** | **Unit** |
| 1,024 bytes | 1 kilobyte - KB | 1,024 bits | 1 kilobyte - KB |
| 1,024 kilobytes - 1,024 KB | 1 megabyte - MB | 1,024 kilobits - 1,024 KB | 1 megabyte - MB |
| 1,024 megabytes - 1,024 MB | 1 gigabyte - GB | 1,024 megabits - 1,024 MB | 1 gigabyte - GB |
| 1,024 gigabytes - 1,024 GB | 1 terabyte - TB | 1,024 gigabits - 1,024 GB | 1 terabyte - TB |

## Binary Addition

Always follow these for rules:

- 0 + 0 = 0
- 1 + 0 = 1
- 1 + 1 = 10 (binary for denary 2)
- 1 + 1 + 1 = 11 (binary for denary 3)

Example - adding 01 + 101
1 + 1 = 0, carry 1
1 + 0 + 0 = 1
0 + 1 = 1

| | | | |
|---|---|---|---|
| | | 0 | 1 |
| | 1 | 0 | 1 |
| **1** | **1** | **0** | |
| | | 1 | |

## Binary Shifts *- a process used to multiply or divide binary numbers*

Example: 00010101 X 4 = 01010100

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

**Multiplication**

To multiply move the digits to the left and fill the gaps after the shift with 0:

- To multiply by two, all digits shift one place to the left
- to multiply by four, all digits shift two places to the left
- etc.

Example: 00100110 ÷ 2 = 00010011

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

**Division**

To divide move the digits to the right and fill the gaps after the shift with 0:

- To divide by two, all digits shift one place to the right
- To divide by four, all digits shift two places to the right
- etc.

## Character Encoding

Because computers work in binary, all characters must be stored as binary numbers.
The characters which a computer can use are called a **Character Set**.
A character code is the number assigned to a character within a character set.
When storing and transmitting characters, the computer will use the character code.
Character codes are grouped together, and run in order within that group.
For example, in ASCII A is 65, B is 66 and so on.
This makes it easy to calculate different character codes.

**ASCII (American Standard Code for Information Interchange)**

Ascii uses 7 bits, giving a character set of 128 characters.
These are represented the ASCII Table.
Each character has its own assigned numbers, some examples are below.
Included in the table are:

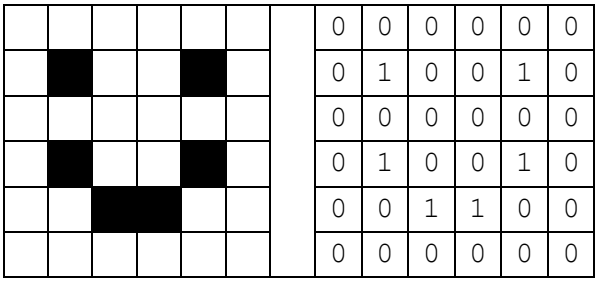
- 32 control codes - mainly to do with printing
- 32 punctuation codes, symbols, and space
- 26 upper case letters
- 26 lower case letters
- Numbers 0-9

| Character | ASCII Decimal | ASCII Binary | ASCII Hexadecimal |
|---|---|---|---|
| **A** | 65 | 1000001 | 41 |
| **Z** | 90 | 1011010 | 5A |
| **a** | 97 | 1100001 | 61 |
| **z** | 122 | 1111010 | 7A |
| **0** | 48 | 110000 | 30 |
| **9** | 57 | 111001 | 39 |
| **Space** | 32 | 100000 | 20 |
| **!** | 33 | 100001 | 21 |

**Unicode**
The ASCII Character set is too small to hold every character and symbol in English and other languages such as Chinese and Arabic.
Unicode uses 16 bits, giving a character set of 65,536 characters.
Unicode uses the same character codes as ASCII up to 127.
Unicode also includes additional symbols and characters such as emojis.

## Representing Images using Bitmaps

Images are broken down using a grid.
Each square in the grid is known as a pixel.
Pixel is short for **Picture Element**.



6 x 6 image, one bit colour
12 pixel resolution, 12 bit file size

**Colour Depth**
The more colours an image uses, the more bits per pixel are used. This is called the Colour Depth.
The greater the colour depth, the larger the image file will be.

- one bit per pixel (0 or 1) - two possible colours
- two bits per pixel (00 to 11) - four possible colours
- three bits per pixel (000 to 111) - eight possible colours

**Image Size**
The size of an image is measured in the number of pixels used.
This is written as pixels wide x pixels high.
For example, 5x5, 20x40, or 1024x768.

**File Size**
The file size of an image in bits can be calculated using the formula :
**height in pixels X width in pixels X colour depth per pixel**
Dividing the result by 8 will give us the value in bytes

Example:
- An image 400 high, 500 wide with 16 bit colour depth
- 400 x 500 x 15 = 3,200,000
- 3,200,000 ÷ 8 = 400,000 bytes
- 400,000 ÷ 1000 = 400 kilobytes

## Representing Sound in Binary

Sound is analogue, but must be converted to binary before computers can understand it.
We do this by measuring the amplitude, or volume, of the sound at a given point in time. This is called sampling.
Many samples are put together to represent the sound.

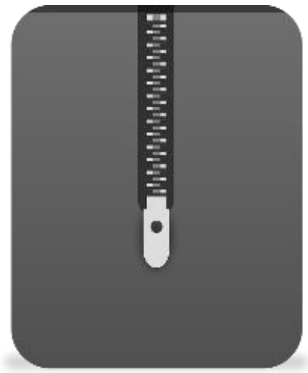
**Amplitude -** The height of the sound wave at the time it was sampled.
The higher the amplitude the louder the sound.
**Sample Rate**
The sample rate is the number of samples taken in one second.
It is measured in Hertz (Hz).
1 Hertz = 1 sample per second.
Small sampling interval = high sample rate = better quality sound file = larger file.

**Sample Resolution**
The sample resolution is the number of bits used to store each sample.
The higher the sample rate the more accurate the representation is, but the more space needed.

