A Level

# COMPUTER SCIENCE

## Paper-1

**Article # 251**
**A LEVEL NOTES SERIES**
CIE SYLLABUS CODE 9608
**2018 Edition**

**Fawad Khan**

Visiting teacher at:

Green Hall Academy
Alpha DHA
BSS
Roots International

Scan QR Code

READ & WRITE
PUBLICATIONS

☎ +92-42-35714038 +92-336-5314141 www.readnwrite.org readandwritepublications/Shop
**Head Office:** 3-C, Zahoor Elahi Road, Gulberg II, Lahore. readandwrite.publications@gmail.com
**Sale Point:** Shop No. 25-28 Lower Ground Floor, Haadia Haleema Centre, Ghazni Street, Urdu Bazar, Lahore

# Table of Contents

# 1.0 Hardware

## 1.1 Input Devices

### 1.1.1 Keyboard

It is an input device used to key in data into the computer. When a user press a key which contains a letter, numeral, symbol or punctuation mark, then the computer sends a signal to the monitor to display corresponding keys pressed.

There are several keyboard layouts such as QWERTY (ENGLISH), AZERTY (FRENCH) due to the language in use. Keyboards are simple, familiar and reliable technology to use. However, they are slow compared to other automatic data capture device such as OCR, OMR etc.

The QWERTY keyboard is the most common keyboard.

**Why do we have different layout?**

Each layout has been designed for fast data entry in their respective language by arranging keys mostly used at strategic positions on the keyboard.

Note:

- ✓ Keyboard is prone to damage from dirt or liquids because of the gaps between the keys.
- ✓ There is a special keyboard known as Braille keyboard which is meant for blind people

**Touch-sensitive / Numeric keypad:**

- Fewer keys simplifies input required and leads to fewer input mistakes.
- Keys are touch sensitive.
- Protected from elements / vandalism.
- Made of more resilient material because of volume of use.

*Musical keyboard*

It is arranged like a piano keyboard and needs a MIDI (musical instrument digital interface).

### 1.1.2 Mouse

It an input device used to point the graphical user interface by an associated cursor. Selection is made by pressing (clicking) switch buttons on the mouse body. It imitates the natural human action of pointing Main types of mouse include: mechanical (uses ball underneath) and optical (senses reflected light).

| Advantage | Disadvantage |
|---|---|
| ▪ Fast method for input commands<br>▪ Intuitive to use to point at things | ▪ Can be easily damaged<br>▪ Disabled person have difficulty to use a mouse |

| | |
|---|---|
| | - Usually needs a flat surface to operate |

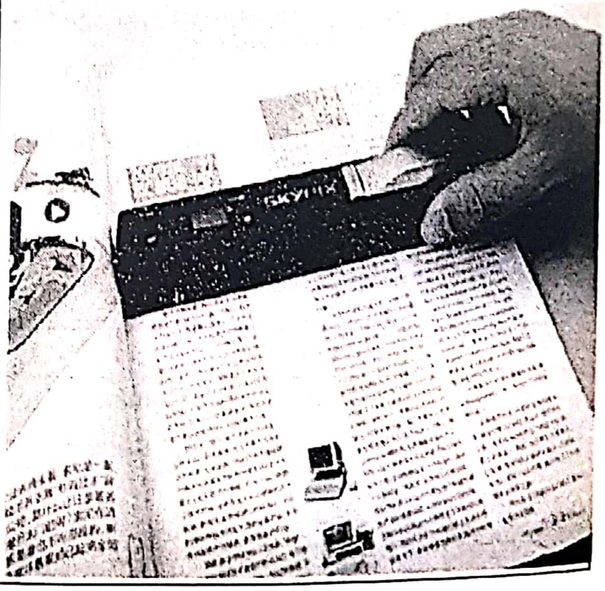
### 1.1.3 Tracker mouse

A tracker ball is rotated by hand and the sensors detect the movement of the ball and relay the data to the CPU. The advantage over a mouse is that it remains stationary and as such does not require as much space as a mouse.

| Advantage | Disadvantage |
|---|---|
| - Disabled person with limited movement can use it.<br>- Usually takes less space since it does not need to be moved around.<br>- It is more accurate than mouse. | - It is more expensive than conventional mouse.<br>- It can be tricky to get used to. |
| - Use : It is used mainly for people who have limited movement | |

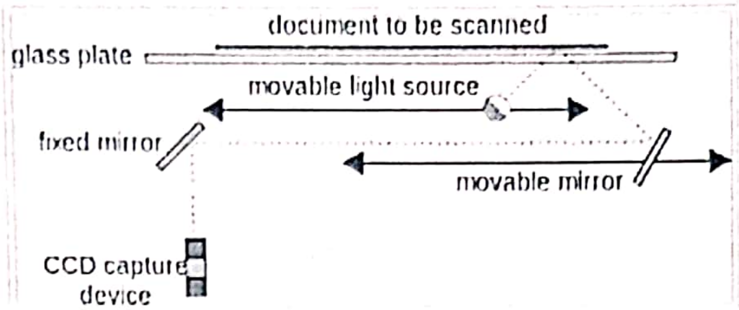
### 1.1.4 Scanner

Scanners usually convert hardcopy format to an image format. There are several types of scanners 2D and 3D scanner. Scanners can be categorised as Flat bed and Hand held scanners.

| Flat Bed Scanner | Hand Held Scanner |
|---|---|
| |  |

### 1.1.4.1 2D Flatbed Scanner

These are usually used to convert images and text into a digital format. The diagram shows how it works.



*How 2D Flatbed Scanner works?*

The object is placed on the glass pane face down. The light moves to the start of the document and illuminates a slices. The latter is then reflected into the CCD where its image is stored. The light and mirror then move down to the next slide. Once all the slices are completed, the final image is constructed.

### 1.1.4.2 3D Scanners

A **3D scanner** is a device that analyses a real-world object or environment to collect data on its shape and possibly its appearance (i.e. colour).The collected data can then be used to construct digital, three dimensional models in the x, y and z axis.

*Limitations*

Optical technologies encounter many difficulties with shiny, mirroring or transparent objects.

*Use*

- Entertainment industry in the production of movies and video games.
- Industrial design
- Orthotics and prosthetics
- Reverse engineering and prototyping
- Quality control/inspection and documentation of cultural artefacts.

### 1.1.5 Sensors

It is an input device that produces an electrical signal in response to a stimulus. There are several stimuli which can be chemical or physical in nature. These devices are highly accurate and are usually used in monitoring and control environments without human intervention. Frequently, an analogue to digital converter (ADC) is needed since sensor data is often in analogue form and computers can only process digital data.

| Advantage | Disadvantage |
|---|---|
| ▪ Take accurate measurement<br>▪ Can give readings continuously and frequency can be set as required<br>▪ Can be used in hazardous environment | ▪ Need an ADC (Analogue to Digital Convertor) to work in microprocessor environment |
| **Use :** Usually used in monitoring and control systems | |

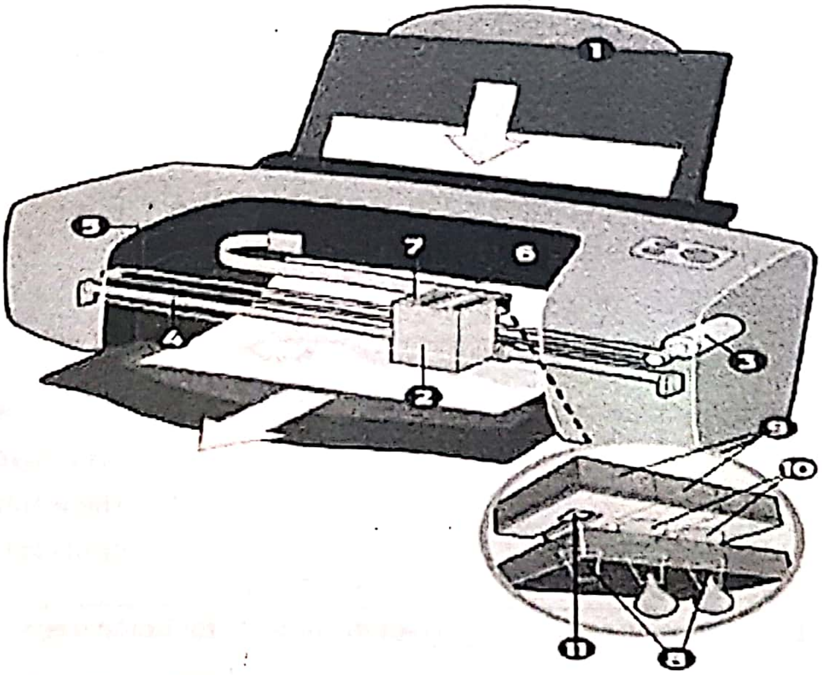
### 1.1.5.1 Types of Sensors

| SENSOR TYPE | APPLICATIONS |
|---|---|
| **Temperature** | ✓ It is used in controlling and monitoring of temperatures in chemical processes , green house , modern vehicle engines temperature.<br>✓ It is used in controlling central heating systems or washing machines |
| **Pressure** | ✓ It is used in measuring blood pressure monitor<br>✓ It is used in chemical plants to control pressure of gases or liquids in vessels<br>✓ It is used in burglar alarm systems |
| **Light** | ✓ It is used for automatic doors to detect presence of a person<br>✓ It is used in car for automated system that control headlights<br>✓ It is used in robotic factory to follow a route<br>✓ It is used in traffic lights to detect the presence of a vehicle |
| **Moisture / Humidity** | ✓ It is used in monitoring of greenhouse environment<br>✓ It is used in controlling heating, ventilation and airconditioning (HVAC) system in musuems, library , research labs and others<br>✓ It is used in meteorological stations to measure humidity for weather forecasting. |
| **Oxygen** | ✓ It is used for environment (e.g. measuring oxygen content in a river to check for pollution) |
| **Infrared** | ✓ It is used to detect movement by interrupting an infra red beam in security systems |