**Calculating Sound File Sizes**
The elements above can be used to calculate the file size of a sound file.
Larger files will give a more accurate representation of the original sound.

File size (bits) = rate x res x secs
rate = sampling rate
res = sample resolution
secs = number of seconds

## Compression – *ways to reduce the amount of storage space required for data*



Large files are difficult and expensive to store and transmit.
Compression techniques reduce file sizes.
There are many different ways to compress data, each with their own pros and cons.
It is important to balance the reduction in file size with any reduction in quality.
Different compression techniques will work best in different scenarios.
Lossy compression means that data is lost and can not be recovered once the file is compressed.
Lossless means that no data is lost and the original contents of the file can be completely recovered.

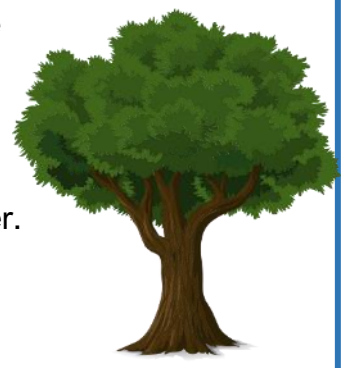
## Huffman Coding – *a way to reduce the number of bits needed to send or store a message*

It is a lossless compression method.
It looks at how often a data item, for example a character in a string, occurs.
It tries to use fewer bits to store common data which frequently occurs.
It is comprised of:

- A Huffman tree, giving each character a unique code
- A binary stream of the character sequence

**Creating a Huffman tree**

1. Count how many times each character appears in the string.
    a. Write down the list in order with the most common letter at the top.
    b. Characters with the same value can be placed in any order.
2. Add together the number of the two least common characters in a new block and label it with the total.
3. Move the new block into the right place in the list based on its number.
4. Repeat steps 2 and 3 until only one block remains.
5. Labe the “branches” in the tree, working from the top down.
    a. Label all the branches going in one direction 1 and the other 0



**Encoding the binary stream**
The tree can be compressed into a single string.

1. Start at the top node.
2. Encode each letter using the path of 1s and 0s.
3. Join the stream together

**Calculating the  Bits Needed to Store Hoffmann Compressed Strings**

1. Take the length of the bit pattern for each character.
2. Multiply it by the times the pattern is used.
3. Add this together for each character.

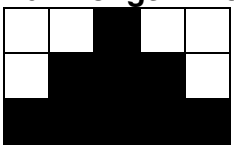
**Calculating the Bits Needed to Store Uncompressed ASCII**

Number of characters x 7
e.g. hello, 5 characters, 5 x 7 = 35

## Run Length Encoding (RLE)

It is a lossless compression method.
It relies on the original data having many repeating digits.
It finds patterns in the original data to save space.
**Runs** of data are sequenced within the original data which have the same value.
These runs are stored in **frequency/data** pairs.

**Run Length Encoding Example**



Binary code = 00100 01110 11111

RLE Pairs = 2 0 1 1 3 0 3 1 1 0 5 1

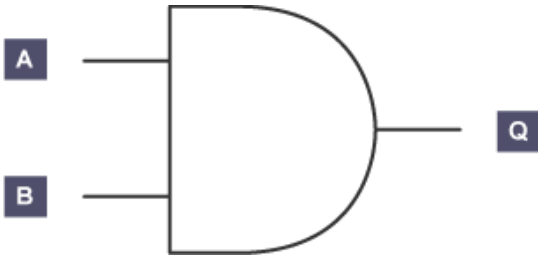
# Topic 3: Computers

## Hardware and Software


**Hardware** is the physical components of a computer, such as the monitor, CPU, RAM or keyboard.
**Software** is the programs and applications which run on the computer such as the operating system, games or a word processor.

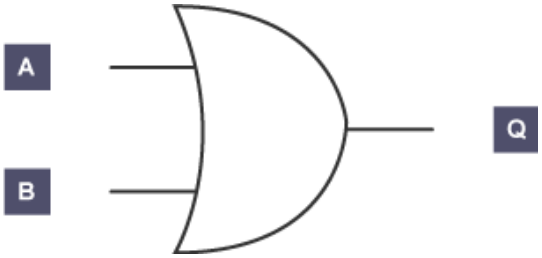
## Boolean Operators

**AND** - two conditions must be met for the statement to be true
Written as AND or **.**



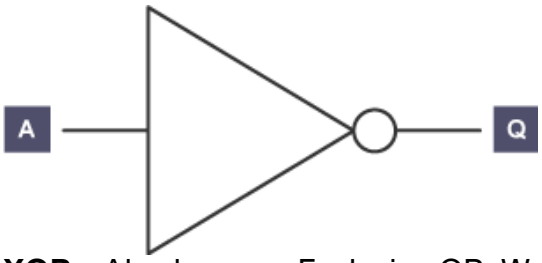

| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**OR** - at least one condition must be met for the statement to be true
Written as OR or +



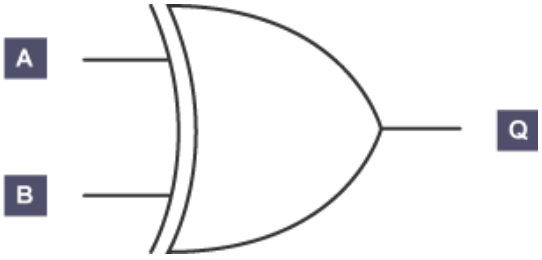

| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**NOT** – inverts the result, e.g. NOT(A AND B) will only be false when both A and B are true
Written as NOT or $^{-}$




| A | Q |
|---|---|
| 1 | 0 |
| 0 | 1 |

**XOR –** Also know as Exclusive OR. Works the same as an OR gate, but will output 1 only if one or the other and not both inputs are 1.
Written as XOR or $\oplus$




| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

## Trace Tables

A method of recording the values used within an algorithm at each stage of processing to help in troubleshooting

- Tests algorithms for logic errors which occur when the algorithm is executed.
- Simulates the steps of algorithm.
- Each stage is executed individually allowing inputs, outputs, variables, and processes to be checked for the correct value at each stage.
- A great way to spot errors

```
X = 3
Y = 1
while X > 0
      Y = Y + 1
      X = X - 1
print(Y)
```

| Stage | X | Y | Output |
|---|---|---|---|
| 1 | 3 | 1 | |
| 2 | | 2 | |
| 3 | 2 | | |
| 4 | | 3 | |
| 5 | 1 | | |
| 6 | | 4 | |
| 7 | 0 | | |
| 8 | | | 4 |

## System Software

Systems software manages the computer system including:

- Controlling hardware, including peripherals
- Allowing other programs to run
- Providing an interface for the user to interact with the computer
- Maintaining the computer system

Operating Systems and Utility Software are the two main types of system software.

**Operating Systems**

Examples include Microsoft Windows, macOS, Linux, Android, iOS

The Operating system provides a foundation for the user to interact with the computer and for other applications to run.

It handles management of key hardware and functions including processors, memory, input/output (I/O) devices, applications and security.

It does so by performing a number of key functions explained below.

<u>File management</u>

Allows users to find and manage data stored by the computer.

Data is stored in files, within folders, within drives.

Uses a virtual file structure of the physical components.

Assigns metadata to files including date created, date modified, last date accessed

<u>Process management</u>

Allows users to run applications such as web browsers or word processors.

Multiprogramming enables several programs to run at the same time.

Each program is made up of program instructions. When these instructions are running, they are called a process.

Allocates use of the main memory and the CPU between processes.

A scheduler is used to time the different processes.

<u>Peripheral Management</u>

Manages input and output between peripherals and a process.

Data is transferred between input devices, the CPU and main memory, and output devices.

Uses device drivers to communicate with devices.

User Management
Individual users can be created and deleted.
Allows more than one person to use  a computer with their own files and settings.
Access levels control user access to systems for security.
A log is kept of files a user creates, accesses, edits and deletes

**Utility Software**
Utility Software helps with the upkeep and maintenance of the system.
A computer will often run several different pieces of utility software performing one or more of the tasks below.

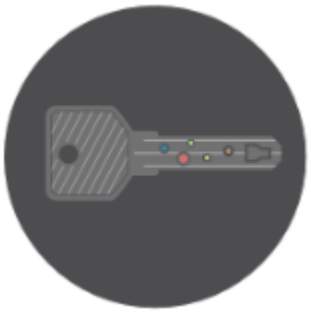
File Repair
Files can become corrupt due to crashes, damaged storage or a virus.
Corrupt files can sometimes be repaired.
Can detect and recover physical errors on the disk.
Can scans the disk surface for damage and mark sections as unavailable.

Backups
Data can be lost accidentally or deliberately.
A copy of data is known as a backup.
These allow damaged or deleted data to be restored.
Full backups include every file. This requires a lot of storage and time.
Incremental backups include new and changed files since the last backup.

Data Compression
Reduces the size of a file using algorithms.
Smaller files are easier to transmit.
Allows more files to be stored in the same space.
Lossless - no data is lost, and the original can be recreated.
Lossy - some data is lost, and the original file cannot be recreated.

Defragmentation
Files on a disk are broken down into a series of segments.
When files are deleted, the segments where they were stored are made available for new files.
The new file may need more segments than the old, and so the segments allocated to it are not together on the disk. This is known as fragmentation.
A fragmented disk takes longer to read from and write to, making the computer slower.
Defragmentation software rearranges the segments so that they are stored next to each other.
This decrease read/write time and improves performance.

Anti-Malware
Protects against viruses, spyware, and other unwanted software.
Scans the system to identifies potential viruses.
Will attempt to delete or fix potential threats once they have been identified.
Runs either when activated or automatically at a specified date and time.



## Low Level Programming Languages – *very close to computer language, hard for humans to understand*

**Machine code**
Processors understand machine code can directly execute it.
Each type of processor has its own specific machine code.
Consists of 0s and 1s (binary) only.
Very difficult to learn, write and debug.

**Assembly Language**
Also known as Assembly Code
Uses mnemonics (abbreviations)
Easier for humans to understand and program but still difficult
Must be translated into Machine Code for execution
Has a 1:1 relationship with machine code
Commonly used to program device drivers
Often used in embedded systems

**Advantages of Low Level Programming Languages**

- More tasks are possible using low level programming languages.
- More flexible.
- Programmers can instruct the processor directly to do whatever is needed,
- Less limited than high level languages.
- More efficient than high level languages

**Disadvantages of Low Level Programming Languages**

- Very difficult to write and understand.
- Much more time consuming to produce code.
- Much harder to debug and fix problems.
- Machine code is specific to a single processor, so code which works on one processor will not work on another

## High Level Programming Languages – *easier for humans to understand, using English like words and phrases.*

Much easier to learn, write and debug.
Examples include Python, Java and C
Code written in these languages must be translated to machine code before it can be executed.

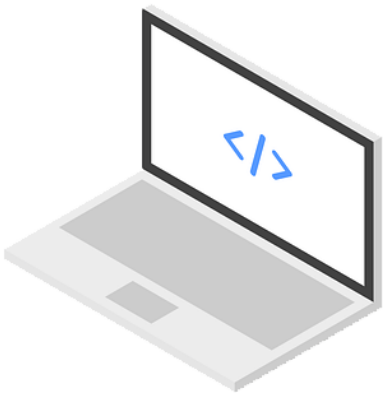
**Advantages of High Level Programming Languages**

- Much more widely understood and used.
- Easier to learn, code in and understand.
- Much quicker to produce usable code.
- More support and learning resources are available.
- Easier to debug and find issues

**Disadvantages of High Level Programming Languages**

- Less flexible.
- Must be translated before being executed
- Very difficult to write and understand.
- Much more time consuming to produce code.

# Assemblers Compilers and Interpreters



**Compilers**
Translates the whole code in one go into Machine Code.
Optimise the code
Used at the end of development when code is finished
Create error reports and object code

**Interpreters**
Does not generate machine code directly.
Calls machine code subroutines using their own code.
Translate and execute source code
Work line by line.
Syntax is checked
If code is correct it is executed
If code is incorrect interpreting is stopped.