| | |
|---|---|
| | ✓ It is also used in counting materials as each one travels through the beam<br>✓ It is used in remote control in TV appliances |
| **Motion** | ✓ It is used in detecting speed of vehicles in radar system |
| **Accoustic** | ✓ It is used to pick up sound<br>✓ It is used to detect movement of liquids in pipes in chemical plants |
| **PH** | ✓ It is used to monitor soil acidity in greenhouses<br>✓ It is used to measure acidity levels in rivers for pollution monitoring applications<br>✓ It is used to monitor/ control processses where acidity levels are crucial |
| **Gas** | ✓ It can be used to monitor the atmosphere in case of leakage in plants or other applications<br>✓ It can be used in environment monitoring of air pollution<br>✓ It can be used in a breathanalyser to measure/estimate the concentration of alcohol vapour in blood |
| **Magnetic Field** | ✓ It can be used to study the field around permanent magnets, coils, and electrical devices. |

## 1.2 Output Devices

### 1.2.1 Inkjet printer



| | | | | | |
|---|---|---|---|---|---|
| 1.Paper tray | 2. cartridge assembly | 3. printhead motor | 4.sliding rod | 5. belt | 6. printhead |
| 7. ink cartridge | 8. nozzles | 9. ink chambers | 10. droplets | 11. resistor | |

Inkjet printing is a non-impact process. Ink is emitted from nozzles while they pass over media

Operation of an Inkjet Printer

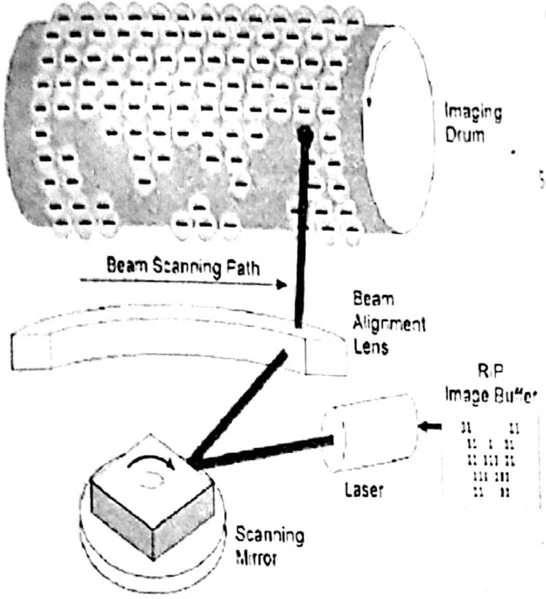
1. Liquid ink in various colours being squirted onto paper and other media, like plastic film and canvas, to build an image.
2. A print head scans the page in horizontal strips, using the printer's motor assembly to move it from left to right and back again, while the paper is rolled up in vertical steps, again by the printer.
3. A strip or row of the image is printed, then the paper moves on, ready for the next strip.
4. To speed things up, the print head doesn't print just a single row of pixels in each pass, but a vertical row of pixels at a time.

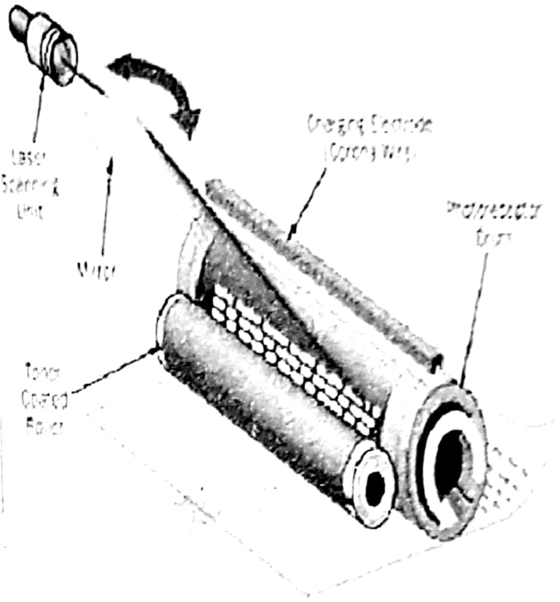
For most inkjet printers, the print head takes about half a second to print the strip across a page. On a typical 8 1/2"-wide page, the print head operating at 300 dpi deposits at least 2,475 dots across the page. This translates into an average response time of about 1/5000th of a second. There are several types of inkjet printing. The most common is "drop on demand" (DOD), which means squirting small droplets of ink onto paper through tiny nozzles; like turning a water hose on and off 5,000 times a second. The amount of ink propelled onto the page is determined by the print driver software that dictates which nozzles shoot droplets, and when.

The nozzles used in inkjet printers are hairbreadth fine and on early models they became easily clogged. On modern inkjet printers this is rarely a problem, but changing cartridges can still be messy on some machines. Another problem with inkjet technology is a tendency for the ink to smudge immediately after printing, but this, too, has improved drastically during the past few years with the development of new ink compositions.

| **Advantage** | **Disadvantage** |
|---|---|
| ▪ It print in relatively high resolution<br>▪ It has a low purchase cost<br>▪ It is physically small and light | ▪ It has a low rate of printing thought suitable for home and office<br>▪ Ink cost is relatively high<br>▪ The water-soluble ink, which runs if prints get damp |
| **Use** : for printing coloured pages documents for home users | |

### 1.2.2 Laser Printer.





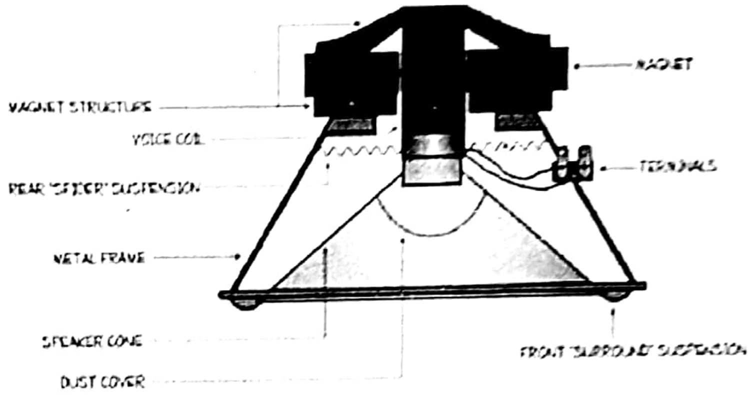
Operation of a laser printer

A laser scans the drum to print an image which is charged negatively and the toner (powdered ink) also is charged negatively. The area where the laser beam has burn the images are neutral and as such toner particles are attracted and the rest rejected. The roller press the sheet of paper against the drum to transfer the toner to paper and then it is fused to melt it definitely on paper.

| Advantage | Disadvantage |
|---|---|
| ▪ Prints high resolution details<br>▪ Print more copies a minute than a ink jet printer.(Rate of Output)<br>▪ Ink lasts for longer so they don't need to change the cartridges so often. | ▪ Expensive to purchase.<br>▪ Bulky and heavy.<br>▪ Toner content are toxic and must be disposed carefully after use |
| ▪ **Use:** Applications which require rapid, high quality and high volumes of output, e.g. most offices and schools. | |

### 1.2.3 Speakers

A speaker converts electrical signals into acoustical energy that is sound. The speaker moves back and forth which increases and decreases the air pressure in from of it and creates sound waves. The diagram below show some essential parts of the speaker.