**Assemblers**
Translates assembly language into machine code.
Create one machine code instruction for each assembly instruction.

## CPU – *Central Processing Unit*

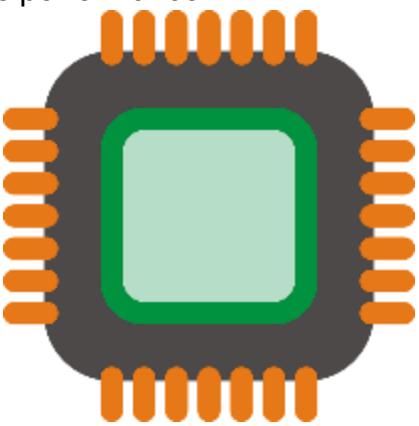
The most important hardware component.
Has two main functions:

- to process data and instructions
- to control the rest of the computer system

Different things affect CPU Performance:

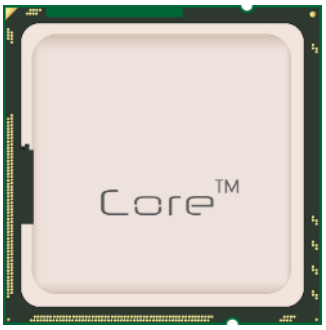
- Clock Speed – The faster the clock speed the quicker the CPU can execute instructions and so the better the performance.
- Number of Processor Cores – The more cores a CPU has, the more executions it can execute at the same time and so the better the performance.
- Cache Size – The greater the cache available to the CPU, the more results it can store. This means less instructions need to be repeated and so improves performance.

The CPU consists of several components:



**Control Unit (CU)**
Fetches, decodes, and manages the execution of instructions
Issues control signals to control hardware
Moves data around the system
**Arithmetic Logic Unit (ALU)**
Performs arithmetic and logical operations.
Where calculations are done and where decisions are made.
**Registers**
Small amounts of high speed memory in the CPU.
Used to store small amounts of data that are needed during processing.
**Cache**
A small amount of high speed memory In the CPU.
Used to temporarily hold data the CPU will reuse.
Allows for faster processing since as the CPU need not wait for data to be fetched from RAM.

**Clock**




Used to coordinate all the computer's components.
Sends out a regular electrical pulse to do this.
The frequency of the pulses = clock speed, measured in hertz.
Higher clock speed = greater number of instructions which can be performed at a time.

**Buses**

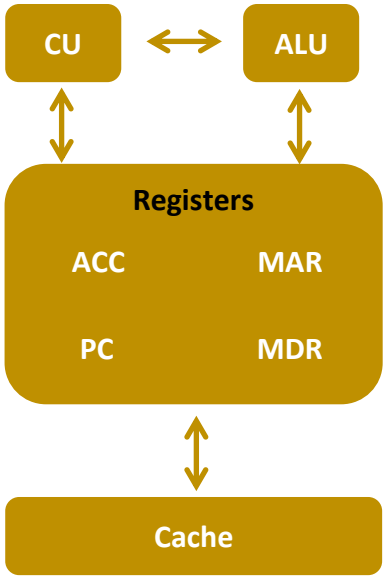
High speed internal connections.
Used to send control signals and data between the processor and other components.

- Address bus - carries memory addresses from the CPU to other components.
- Data bus - carries data between the CPU and other components.
- Control bus - carries control signals from the CPU to other components.

## Von Neumann Architecture - *the design on which most computers are based. Uses the stored program concept.*

Key elements:

- Data and instructions are stored in binary.
- Data and instructions are stored together in **RAM**.
- Instructions are fetched from **RAM** one at a time in order
- The **CPU** decodes and executes an instruction, before fetching the next instruction
- The cycle continues until no more instructions are available




A CPU using Von Neumann architecture have five special registers

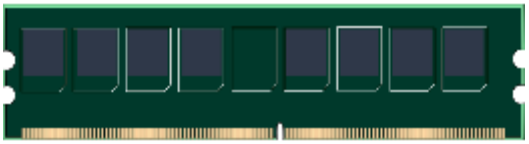
- **Program counter** - holds the memory address of the next instruction to be fetched.
- **Memory address register (MAR)** - holds the address of the current instruction.
- **Memory data register (MDR)** - holds the content at the address held in the MAR.
- **Current instruction register (CIR)** - holds the instruction that is currently being decoded and executed
- **Accumulator (ACC)** - holds the results of processing

## The fetch-decode-execute cycle - *followed by a CPU to process an instruction.*

There are seven steps:

1. The memory address held in the program counter is copied into the MAR.
2. The address in the program counter is then incremented - or increased - by one. The program counter now holds the address of the next instruction to be fetched.
3. The processor sends a signal containing the address of the instruction to be fetched along the address bus to the computer's memory.
4. The instruction held in that memory address is sent along the data bus to the MDR.
5. The instruction held in the MDR is copied into the CIR.
6. The instruction held in the CIR is decoded and then executed. The results of processing are stored in the ACC.
7. The cycle then returns to step one.

## RAM - *Random Access Memory*



Volatile memory – contents is lost when the computer is turned off.
Called random access because data can be directly written to or read from any location.
Used to hold data and instructions that are currently in use.
The more RAM a computer has, the more data it can hold simultaneously.

## ROM – *Read Only Memory*

Non-volatile main memory – contents are not lost when the computer is turned off.
Can be read from, but not written to.
Ideal for storing instructions and data that are needed for the computer to run.
Usually programmed by the computer's manufacturer and cannot be overwritten.
The **Basic Input Output System (BIOS)** is an example of a program stored in ROM.

## Main Memory vs Secondary Storage

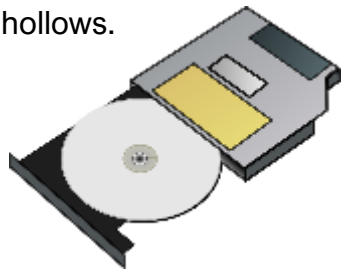
Main memory is directly accessible by the CPU.
Secondary storage is not directly accessible by the CPU.

## Secondary Storage – *used to store programs and data for longer term when the computer is switched off*

Non-volatile – data is retained with the computer is switched off.
Stores the operating system, software and other files needed for the computer to operate
Stores the applications used on the computer.

**Magnetic devices**
Use magnetic fields to magnetise individual sections of a spinning disk.
Each section represents one bit.
A read/write head moves across its surface.
Fairly cheap, high in capacity and durable.
Susceptible to damage if dropped.
Vulnerable to magnetic fields.

**Optical Devices**
Use a laser to scan the surface of a spinning disc.
The disc surface is divided into tracks, with each track containing flats and hollows.
The flat areas are known as lands and the hollows as pits.
Lands reflects the laser light back; pits scatter the beam.
ROM (Read Only Media) cannot be overwritten. Used for music, films, software and games.
Read (R) media is blank, can only be written to once, but read many times.
Read/write (RW) media can be written to more than once.



**Solid State Devices**
Use flash memory to store data indefinitely.
Have faster access times than other devices
Because they have no moving parts, are more durable.
More expensive so tend to be smaller in capacity.
Require little power, so used where battery life is a consideration.
Portable due to their small size and durability.

## Cloud Storage - *storing data at a remote location online*

Remote storage over The Internet.
Files are stored on a server hosted by a cloud storage provider in a secure datacentre.
The provider will store files on magnetic or solid state storage.
Users do not know physically where their data is stored.
Some providers offer the option to choose a specific country or continent to store data in.
Files can be uploaded and downloaded as required from anywhere with internet access.

**Advantages**

- Provides the ability to access files from any location or device with Internet access.
- Files can easily be shared with other users.
- Cloud storage services often back up data for their users.
- Provides increased security as data is stored in high security data centres.
- Allows small devices such as phones or tablets to access huge amounts of data which it would not be practical to store directly on them.
- More storage can be easily added without having to buy and install more hardware.

**Disadvantages**

- Can be targeted by hackers.
- Data is stored by someone else, so the owner has less control.
- Data can only be accessed in locations with a working internet connection.

## Embedded Systems *– a small computer which includes hardware and software, designed to control a specific device.*

Forms a part of a larger device such as a washing machine.
Can perform only a limited number of tasks.
Have several advantages:

- Cheaper to design and build.
- Require less power.
- Do not need much processing power.
- Less susceptible to viruses.

Have several disadvantages:

- Much more limited in function.
- Adding functionality requires the system to be rewritten.
- Requires specialist skills to build and update.

**Examples of Embedded Systems**
ATM, Watch, Calculator

**Examples of Non Embedded Systems**
Desktop computer, server, Apple Mac

# Topic 4: Networks

**Network** – Two or more computers *connected together for the purpose of communication*

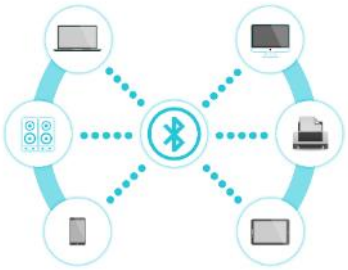
**Advantages**

- Software and files can be shared.
- Hardware such as printers can be shared
- Users can communicate via email, chat, etc.
- Centralised maintenance and updates.
- Centralised security.
- User monitoring.
- Different users can be given different access rights or permissions.

**Disadvantages**

- Cost, additional equipment is needed.
- Additional management by specialist staff.
- Spread of malware.
- Potential for hacking.

**PAN** – Personal Area Network – personal devices connected by Bluetooth



Spread over a very small area.
Used to connect personal devices e.g. smartphone and wireless headphones.
Uses Bluetooth to connect devices.

**LAN** – Local Area Network – Confined to a single location, owned and maintained by a single organisation

Used by organisation such as schools and small businesses
Connected by cables or wireless

**WAN** – Wide Area Network – covers a wide geographical area

Used by organisations with several different sites such as banks or universities
Allows all the sites to communicate and share data
Uses national or international long distance media
The Internet is the biggest example of a WAN
Can owned collectively by several organisations, for instance a group of schools

**Wired Networking** – using fibre or coper cable to connect devices in the network together.

Fibre cable provides a faster connection and can cover longer distances.
Copper cable is cheaper and easier to work with.
Advantages:

- Faster data transfer
- Less likely to suffer from interference
- More difficult for unauthorised users to intercept data

Disadvantages:

- Expensive to install or reconfigure
- Harder to move devices so less flexible

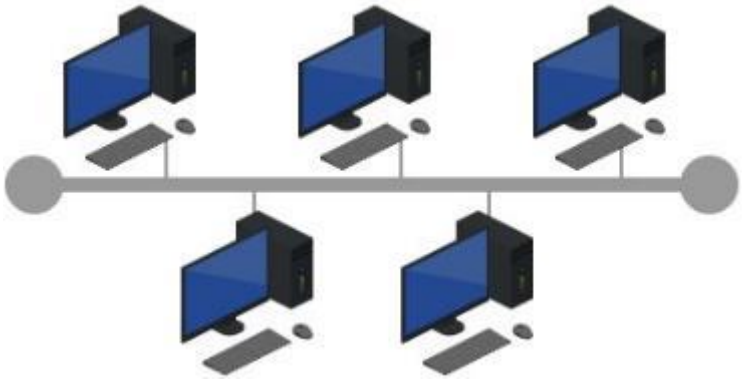
## Wireless Networking – Using radio signals or infrared light to connect devices in a network together.

Advantages
- Devices can easily be added
- Users can move around freely and stay connected

Disadvantages:
- Signals have a limited range.
- Can suffer from electromagnetic interference from other devices.
- Signals can also be blocked by walls or other objects.
- Each wireless access point (WAP) only has so much bandwidth.
- Signals can be intercepted by unauthorised users.