The larger the cone, the more mass and surface area a speaker will have. This more air will move which results in a louder speaker. Essentially the voice and standing magnetic field trigger the cone movement. When an electrical signal from an audio amplifier is put into a speaker's voice coil, the voice coil generates an alternating magnetic field. Hence, this causes attraction and repulsion of the cone attached to the voice coil to move. When the cone moves between 20Hz to 20 KHz, the speaker generates the sound wave.

## 1.3 Storage Devices

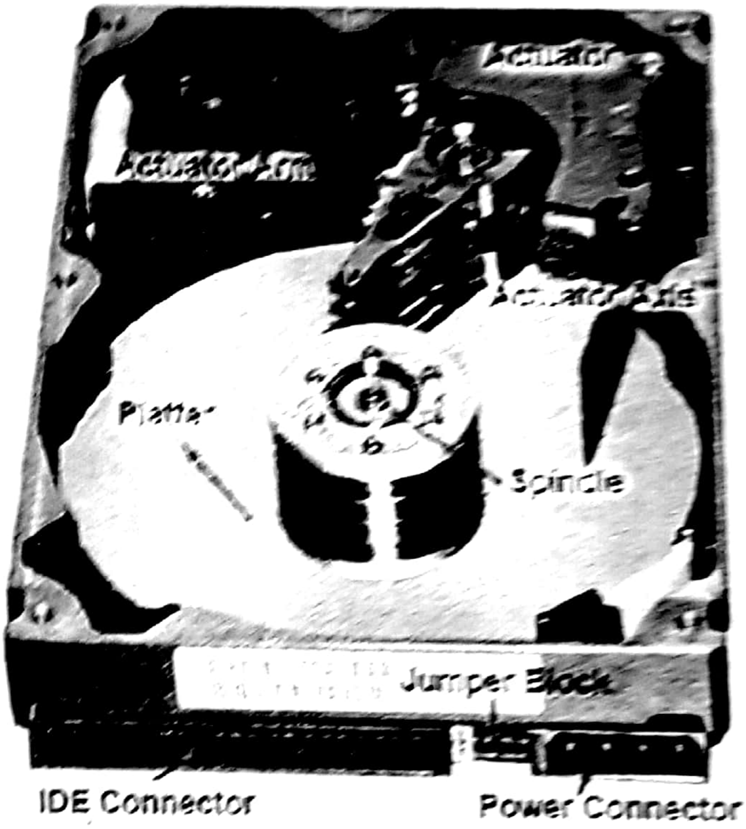
### 1.3.1 Solid State flash memory

These are non-volatile re-writable memories evolved from the EEPROM; they are often connected to the USB port on the computer enabling a user to transfer files between computers.

- SD (secure digital) cards are a type of flash memory used in digital cameras and mobile phones.
- XD (extreme digital) cards are also a type of flash memory used in digital cameras and are similar to SD cards.
- Digital storage cards (DSCs) are used as removable storage for phones, camera, PDA and other devices.

### 1.3.2 Hard disk

The hard disk consists of rotating disks called platters in the form of a cylinder that spins usually between 5400 to 10000 revolutions per minute(RPM). A thin magnetic coating is layered onto the surface of the platter that is polished to mirror type smoothness.



There is a Read Write head per platter each one attached to the actuator shaft which moves altogether during reading and writing of data. The data are stored in microscopic areas called magnetic domains. The HDD records data in concentric circles known as tracks which are numbered from the outermost to the innermost edge of the platter. These tracks are subdivided into smaller units called sectors.

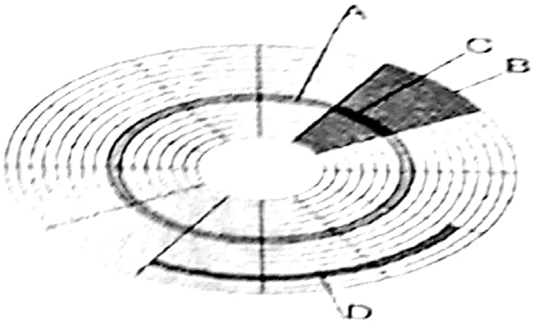
#### Disk Structure

A – Track

B – Geometrical Sector

C – Track Sector

D - Cluster



### 1.3.3 Optical discs

Optical disks as the name suggest uses reflection of a laser from the pitted surface of the disk to read information. Large amount of data can be stored on the surface and offers portability from one machine to another.

#### 1.3.3.1 DVD – RAM

It is similar to DVD-RW but differs in the way data is stored on the surface of the disc. It can be rewritten more times than conventional DVD-RW

### 1.3.3.2 Blu-Ray

Blu-ray are similar in appearance to a normal DVD but has larger storage capacities (usually from 25 GB (Single Layer) and 50 GB (Dual Layer)).The format was developed to enable recording, rewriting and playback of high definition video.

| | |
|---|---|
| Use | : HD Video industry |
| Drawback | : The disc is very expensive compared to corresponding DVD media type. |
| BD-ROM | - read only format for distribution of HD content, games, software etc. |
| BD-R | - recordable format for HD video recording and PC data storage. |
| BD-RE | - rewritable format for HD video recording and PC data storage. |

### 1.3.3.3 Digital Versatile Disks (DVDs)

A single DVD can hold as much information as several CDROMs. A typical DVD has a storage capacity ranging between 4.7 GB -8.4 GB. (Single or Dual Layer). This means that a single DVD can be used to hold an entire movie. A film stored on a DVD can even give the viewer a choice of several languages in which it can be played back. The device used to read DVD is called DVD-ROM and the media on which files are recorded is called DVD-R.

### 1.3.3.4 Compact Disks (CDs)

It is secondary storage medium and is portable. It can be read only memory and as such we cannot change data once written. E.g. used to store programs/data/pictures/films, etc. Its storage capacity varies between 650 -700 MB. The device use to read CD is called CD-ROM and the media on which files are recorded is called CD-R.

| Advantage | Disadvantage |
|---|---|
| ▪ Portable, many computers have a CD ROM drives.<br>▪ Large storage capacity<br>▪ Fast access time | ▪ Slower access times and smaller storage capacity than a hard disc.<br>▪ Alterable. |
| **Use** : Applications which require a single 'burning' of data: | |
| **E.g.**<br>▪ Recording of data for archiving or a one off backup.<br>▪ Recording of music downloads from the Internet.<br>▪ Recording of music from MP3 format. | |

### 1.3 3.5 CD RW

| **Advantage** | **Disadvantage** |
|---|---|
| - Portable, many computers have a CD ROM drives.<br>- Large storage capacity<br>- Fast access time<br>- Alterable (as many times as you want). | - Slower access times and smaller storage capacity than a hard disc.<br>- Alterable. |
| **Use:** Applications which require the updating of information and ability to record over old data. | |

### 1.3.4 Actuators

An actuator is defined as a type of motor that is responsible for moving or controlling a mechanism or system. The actuator is the mechanism by which a control system acts upon an environment.

## 1.4 Hardware – Main Memory

### 1.4.1 RAM V/S ROM

| **RAM (Random Access Memory)** | **ROM (Read Only Memory)** |
|---|---|
| **Temporary Storage** -Data is temporary - it is only there as long as the computer is on and it can be changed<br>**Volatile** -Data is lost if the computer crashes or is turned off.<br>**Read /write** -Data can be written to or read from.<br>**Use :** To store OS, Programs | **Permanent Storage** -Data is permanent - it can never be changed<br>**Non Volatile** -Data is never lost<br>**Read Only** -Data can only be read<br>**Use:** To keep the bootstrap routine. |

### 1.4.2 DRAM (Dynamic RAM)

This is the RAM people are talking when they are adding memory in their computer. It is used in computers since it is cheaper than any other type of memory. Dynamic refers to the memory chips' need for a constant update signal (also called a refresh signal) in order to keep information that is written there. In case this signal is not received every so often, the information will cease to exit. Popular implementation of DRAM are: SDRAM, DDR, DDR2, and DDR3

### 1.4.3 SRAM (Static RAM)

SRAM does not require a refresh signal like DRAM. The memory chips are more complex and as such more expensive. However, they are faster. Static RAM uses switching circuitry instead of capacitors and can hold a charge without a constant refresh. Due to the speed, SRAM is commonly used for CPU cache but is rarely used as the primary RAM because of its cost. E.g. of SRAM include: CPU L1 Cache, CPU L2 Cache

# 2.0 Information Representation – Number Representation

**Introduction to Binary Numbers**

## 2.1 Number Systems

Number systems are simply ways to count things. Ours is the base-10 or radix-10 system.

| $10^9$ | $10^8$ | $10^7$ | $10^6$ | $10^5$ | $10^4$ | $10^3$ | $10^2$ | $10^1$ | $10^0$ |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

1. **BASE 2** - BINARY - 0, 1
2. **BASE 8** - OCTAL - 0,1,2,3,4,5,6,7
3. **BASE 16** - HEXADECIMAL - 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
4. **BASE 10** - DENARY - 0,1,2,3,4,5,6,7,8,9

### 2.1.1 Denary (Base 10)

Base 10: 0,1,2,3,4,5,6,7,8,9 - It uses those 10 digits only

- ✓ From an Abacus point of View

| $10^3$<br>TH | $10^2$<br>H | $10^1$<br>T | $10^0$<br>U |
|---|---|---|---|
| 4 | 7 | 2 | 1 |

$$\mathbf{4721} = 4 * 10^3 + 7 * 10^2 + 2 * 10^1 + 1 * 10^0$$

### 2.1.2 BINARY (Base 2)

- ✓ It is also known as Binary Digits
- ✓ It uses either 0 or 1 to represent the numbers in the required number of bits

#### Nibble

- Half a byte is known as a nibble, i.e. 4 bits

#### Byte

- A byte consists of 8 bits and is usually used to represent a character.