## Network Topologies – arrangements or methods for connecting devices together in a network.

**Bus Network**



All devices are connected to a single cable (called the bus)
A terminator is at each end of the cable.
Advantages:
- Easy to install extra devices.
- Cheap to install as it doesn't require much cable.

Disadvantages
- If the cable fails or is damaged the whole network will fail.
- Performance becomes slower ad additional devices are connected due to data collisions.
- Each device receives all data, a security risk

**Star Network**

All nodes are connected to one or more central switches.
Often used with wireless networks, where a Wireless Access Point or WAP will be the central connection
Advantages:
- Every device has its own connection so failure of one node will not affect others.
- New devices can be added by simply connecting them to the switch.
- Usually have higher performance as a message is passed only to its intended recipient.

Disadvantages:
- If the switch fails it takes out the whole network.
- Requires a lot of cable so can be expensive.

## Networking Protocols - rules for the communication of data over networks

**Ethernet** – a family of related protocols which cover how data is sent on wired networks. It is not a single protocol. The protocols include how the hardware is managed, how data is sent and received and how data collisions are handled.


**Wi-Fi** – a family of protocols which cover how data is sent through wireless connections. Wi-Fi is a trademark, the generic term for these networks is WLAN. Any device with the Wi-Fi logo uses the Wi-Fi protocols.

**TCP - Transmission Control Protocol**

Controls the sending of data.

Data is broken down into packets which are addressed and tracked through the network to make sure that they arrive at their destination.

Any packets which don't arrive are resent.

TCP is more reliable and more widely used than UDP.

**UDP – User Diagram Protocol**

Controls the sending of data however but without any tracking.

Everything is sent once, and packets which don't arrive aren't resent.

UDP it is a lot quicker than TCP and is often used in live streams where quality is less important than speed.

**IP – Internet Protocol**

Manages the addressing of packets.

Adds the sender and receiver IP addresses to each packet.

Works alongside TCP to make sure data is sent securely across The Internet.

**HTTP – Hypertext Transfer Protocol**

Responsible for transferring web pages.

Indicated by http:// at the start of a web address.

**HTTPS - Hypertext Transfer Protocol (Secure)**

An encrypted version of HTTP.

Should be used for websites which send sensitive data such as payment details or passwords.

Indicated by https:// at the start of a web address.

**FTP** – File Transfer Protocol - transmission of files across a network and The Internet.

## Email Protocols – rules for the transmission of emails which allow different email systems to talk to each other

**SMTP** – Simple Mail Transfer Protocol – used to send email.

**IMAP** – Internet Message Access Protocol – controls the download of emails from an email server into an email client application.

## Network Security

The connected nature of computer networks makes it easier for people to access data they should not be able to.

Hackers attack computer networks in an attempt to damage them or steal information.

Damage to a network or loss of data can cause people and companies to lose money and damage their reputation.

**Encryption**

Turning data into an unreadable format , requiring a key to decrypt it and make it readable again.
This means that if the data is stolen it cannot be read without the key.
Data can be encrypted before being sent over a network or when stored.
Encryption is often used alongside authentication by requiring a username and password to decrypt data and access the key.

**Authentication**

Ways to make sure a user is who they say they are.
Examples include passwords, security dongles and biometric such as fingerprints.
The most basic security feature and widely used.
Different levels of authentication are used depending on the security level needed.
Secure systems require **two-factor authentication** is now needed, which requires two forms of authentication, such as a fingerprint and password.
Allows the use of **access rights** to grant different users access to different systems or areas of a network.

**Firewall**

Monitors traffic going into and out of the network, and either allows or blocks it.
A barrier between trusted and untrusted networks.
This decision is based on rules, known as the firewall policy.
Can be hardware based or software based.
Hardware firewalls are expensive, but more effective and powerful.

**MAC Address Filtering**

All network adapters have a unique physical address known as a MAC Address.
This address cannot be changed and allows individual devices on a network to be identified easily.
Different devices can be blocked or allowed to connect to a network.

## The Four Layer TCP/IP Model

Breaks up the process for sending of messages into separate components.
Each component handles a different part of the communication.
Helps to understand the transmission process.
Provides a basis to begin troubleshooting when something goes wrong.

- **Application Layer**
  - Encodes and decodes messages.
  - Where applications such as browser and email clients operate.
  - HTTP, HTTPS, SMTP, IMAP and FTP protocols operate at this layer
- **Transport layer**
  - Manages the communication between hosts.
  - Breaks data down into packets.
  - Hosts will agree settings such as the language and size of packets.
  - TCP and UDP protocols operate at this layer.
- **Internet layer**
  - Adds the sender and recipient IP address and transmits the message.
  - Routes packets across the network.
  - IP Protocol operates at this layer
- **Data link layer**
  - Provides physical transfer of packets over the network.
  - NIC (Network Interface Card) is at this layer
  - OS device drivers are at this layer.

# Topic 5: Cyber Security

## Cyber Security

Practices, processes and technologies designed to protect computers, programs and data from attack, damage or unauthorised access.
There are an increasing number of threats to computer security.
As reliance on technology increases it becomes more important to protect computers from attack

## Pharming

Redirects website traffic to another, fake website.
May involve changing the host files on the victim's computer, or tampering with the DNS system.
May take advantage of misspent web addresses.

## Weak and Default Passwords

Using easy to guess passwords makes it easier for hackers to access systems.
Passwords such as password, 1234 or the user's name are the first hackers will try.
Passwords should contain a combination of lower and upper case characters, numbers and symbols.
Longer passwords are much harder to guess.
Many devices and applications have a default password when first setup.
These passwords are commonly known and available online.
Hackers will try these passwords first.
All passwords should be changed to a secure password when devices and software are installed.

## Misconfigured Access Rights

Sometimes users are given access to information or systems they should not have access to.
Individual user accounts allow access to be restricted to specific users.
This is not always set up correctly.
Hackers can use this to steal information or damage systems.

## Removable Media

Hackers can use removable media to steal data from systems.
Removable media can be used to install malware on systems.
Removable media is small and portable making it easy to steal.
Organisations will often prevent the use of removable media on their systems.
Data stored on removable media should be encrypted so that it cannot be easily read if the device is lost or stolen.

## Unpatched and/or Outdated Software

Software often contains bugs.
These can be serious and allow hackers to access parts of the software they should not.
Software providers release patches, containing code to fix these bugs.
If these patches aren't installed, hackers can use flaws in the system to gain unauthorised access to the information.
Old software often is no longer supported by its developers and so will not receive patches.
This means security flaws can go unfixed for long periods of time or even forever.

## Penetration Testing

The process of attempting to gain access to a system without knowing usernames, passwords or other normal ways to access it.
Useful to test systems to identify where weaknesses are .

**White-Box Penetration Testing**
Simulates a malicious insider with knowledge of and/or basic credentials for the target system.
The person performing the test is given some information about how the system works.
They use this to identify possible holes prior to starting the testing.

**Black-Box Penetration Testing**
Simulates an external hacking or cyber warfare attack
The person performing the test has no information about the system and is not given any credentials.
They look for any possible weaknesses or flaws using a trial and error approach.

## Social Engineering

Manipulating people to give up confidential information.
There are many different forms of social engineering.

**Blagging (Pretexting)**
Using an invented scenario to engage a specific victim in a way that increases the chance the victim will give out information or perform actions which would be unlikely in ordinary circumstances.
This often involves researching the target on Facebook or other social media.
Limiting the amount of personal information posted online helps prevent this.

**Shouldering (Shoulder Surfing)**
Watching someone enter person private information over their shoulder.
This could be watching someone enter their PIN at a cashpoint.
Can be done in person or using hidden cameras.
Can be prevented by being aware of one's surroundings and being cautions about where sensitive information is entered.
Screen filters on laptops can help here.

**Phishing**
Fraudulently obtaining private information from someone.
Often uses email and SMS.
Phishing targets lots of people at the same time, whilst blagging targets a specific individual.
Phishing emails often contain errors or vague information.
They may also use fake addresses.

## Malware and Malicious Code

Malware refers to many different forms of hostile, dangerous or intrusive software.
Anti malware software exists specifically to protect against malware.
Common sense and caution can reduce the risk of malware.

Avoiding download untrusted or suspicious programs can reduce the risk.

**Viruses**
A program designed to disrupt or damage a computer system.
May cause the system to stop functioning or loose data.


**Trojans**
Malicious software hidden in what seems to be a normal program.
Free games or music often contain trojans.
Once installed will damage the system or attempt to steal data.

**Spyware**
Records activity on a computer system.
Used to steal personal data.
A key logger is spyware which records every single key typed on the keyboard.

## Methods to Detect and Prevent Cyber Security Threats

The measures below may be used individually or together to protect systems.

**Biometrics**
Uses things such as fingerprints or facial recognition which are unique to a person's individual biology.
Much harder to fake or crack than traditional passwords.
Much more convenient than remembering a password.
Often used on mobile devices.

**Passwords**
A code or word known only by those who should have access to a system.
The most basic form of protection.

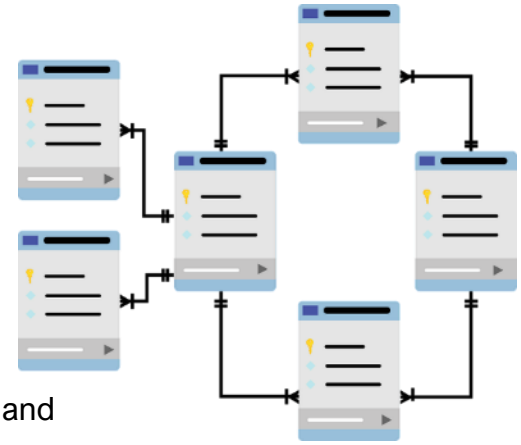
**CAPTCHA**
Aims to detect automated attempts to access a system.
Detects humans from computers
Asks the user to read a difficult word or identify an image or other task only a person could do easily.
Stops bots from being able to repeatedly try to access a system to crack it.


**Email Confirmation**
Requires a unique code or link sent in an email.
Means only those with access to a particular email account can access the system.
Often used to confirm password resets.

**Automatic Software Updates**
Installs patches automatically as soon as they are available.
Prevents patches from being forgotten about.
Allows patches to be installed more quickly than a human might be able to (e.g. for patches released at night)
Fixes bugs before they can be exploited.

# Topic 7: Relational Databases and Structured Query Language (SQL)

## Databases



A collection of organised and related data.
Data is organised into tables of data.
The columns are known as fields.
Rows are known as records.
Flat file databases store all data in a single table

## Relational Databases

Store data in multiple tables linked together.
Pieces of data are stored only once, preventing inconsistency and reducing storage space.
Links between tables join data together.
Stops the same data being stored multiple times (data redundancy)

## Database Key Concepts

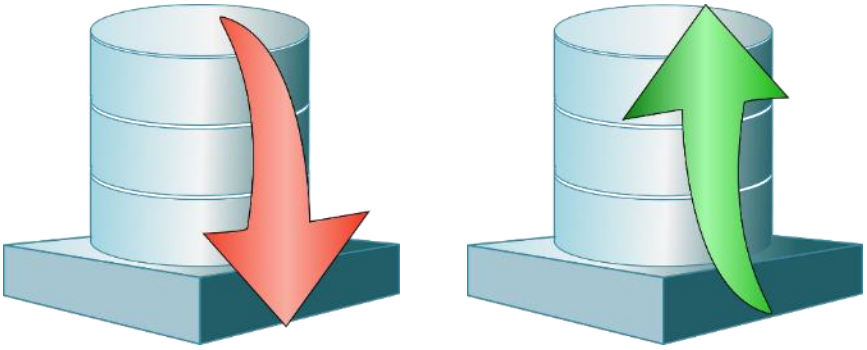
**Table**
Contains all the fields and the records.
A database may contain more than one table.