#### Converting a Binary Number to Denary

To convert a binary number to denary, simply follow the following steps:

1. Starting from the RHS (Right Hand Side), Label the binary digits starting from 2 power 0 to the number of digits in ascending order.
2. Multiply each binary digit with the corresponding power of 2 above it.

3. Simply add the sum of each multiplication and you get the final result

e.g. $(1110\ 1111)_2 = (239)_{10}$

| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

$= 1 * 2^7 + 1 * 2^6 + 1*2^5 + 1*2^3 + 1* 2^2 +1 * 2^1 + 1 * 2^0$

$= 239$

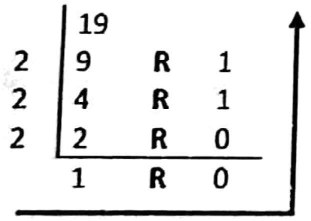
Convert Denary to Binary

There are 2 methods that may be used to convert denary to Binary:

- ✓ **Method 1:** Long Division
- ✓ **Method 2:** Tabular method or Weight System

### Method 1:

✓ Repeatedly divide the number by 2 till number is 1 and note the remainder in reverse order as shown in the diagram below



✓ Hence, $(19)_{10}$ in 8 bits = $(000\ 1\ 0011)_2$

**Method 2:**

✓ Imagine in a country the following notes are available and you have to pay exactly using those notes. Each time you pay the largest note and try to settle the rest by highest note available.

✓ In fact it is a table of base 2 in 8 bits or more if required

2.1.3 Octal Numb

| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | Value |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 15 |

The Octal Number system uses digits from **0 – 7** only. Hence, all the numbers are represented using **3 bits** binary digits. The table below shows the denary number with its octal and binary equivalent

| DENARY | BINARY | OCTAL |
|---|---|---|
| 0 | 000 | 0 |
| 1 | 001 | 1 |
| 2 | 010 | 2 |
| 3 | 011 | 3 |
| 4 | 100 | 4 |
| 5 | 101 | 5 |
| 6 | 110 | 6 |
| 7 | 111 | 7 |

### Converting From Octal to Denary

To convert from octal to denary, simply multiply each digit by the weight system of base 8. Hence:

e.g. $(723)_8 = (467)_{10}$

| $8^2$ | $8^1$ | $8^0$ |
|---|---|---|
| 7 | 2 | 3 |

$= 7 * 8^2 + 2 * 8^1 + 3 * 8^0$

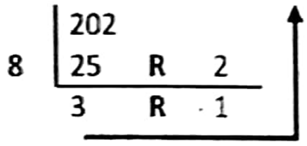
$= 467$

**Tip for faster calculation:**

- $8^0$ = 1   $8^1$ = 8
- $8^2$ = 64   $8^3$ = 512
- $8^4$ = 4096

### Convert from denary to octal

Simply divide the number repeatedly by 8 till value is less that 8 to express to octal form.



### Convert Binary to Octal

In the octal (base 8, number system, only the 8 digits from 0 to 7 are used). To convert binary to octal, simply dive the number into groups of 3 bits staring from the right hand side first.

Each set of 3 bits can then be represented by a number between 0 and 7. E.g. $(11\ 010\ 110)_2 = (326)_8$

$$\underbrace{11}_{3}\quad\underbrace{010}_{2}\quad\underbrace{110}_{6}\ .$$

### 2.1.4 Hexadecimal Number System

The Octal number system uses digits from 0 -9 , A(10) ,B(11),C(12) ,D(13) ,E(14) ,F(15) only . All the numbers are represented in 4 bits binary digits. Hexadecimal is also abbreviated to HEX, sometimes X, or base 16

| DENARY | BINARY | HEXADECIMAL (HEX) |
|---|---|---|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |

### Convert From Hexadecimal to Denary

To convert from hexadecimal to denary, simply use the weight system and as base use 16. If it is a letter, A-F, replace it with the corresponding denary value

E.g. 1

$$(769)_{16} = (1897)_{10}$$

| $16^3$ | $16^2$ | $16^1$ | $16^0$ |
|---|---|---|---|
|  | 7 | 6 | 9 |

$= 7 * 16^2 + 6 * 16^1 + 9 * 16^0$

$= 1897$

E.g. 2

$(1AB)_{16} = (427)_{10}$

$16^3$ $16^2$ $16^1$ $16^0$

1 A B

$= 1 * 16^2 + A * 16^1 + B * 16^0$

$= 1 * 16^2 + 10 * 16^1 + 11 * 16^0$

$= 427$

**Tip for faster calculation:**

- $16^0 = 1$
- $16^2 = 256$
- $16^4 = 65536$

$16^1 = 16$

$16^3 = 4096$

## Convert From Denary to Hexadecimal

To convert from denary to hexadecimal, simply do long division with base 16 till the value is less than 16 and look up for corresponding values in hexadecimal table

e.g. 1. $(69)_{10} = (45)_{16}$

16 | 69

4 **R** 5

e.g.2. $(58)_{10} = (3A)_{16}$

16 | 58

3 **R** 10 (A)

**Note:** in this case, the remainder is 10 which is equivalent to A in Hexadecimal

## Convert Binary to Hexadecimal

To convert binary to hexadecimal, simply group the binary numbers in 4 bits and convert it to the denary equivalent or look up in the hexadecimal table values. There are 2 methods to convert binary to hexadecimal

**Method 1:**

i. Convert Binary -> Decimal
ii. Convert Decimal -> Hex

- E.g. 0110 1110 = $2^6+2^5+2^3+2^2+2^1$
- $(0110\ 1110)_2$ = $(110)_{10}$
- $(110)_{10}$ = $(6E)_{16}$

**Method 2:**

- To convert a binary number to hexadecimal, divide to groups of 4 bits starting from the right hand side (RHS). Then write the hexadecimal equivalent of each group of 4 bits by looking at the hexadecimal table.

$$\underbrace{1101}_{D}\quad\underbrace{0111}_{7}$$

$(1101\ 0111)_2 = (D7)_H$

**Note:**

Hex is used to write assembly code instructions because:

- ✓ Hex is short form of binary
- ✓ It is easy to convert hex to binary and vice versa

### 2.1.5 BCD (Binary Coded Decimal)

Some numbers are not proper numbers because they don't behave like numbers. A barcode for chocolate looks like a number, and a barcode for sponge cake looks like a number, but if the barcodes are added together the result is not the barcode for chocolate cake. The arithmetic does not give a sensible answer. Values like this that look like numbers but do not behave like them are often stored in binary coded decimal (BCD). Each digit is simply changed into a four bit binary number which are then placed after one another in order.

In binary coded decimal, each decimal digit is represented in 4 bits binary

| DECIMAL | BCD |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |

| | |
|---|---|
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

e.g. 902 in BCD is $(1001\ 0000\ 0010)_{BCD}$

Simply replace all decimal digits by its corresponding BCD value

The following are ways data may be represented.

1. Representing unsigned integers (Values **without Sign Consideration**)
2. Representing signed integers
    - ✓ Signed magnitude
    - ✓ One's complement
    - ✓ Two's complement
3. Hexadecimal
4. Octal
5. Binary Coded decimal
6. Representing real numbers
    - ✓ Fixed point
    - ✓ Floating point
7. Representing characters
    - ✓ ASCII
    - ✓ EBCDIC
    - ✓ UNICODE

**Integer** : are whole number both positive and negative, e.g. +17, -18, 45. It is used to represent number of pupils in a class, counting variables, etc.

**Real numbers** : numbers that contain a decimal or an exponent or both. For example 1.75, $18 * 10^{-3}$, $1.58 * 10^{2}$.It is used to represent the result of a division or average.

## 2.2 Representing Unsigned Integers **(means no sign such as positive or negative)**

✓ Unsigned integer: 0, 1, 2, 3 .....

5 = 0000 ← 0101 → LSB ( Least Significant Bit)
(MSB) Most Significant Bit

Consider a Byte,

- ✓ Maximum unsigned integer that can be represented in 8 bits = 1111 1111 = 255
- ✓ Minimum unsigned integer that can be represented in 8 bits = 0000 0000 = 0
- ✓ Maximum unsigned integer in 8 bits = 255 = $2^8 - 1$
- ✓ Hence, in n bits, unsigned integer maximum = $2^n - 1$

### 2.2.1 Representing Signed Integers

If a computer system uses a byte to store a number in the way that was suggested above, there are three problems that arise.

1. The first is that the biggest number that can be represented is 255 because there aren't enough bits to store bigger numbers. This is easily solved by using more than one byte to represent a number. Most computer systems use either two or four bytes to store a number. There is still a limit on the size that can be represented, but it is now much larger.
2. The second problem is not so easy to solve, how to represent fractions. This will be looked at later in this chapter.
3. The third problem is how to store negative numbers.

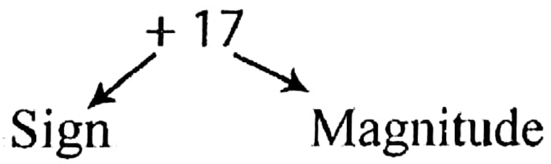
The example we used for binary storage was 117 which becomes 0111 0101 in binary. If we want to store +117 or –117, these numbers need a second piece of data to be stored, namely the sign.

Hence, the following representation were introduced:

1. Sign Magnitude (SM notation) = wrong method
2. One's complement
3. Two's complement

**Note:** using the above 3 notations, their positive representation are the same but they differ in the way they represent negative values.

#### 2.2.1.1 Sign Magnitude



+ 17 :

- Sign = +
- Magnitude = 17

-17:

- Sign = -
- Magnitude = 17

There are only 2 signs (+ or -). One bit is sufficient to represent sign.

Usually 0 represent positive and 1 negative. Hence, using n bit to represent a signed magnitude number, it is to be noted that the most significant bit is reserved for sign and the rest for magnitude

e.g. if we use 8 bits , 1 bit for sign (left most ) and 7 bit magnitude( the rest)

| | | |
|---|---|---|
| **SM NOTATION** | **SIGN (1 BIT)** | **MAGNITUDE (REST)** |

**Procedure (for negative values):**

1. Convert the denary number to binary first
2. Then put the positive denary number in the number of bits required
3. For Negative , simply change the MSB to 1

For example, using 8 bits to store a signed integer, using SM notation

| | | |
|---|---|---|
| + 17 = | 0 | 001 0001 |
| -17 = | 1 | 001 0001 |

**Disadvantage of SM Notation**

SM is not a good method because there is need to compare sign and magnitude before adding or subtracting which makes these operations slow.

### 2.2.1.2 One's Complement

If the number is positive, it is represented as an ordinary binary number (sign bit = 0 (MSB))

If the number is negative, it is represented as the one's complement of the positive number.

$$0 \longrightarrow 1$$
$$1 \longrightarrow 0$$

**Procedure (for negative values):**

1. Convert the denary number to binary first
2. Then put the positive denary number in the number of bits required
3. Inverse all the bits , i.e. all 0 to 1 and all 1 to 0

For example, using 8 bits to store a signed integer, using One's Complement

| | |
|---|---|
| + 17 = | 0001 0001 |
| -17 = | 1110 1110 |

### 2.2.1.3 Two's Complement

If the number is positive, it is represented as an ordinary binary number (sign bit = 0).

If the number is negative, then sign bit = 1 .

✓ There are 2 methods that may be used to convert a number to 2's Complement

Method 1:

1. Invert the bits as in one's complement
2. Add 1 to the answer

| | | |
|---|---|---|
| +17 = | 0001 0001 | |
| - 17 = | 1110 1110 | One 's complement representation of -17 |
| | 1 | + |
| 2's complement | 1110 1111 | |

**Note:** In 2's complement, the Most Significant bit value is negative. E.g. using 8 bits

| $-2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

Hence,

| | $-2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|---|---|
| -17 = | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| -17 = | $-2^7 + 2^6 + 2^5 + 2^3 + 2^2 + 2^1 + 2^0$ | | | | | | | |

Method 2

Starting from the RHS (Right Hand Side) i.e. Least Significant Bit, the least significant zeroes and the first 1 remain unchanged and the rest of the digits are inverted

e.g.

| | $-2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|---|---|
| + 56 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| - 56 = | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| -56 = | $-2^7 + 2^6 + 2^3$ | | | | | | | |

**Maximum and Minimum Values**

| | $-2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|---|---|
| Maximum in 8 bits | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| = | $2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ | | | | | | | |

= $127 = 2^7 - 1$

Maximum in n bits $2^n - 1$

Minimum in 8 bits

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

= $-2^7 = -128$

Minimum in n bits $-2^{n-1} - 1$

## 2.3 Binary Addition Laws

| | **SUM** | **CARRY** |
|---|---|---|
| 0 + 0 | **0** | 0 |
| 0 + 1 | **1** | 0 |
| 1 + 0 | **1** | 0 |
| 1 + 1 | **0** | 1 |
| 1 + 1 + 1 | **1** | 1 |

## 2.4 Two's Complement Addition

Add two numbers, including sign bits and start any carry out. Negative numbers must be in two's complement representation. The sum is obtained after addition if negative is also in the two's complement representation.

Consider the following addition in 8 bits which shall apply to any number of bits by principle

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **0** | | | | | | **1** | 0 | **Carry In = 0** |
| +9 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | |
| +5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | |
| +14 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | **Carry Out** = 0 |

Note : = $2^3 + 2^2 + 2^1$

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | | 1 | 1 | 0 | **Carry in = 0** |
| + 9 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | |
| -5 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | **Two's Complement of +5** |
| +4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | Carry Out = 1 ,Discarded to obtain correct value |

Note : = $2^2$

**Advantage of Two's Complement**

- ✓ There is no need to compare sign and magnitude before adding or subtracting

### 2.4.1 Overflow

Overflow occurs when the result an arithmetic operation is too large to fit in the number bits allocated to it. Consider, the following example in 8 bits

- ✓ Largest 2's complement in 8 bits value = +127

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | | | | | 0 | |
| + 80 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | **Carry in = 0** |
| +60 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | **Two's Complement of** |
| +140 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | |

Note : = $-2^7 + 2^3 + 2^2 = -116$ **Carry Out = 1 , MSB = 1**

**Answer should have been +140 but we obtained -116**

The digit 140 cannot be stored in 8 bits since largest value that can be represented in those number of bits is +127.

If two positive numbers are added (and value cannot be represented in number of bits allocated) and an overflow occurs, the overflow will be negative
Smallest two's complement in 8 bits = -128

If two negative numbers are added and an overflow occurs, the overflow will be positive.

Consider the following example in 8 bits

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **0** | | | | | | | **0** | |
| **- 80** | **1** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **Carry in = 0** |
| **-60** | **1** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **Two's Complement of** |
| **-140** | **0** | **1** | **1** | **1** | **0** | **1** | **0** | **0** | |

**Note : =** $2^6 + 2^5 + 2^4 + 2^2 = +116$ **Carry Out = 1 , MSB = 0**

**Answer should have been -140 but we obtained +116**

**Algorithm to detect overflow**

```
If Carry In <> Carry Out then
        Overflow has occurred
End if
```

## 2.5 Character Set

The computer system usually uses several types of encoding system to exchange or store data. Usually ASCII and Unicode are the most common one.

### 2.5.1 ASCII

ASCII (American Standard Code for Information Interchange) is most popular and widely used. The ASCII coding system uses seven bits to represent each character and the eighth bit as a means of checking the rest.

This means that 128 different characters can be represented in the standard ASCII character set. Or 256 for Extended ASCII, (As this is the most common character set, people generally consider a byte to be eight bits). e.g. Character 'A' = 65 in ASCII and 'Z' = 90

**Advantage of Using ASCII**

Since each character uses only 8 bits, the data stored takes less space in memory compared to Unicode

### 2.5.2 Unicode

Unicode is a more recent, 16-bit code that uses two bytes or 32 –bit (latest) that uses 4 bytes to encode characters. Using 16 bits makes it possible to represent over 65,000 characters. This means that all the characters used by the world's languages can be represented in Unicode.

**Use**

- It is widely used to handle documents, particularly if a single document needs to be written in, for example, English, Arabic and Chinese.

A byte consists of 8 bits. E.g. 0101 1010. It is used to represent a character. There are different version commonly known as UTF- 8, UTF -16, and UTF-32 where the digits accompanying the UTF- is the number of bits used.