**Records**
A collection of data comprising a single row in a table.
The data about a single thing.

**Fields**
The column headings in a table.
Each field contains a different thing such as an address or name.
Each field might require different data types.

**Primary Key**
A unique identifier for each record.
Makes each record unique and allows it to be identified.
Each record has one primary key.
Primary keys are not used again when a record is deleted.
Can be automatically generated when data is added.

**Foreign Key**
The primary key from another table.
Used to link two tables together.



## SQL (Structured Query Language)

A programming language used to work with database.

**SELECT** – specify one or more fields to be included
**FROM** – specify one or more tables to be included
**WHERE** – specify one or more criteria to filter the results
**ORDER BY** – how the results should be sorted. **ASC** and **DESC** specify if the results should be sorted in ascending or descending order.

**INSERT INTO** – adds a record into a table
**VALUES** – the values to add

**UPDATE** – edits existing data
**SET** – the new values
**WHERE** – specify one or more criteria for the data to update

**DELETE FROM** – Deletes existing data
**WHERE** – specify one or more criteria for the data to delete

**Example**

```
SELECT customers.name, customers.address
FROM customers
WHERE customers.name = 'John'
ORDER BY address ASC
```

Returns the name and address of all records in the customers table where the name is John. Sorts the results by address in ascending order.

**Example**

```
INSERT INTO books (name, author)
VALUES ('Top Cheeses', 'John Smith')
```

Inserts a record in the books table with the name Top Cheeses and author John Smith

**Example**

```
UPDATE orders
SET status = 'paid'
WHERE customer = 'James'
```

Sets the status field to paid in the orders table for all records where the customer name is James .

**Example**

```
DELETE FROM students
WHERE name = 'Dave'
```

Deletes any records in the students table where the name is Dave.

# Topic 8: Ethical, Legal and Environmental Impacts of Digital Technology on Wider Society, Including Issues of Privacy

## General Principals

The areas below are rapidly evolving and changing.
In many cases there is no right or wrong answer and many people have differing viewpoints.
It is important to be aware of different views and concerns and to be able to discuss them.
In general people value privacy and do not like governments, security services or companies to have too much access to their data.
Governments often argue that they need this data to protect people, prevent terrorism and maintain security.
Companies often argue they need access to data to monitor their services and improve them.
**Digital Divide** is a term referring to a divide in society between those with access to technologies and those without it.

## Cyber Security, Hacking and Unauthorised Access

Discussed in more detail in Topic 6.
Most countries have laws against hacking and unauthorised access to systems.
The theft of computer equipment is covered by conventional legislation.
Data protection laws require companies to keep the data they hold secure.

**Issues**
How can we prove someone deliberately hacked a system?
Hackers may post people's information online, impacting their privacy.
How can laws keep up as technology evolves?

## Mobile Technologies

Mobile phones, tablets and other mobile technologies continue to rapidly evolve.
Devices are getting smaller and more powerful.

**Issues**
Is there a risk of a "digital divide" between those who can and cannot afford these devices?
Some services may need a phone number of smartphone app to access. What about those who do not have, want or know how to use these devices?
Devices change rapidly, causing excess waste which can be harmful to the environment.
Devices are often hard to repair, and so are replaced rather than repaired.
Should governments be able to intercept communications to maintain security and prevent terrorism?

## Wireless Networking

Wireless networks are more and more common.
Many businesses offer free wireless Internet access to people using their services.

**Issues**
How secure is the communication between devices?
Should governments be able to intercept communications to maintain security and prevent terrorism?
Some people have concerns about the health impacts of wireless networks.
Is there a risk of a "digital divide" forming between those with access to fast wireless networks and those without it.
Data sent on public wireless can often be easily intercepted.
Who's responsibility is it to keep data secure? The person using the network or the person who operates it?
How can we track who is using public wireless networks if they use them to commit a crime such as hacking or exchanging illegal material?
Governments or the providers of the networks may look at the data of those using the network. Is this legal and ethical?

## Cloud Storage

Discussed in more detail in topic 3.
The storage of files and data at a location which is accessed via The Internet.

**Issues**
How secure is the communication with cloud storage?
Data is subject to different laws depending on where it is stored.
Laws may require organisations to only store data in certain countries.
The data centres used for cloud storage require huge amounts of power to run and cool servers, where does this power come from? What are the environmental implications of generating it?
How private is data stored in the cloud? Can the provider's employees access it?
Should governments be allowed to access data stored in the cloud to prevent crime?

## Wearable Technologies


Technology such as smart watches, smart glasses and other fitness devices which are worn on the body.
The most known example is the fit bit.
May collect data about the wearer.
May allow the wearer to easily access data and control devices.

**Issues**
What happens to data collected by these devices?
How secure is the communication between these devices?
Is there a risk of a "digital divide" between those who can and cannot afford these devices?
These devices are inconspicuous , making them easier for criminals to use.

## Computer Based Implants

Technology or other electronics implanted into the human body.
Often used to assist those with a disability such as a bionic eye.
GPS tracking chips can be implanted into people.

**Issues**

Is it ethical to implant electronics into people?
How secure is the communication between these devices?
Is there a risk that those who cannot afford this may loose out?
How can the data collected be kept safe and secure?
The data from these devices is processed by different companies, apps and devices. How is it kept private?
Who owns the data collected?
Should governments have access to data collected to monitor their citizens health?
As these devices are relatively new, the long term health risks are often not known.

## Autonomous Vehicles

Self driving cars are developing at a rapid pace.
They use cameras, computers and sensors to know what is around them and drive accordingly.

**Issues**

Who is legally responsible in the case of a crash? The drive, the insurer, or the vehicle manufacturer?
Should a vehicle swerve onto the pavement to avoid a crash which would kill the driver, but in the process kill a pedestrian?
Should a vehicle drive to avoid hitting a dog if doing so would cause a crash which might injure the driver?
How can the batteries and other components be manufactured and disposed of without damaging the environment?
Are these vehicles legal to use?
Is it ethical for governments to access the cameras on vehicles to detect crime?
How can the data collected, such as car's location, be kept private?