**Advantage of Using Unicode over ASCII**

- ✓ ASCII stores a much smaller set compared to Unicode and hence, ASCII is limited in the representation of Latin character set and characters in other languages.

## 3.0 Information Representation - Images

**Images**

Digital images requires lots processing due to the amount of information they need to store. They can be grouped in two main categories which are namely:

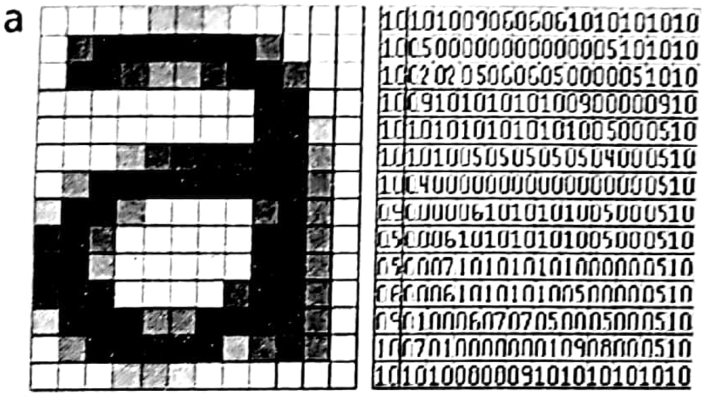
- Vector
- Bitmap.

## 3.1 Bitmap (also known as device independent bitmap (DIB) / raster image)

**Bitmap graphics** is defined as a collection of pixels from an image mapped into specific memory locations whereby their binary colour value is stored.

Sometimes it is also called raster images. The image takes the form of an array where the value of each element represented by a pixel represents the colour of that part of the image. The diagram below shows how a letter 'a', might be represented in a matrix whereby the values indicated on right hand side depicts the image brightness.

**Note:** Each horizontal line in the image is known as a scan line.



101010090606061010101010
10C500000000000005101010
10C2020506060500000051010
10C9101010101010090000000910
10101010101010101005000510
10101005050505050504000510
10C4000000000000000000000510
09C000061010101010050000510
05C00061010101010100500510
09C007101010101010000000510
06C0061010101010050000000510
09C1000607070500050000510
10C70100000000010908000510
10101000000009101010101010

### Use of Bitmap

- It is usually used to store natural images, graphic images with lots of details and photographs.

### BITMAP file Header

It is defined as the block of bytes at the start of the file which is normally used to identify the file.

**Use :** Application usually read this block initially to ensure that the file is a typical BMP file and is not damaged.

## 3.2 Image Terminologies

### Pixel

It is defined as the smallest possible addressable area defined by a solid colour, represented as binary, in an image. In the above diagram, each box represents a pixel

### Resolution

It is a measure of the sampling density and is often measured in **ppi** (pixel per inch). The higher the resolution, the crisper is the image since there is more pixel available to represent the image.

## Image Resolution

It is defined as how many pixels an image contains per inch/cm.

## Screen Resolution

It is defined as the number of pixels per row by the number of pixels per column.

## Colour Depth

The number of bits used to represent the colour of a single pixel.

## 3.3 File Estimation of Bitmap images

We are going to work out how many pixels are required to display a single frame on a VGA screen.

Checking the resolution:

- Height = 480
- Width = 640
- Area = Width * Height = Total Pixels
- Area = 640 * 480 = 307200

**Area = Width * Height = Total Pixels**

## Questions on Images Bitmaps

1 **What is the resolution of a 100 pixel by 70 pixel image? :** 100 * 70 = 7000 pixels

2 **What is a benefit of having a higher resolution image?**

- Higher resolution images are able to display more detail, providing crisper images

3 **What might be a drawback of having a very high resolution image?**

- It will require a lot of space to store it. Meaning you'll quickly run out of memory, or it'll take a long time to transmit images across the internet or other data route.

## 3.4 Colour Depth

Consider the following set of images to understand colour depth implication

| **Colour Depth** | **1 Bit** | **2 Bit** | **4 Bit** |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **Example** | | | |
| **Description** | Stores only white and black- also known as monochrome | Stores 4 colours | Stores limited colours |
| **No_ of Colours per Pixel** | $2^1 = 2$ | $2^2 = 4$ | $2^4 = 16$ |
| **Colour Depth** | **8 Bit** | **24 Bit** | |
| **Example** | | | |
| **Description** | Close to reality | Very hard to see any difference between reality | |
| **No_ of Colours per Pixel** | $2^8 = 256$ | $2^{24} = 16777216$ | |

### Calculating file size for different colour depths

All the images above are of the same resolution: 300*225 = 67500 pixels

**If the first image uses 1 bit to store the colour for each pixel, then the image size would be :**

Number of Pixels * Colour Depth = Image Size

67500 * 1 bit = 67500 bits

For the second image uses 2 bits to store the colour for each pixel, then the image size would be:

Number of Pixels * Colour Depth = Image Size

67500 * 2 bit = 135000 bits

For a colour depth of 8 bits, how many colours could each pixel store ? $2^8$

To represent 1024 colours per pixel, what colour depth would I need? $2^{10} = 1024$

How many colours can each pixel store if it has a colour depth of 4bits? $2^4 = 16$ colours

How many bits does the colour depth have to be if we want to store 64 colours per pixel? $2^6 = 64$ , therefore ->6

How many bits would an image be that has a size of 20 by 30 pixels, with each pixel able to display 8 colours?

8 colours is 3 bits per pixel as: $2^3 = 8$ colours

h * w * b = 20 * 30 * 3 = 1800 bits

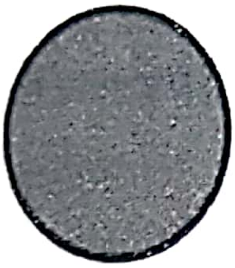
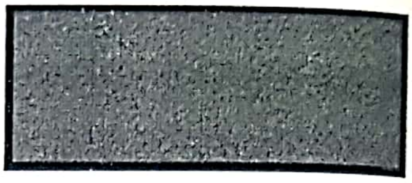
**When might one want to decrease the colour depth for an image?**

When you want to save file space or when you only need a specific palate of colours such a mono-chrome

## 3.5 Vector Images

Vector image describe an image using shapes and geometric forms such as curves, points, circles, rectangles, lines and polygons. A coordinate system is used to represent the shapes. The objects and properties are stored mathematically. Each of these objects has properties to tell you the size, colour, position etc.

A vector image is resolution independent and hence we can enlarge or shrink the image without affecting the output quality.

| | Circle | Rectangle |
|---|---|---|
| Image |  |  |
| **Drawing List** | <circle cx = "100" cy = "100" r = "50" fill = " blue" stroke = "black" stroke-width = "5" /> | <rect x= "14" y = "23" width ="250" height "50" fill= "blue" stroke = "black" stroke-width = "3" /> |
| **Notes** | Centre is defined via cx,cr and r is the radius | X and y gives the tip left start location |

**Use of Vector Image**

- To design logo's and scalable high end graphics

**Example of Popular SVG Format**

**What is SVG?**

SVG stands for Scalable Vector Graphics and is used to define vector-based graphics for the Web. It defines the graphics in XML format and does NOT lose any quality if they are zoomed or resized. Every element and every attribute in SVG files can be animated. SVG is a W3C recommendation.

**SVG Advantages**

**Advantages of using SVG over other image formats (like JPEG and GIF) are:**

- ✓ SVG images can be created and edited with any text editor
- ✓ SVG images can be searched, indexed, scripted, and compressed
- ✓ SVG images are scalable
- ✓ SVG images can be printed with high quality at any resolution
- ✓ SVG images are zoomable (and the image can be zoomed without degradation)
- ✓ SVG is an open standard
- ✓ SVG files are pure XML

The main competitor to SVG is Flash. The biggest advantage SVG has over Flash is the compliance with other standards (e.g. XSL and the DOM). Flash relies on proprietary technology that is not open source.

SVG images can be created with any text editor, but it is often more convenient to create SVG images with a drawing program, like Inkscape. To create an SVG, you need to add the tags **<svg xmlns="http://www.w3.org/2000/svg">** at the beginning and **</svg>** at the end.

- Copy the following into a text file and save it as **image.svg**

```
<svg xmlns = “http://www.w3.org/2000/svg”>
    <rect width = “100” height = “80” x = “0” y = “0”  fill = “green” />
    <line x1="5" y1="5" x2="250" y2="96" stroke="red" />
    <circle cx = “90” cy = “80” r= “50” fill = “blue” />
    <text x = “180” y = “60” > sample Text </text>
</svg>
```

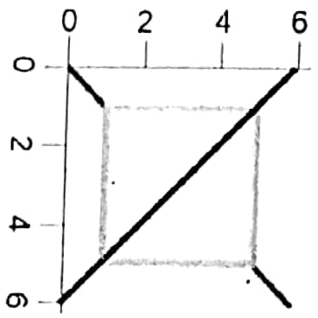
Once you have saved this, drag it into a browser window and you should see some shapes. SVGs are very powerful and you can attach code to their structure, making them interactive.

Questions

1. **Give some objects that can be used in vector graphics:**
    - Line
    - Text
    - Rectangle
    - Circle
2. **Give the properties needed to display a rectangle:**
    - x,y
    - width, height
    - fill
    - stroke (colour), stroke-width
3. **Give the properties needed to display a line:**
    - x1,y1 - start coordinates
    - x2,y2 - end coordinates
    - width
    - fill
    - stroke (colour)
4. **What would the following drawing list produce?**

```
<line x1="0" y1="0" x2="6" y2="6" stroke="red" />
<rect width="4" height="4" x="1" y="1" fill="yellow" stroke="green" stroke-width=1 />
```

`<line x1="6" y1="0" x2="0" y2="6" stroke="black" />`



3.6 Comparison between Bitmap and Vector Graphics

| Bitmap Images | Vector Images |
|---|---|
| ▪ Bitmap images scale resulting in file size increase / decrease<br>▪ Bitmap images distort (pixelate) when scaling<br>▪ Bitmaps are better for photo editing<br>▪ Bitmaps require less processing power to display | ▪ Vector images scale without file size increase / decrease<br>▪ Vector images scale without distortion to the image |

**Advantages of vector images over bitmapped images for displaying images on the Web**

- ✓ Scalable
- ✓ Efficient

**Why is the Adobe Flash .swf file format becoming a less appropriate way of presenting graphical content on the Web?**

- ✓ Not supported by iOS and Android, and mobile is growing
- ✓ Improving support for SVG, which is a W3C standard

**Image file format best suited to presenting photographic images on the Web.**

- ✓ JPEG

JPEG

**Joint Photographic Experts Group** format is used to display photographs and other continuous-tone images on the web

JPEG format supports CMYK, RGB (millions of colours), and Grayscale colour modes. Unlike GIF format, JPEG retains all colour information in an RGB image but compresses file size by selectively discarding data.

### Compression Type

Lossy compression and this results in a loss of quality if a high level of compression is applied. In most image editing programs you can specify how much compression or loss of quality you want.

### Difference between GIF and JPEG

- Another important difference between GIF and JPEG is that JPEG **does not preserve** transparency.

### Image file format best suited to presenting non-photographic images on the Web

✓ **PNG**

### PNG

**Portable Network Graphics (PNG)** is a little bit like the best of both worlds.

### Compression Type

Lossless compression and for display of images on the web.

Unlike GIF, PNG supports 24 bit images and produces background transparency without jagged edges; however, some older web browsers do not support PNG images.
PNG format supports RGB, Indexed Colour, Grayscale, and Bitmap mode images. PNG also preserves transparency in grayscale and RGB images.

## 4.0 Information Representation - Sound

### 4.1 Sound

Sound is an oscillation of pressure transmitted through a solid, liquid, or gas (there is no sound in outer space as space is a vacuum and there is no solid, liquid or gas to transmit sound through!).

Sound is often recorded for two channels, stereo; feeding a left and right speaker whose outputs may differ massively. Where one channel is used, this is called mono. 5.1 surround sound used in cinemas and home media set ups use 6 channels.

Sound waves in nature are continuous; this means they have an almost infinite amount of detail that you could store for even the shortest sound. This makes them very difficult to record perfectly, as computers can only store discrete data, data that has a limited number of data points.

Human can hear sounds in the frequency range of 20Hz – 20, 000 Hz

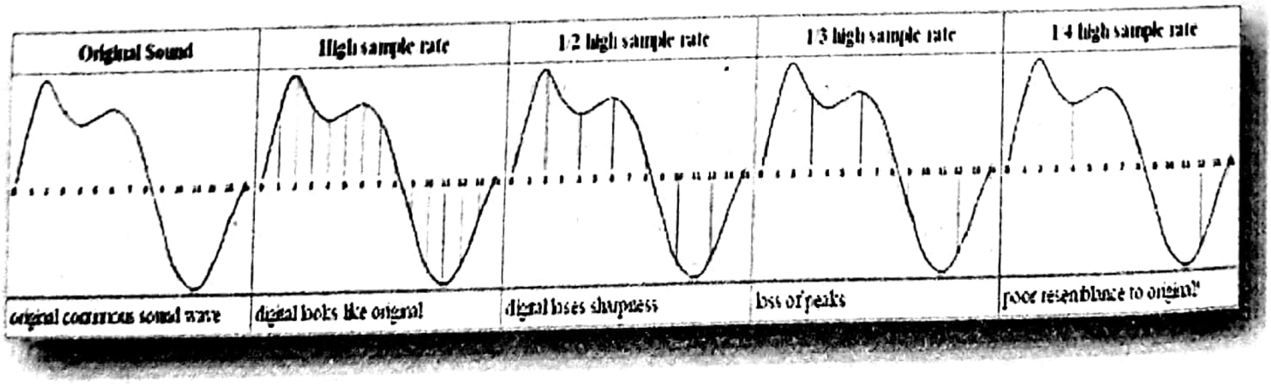
## 4.2 Digital Sound

It is recorded often using a microphone which converts analogue sound into digital format by altering the output signal from the device. Hence, to store the sound file digitally, the voltage is sampled at frequent intervals (such as 48 KHz)) and stored in binary code usually in 16 or 32 bits per sample.

To create digital music that sounds close to the real thing you need to look at the analogue sound waves and try to represent them digitally.

This requires you to try to replicate the analogue (and continuous) waves as discrete values. The first step in doing this is deciding how often you should sample the sound wave, if you do it too little, the sample stored on a computer will sound very distant from the one being recorded. Sample too often and sound stored will resemble that being recorded but having to store each of the samples means you'll get very large file sizes.

To decide how often you are going to sample the analogue signal is called the sampling rate.



## 4.3 Sound Terminologies

**Sampling Rate** - The number of samples taken per second

**Hertz (Hz)** - The SI unit of frequency defined as the number of cycles per second of a periodic phenomenon

**Sampling Resolution** - The number of bits assigned to each sample

The sampling resolution allows you to set the range of volumes storable for each sample. If you have a low sampling resolution then the range of volumes will be very limited, if you have a high sampling resolution then the file size may become unfeasible. The sampling resolution for a CD is 16 bits used per sample.

**File sizes**

- ✓ **Bit rate** - the number of bits required to store 1 second of sound

To work out the size of a sound sample requires the following equation:

File Size = Sample Rate * Sample Resolution * Length of sound

Or

File Size = Bit Rate * Length of sound

**Sound File Sizes**

If you wanted to record a 30 second voice message on your mobile phone you would use the following:

**Sample Rate** = 8,000Hz, **Sample Resolution** = 16 bit
**Length of Sound** = 30 seconds

Therefore the total file size would be: 8,000 * 16 * 30 = 3 840 000 Bits = 480 000 Bytes

## 4.4 Sound Editing

If you are interested in sound editing you can start editing your own music using a program called Audacity. Using Audacity you can create your own sound samples with different sample rates and sample resolutions, listening to the difference between them and noting the different file sizes.

**Sampled sound**

**Why might a digital representation of a sound struggle to be a perfect representation?**

A sound wave is continuous data, whilst digital data is discrete and the representation is an approximation of the original

**Why might you choose to have a lower sampling rate than a higher one for storing a song on your computer?**

The higher the sampling rate the more data is needed to be stored, meaning the larger the file size.

**What is the sampling resolution?**

The number of bits assigned to each sample, effecting the range of volumes that can be stored in a sample

**What is the equation to work out the bit rate of a song?**

Sampling Rate * Sampling Resolution

For the following sound sample work out its size:

- ✓ **Sample Rate** = 16,000Hz
- ✓ **Sample Resolution** = 8 bit
- ✓ **Length of Sound** = 10 seconds
- ✓ **Answer** : 16,000 * 8 * 10 = 1 280 000 Bits

Work out the sample rate of the following sound file:

- ✓ **Sound File** = 100,000 bits
- ✓ **Sample Resolution** = 10 bit
- ✓ **Length of Sound** = 5 seconds
- ✓ **Answer** : 100,000 / (10 * 5) = 2,000Hz

Why might a song recorded with the following settings:

- ✓ **Sample Rate** = 22,000Hz
- ✓ **Sample Resolution** = 16 bit
- ✓ **Length of Sound** = 10 seconds
- ✓ Have a file size of 7,040,000 bits?
- ✓ **Answer**: The file might be recorded in stereo, meaning twice the amount of data would have to be stored

## 4.5 Sound Compression

As you can see we have some serious issues with the size of sound files. Take a look at the size of a 3 minute pop song recorded at a sample rate of 44 kHz and a sample resolution of 16 bits.

44,000 * 16 * 180 = 126 720 000 bits (roughly 15 MB)

As you are probably aware an mp3 of the same length would be roughly 3Mb, a fifth of the size. So what gives?

It is easy to see that the raw file sizes for sounds are just too big to store and transmit easily, what is needed it a way to compress them

### 4.5.1 Lossless

**Lossless compression** - compression doesn't lose any accuracy and can be decompressed into an identical copy of the original audio data.

**Example of Lossless compression**

- WAV files don't involve any compression at all
- FLAC compress the WAV file into data generally 50% the original size.

To do this it uses run length encoding, which looks for repeated patterns in the sound file, and instead of recording each pattern separately, it stores information on how many times the pattern occurs in a row.

Let us take a hypothetical set of sample points:

000000000000000000000123454321000000000000000000012345678765678987 6

As you can see the silent area takes up a large part of the file, instead of recording these individually we can set data to state how many silent samples there are in a row, massively reducing the file size:

(21-0)123454321(17-0)123456787656789876

Another technique used by FLAC files is linear prediction.

### 4.5.2 Lossy

FLAC files are still very large, what is needed is a format that allows you to create much smaller file sizes that can be easily stored on your computer and portable music device, and easily transmitted across the internet.

**Lossy compression** - compression loses file accuracy, generally smaller than lossless compression

**Example of Lossy compression**

- MP3
- AAC
- OGG

As we have already seen, to make smaller audio files we can decrease the sampling rate and the sampling resolution, but we have also seen the dreadful effect this can have on the final sound.

**How it works?**

The compression works by reducing accuracy of certain parts of sound that are considered to be beyond the auditory resolution ability of most people.

This method is commonly referred to as perceptual coding. It uses psychoacoustic models to discard or reduce precision of components less audible to human hearing, and then records the remaining information in an efficient manner.

Because the accuracy of certain frequencies are lost you can often tell the difference between the original and the Lossy versions by being able to hear the loss of high and low pitch tones.

# 5.0 Information Representation - Video

## 5.1 Videos

Video is defined as an electronic medium for the recording, copying, playback, distribution, display of moving visual and audio media. The world of video has evolved from CRT to more advance modern technologies. Video data may be represented as a series of still image frames.

## 5.2 Video Terminologies

1. **Frame rate:** is the number of still images (or frames) recorded every second.

Usually frame rate is expressed in frames per second (fps) and most video cameras record at 30 fps.

2. **Resolution:** is how many pixels the image has.

Resolution is usually expressed by numbers for horizontal and vertical: 640 by 480 means 640 pixels wide, by 480 pixels tall. Multiply the numbers and you get the total number of pixels. In this case 640x480 = 307,200.

3. **Aspect Ratio:** is what defines the width and height of your images

The most common aspect ratios are 3:2, 4:3, and 16:9.

4. **Compression and Format:** is to save space the movie gets compressed to make it smaller.

The way a camera compresses the image data and records it is the recording format.

## 5.3 Characteristics of Video Streams

**Frame Rate**

It is defined as the number of still images per second. Usually it ranges from 6 -8 for old camera to 120 or more for modern cameras.

**Common Standards:**

- PAL
- SECAM

- NTSC

## 5.4 Interlaced V/S Progressive encoding

### 5.4.1 Interlaced

Interlacing was designed to reduce flicker in early video displays and devices without incrementing the number of frames per second, which would have sacrificed image details to remain within the limitations of narrow bandwidth.

How it works?

It uses the horizontal scan lines of each complete frame with them being treated as numbers consecutively and captured as two fields (Upper field-consist of odd numbered lines, lower field – consist of even numbered lines).

Hence, analogue display devices reproduce the frame rate similarly and even doubling the frame rate to overcome issues of flicker.

Images capture devices acquire the fields one at a time and doubles the frame rate motion. This results in smoother real life details of rapidly moving parts of the image when viewed on an interlaced CRT device but causes problems when displayed on a progressive scan device.

*Examples of Interlaced Formats:*

✓ NTSC ,PAL and SECAM

Sometimes the letter I is added to video resolution to indicate interlacing. E.g. PAL video format -? 576i50 where 576 is the total number of horizontal lines and 50 indicates the fields (half frames) per second.

### 5.4.2 Progressive Scans

In such systems, during each refresh period, all the scan lines in each frame in a sequence is updated.

It results in optimum spatial resolution of both stationary and moving parts of the image when displaying a recorded signal or natively progressive broadcast.

It results in degradation of spatial resolution when natively interlaced signals are displayed. Flickers and "comb" effect is often seen in moving parts of pictures unless the signal is processed in a special way. The latter is known as de-interlacing which results in optimisation of display of an interlaced video from a source such as DVD, analogue or satellite source on devices like LCD Television, digital video projector or plasma panel.

Note: Despite using de-interlacing, the produced video quality does not match true progressive scan source materials.

## 5.5 Video Compression

Video compression is the practical implementation of source coding in information theory and is used to reduce redundancy in video data.

In practice, most video codecs also use audio compression techniques in parallel to compress the separate, but combined data streams as one package

We need to compress video (and audio) in practice since:

a) Uncompressed video (and audio) data are huge. Hence, it causes big problems for storage and network communications.
b) Lossy methods have to employ since the *compression ratio* of lossless methods (e.g., Huffman, Arithmetic, and LZW) is not high enough for image and video compression, especially when distribution of pixel values is relatively flat.

### 5.5.1 Compression Algorithm

#### 5.5.1.1 Interframe

It is one of the most powerful technique for video compression. The latter uses one or more earlier or later frames in a sequence to compress the current frame.

**How it works?**
Simply each frame is compared with the previous one and copies unchanged portion into the next one. If part of the frame moves in a simple manner, the compressor issues command to the decompressor to manipulate the frame. (Shift, rotate, lighten or darken the copy)

**Use**

- Interframe suits well for program that will playback video but poses several issues if the video sequence needs to be altered.

As Interframe algorithm copies data from one frame to another, if original frame is cut out, then the consequent following frames cannot be reconstructed properly.

**Commonly used compression types**

a) Spatial Redundancy Removal - Intraframe coding (JPEG)
b) Spatial and Temporal Redundancy Removal - Intraframe and Interframe coding (H.261, MPEG)

### 5.5.1.2 Spatial Redundancy

It is defined as the redundancy of information that exists within the same frame (also known as intra frame redundancy). There are similarity between the pixel values between frames within the space of the given frame. Hence, a frame contains pixels which have similar or near similar values to their adjacent neighbours.

If this is exploited, predictions can be made about adjacent neighbours with acceptable accuracy. Eventually, this correlation means we have lesser information to be encoded and transmitted. As such we have video frame compression.

If the picture section is not complex, then this algorithm is very efficient in compressing a video sequence

E.g. in video telephony, the adjacent pixels in the face or eye region have similar values, there is minimal (numerical) difference among them which means there is lot of spatial information that can be reduced

### 5.5.1.3 Temporal Redundancy

This technique looks for differences between consecutive frames. If most of an image in two frames hasn't changed, why should we waste space to duplicate all of the similar information?

Spatial compression removes redundant information within a frame.

For instance, a line compression algorithm, instead of representing a white line as a series of dots with individual colour info, it can represent it as how many dots of white colour (saving storage space)

This problem is essentially the same as that faced when compressing still images.

## 5.6 Data Compression

It is important that data be represented efficiently for two reasons:

a) **storage of information**

To reduce the bandwidth required for transmission and also to reduce storage requirements (video, audio and speech)

b) **transmission of data**

To reduce the volume of data to be transmitted (text, fax, images)

Some common text compression techniques:

- ✓ keyword encoding
- ✓ run-length encoding
- ✓ Huffman encoding

Compression can be classified as:

i. **Lossless compression**

This technique is normally used for legal and medical documents, computer programs etc. It exploit only data redundancy and is not as effective as Lossy compression.

ii. **Lossy compression**

This technique is normally used for digital image, audio and video where some errors or loss can be tolerated. It basically exploit both data redundancy and human perception properties

iii. **Constant bit rate versus variable bit rate coding**

### 5.6.1 Keyword Encoding

In this encoding technique, frequently used words are replaced with a single character. For example:

| Word | Symbol |
|---|---|
| and | + |
| many | $ |
| as | ^ |
| the | ~ |
| it | # |
| in | & |
| is | % |

***Consider the following:***

*Computer Science is merging as the most important field for a nation development in the current world. As it is right now many countries are adapting the curriculum to inculcate the essence of this subject in their education system. As it is, England has been the first one to gear its educational approach in this sense and is being followed by many developed countries.*

**Encoded Text**

Computer Science % merging ^ ~ most important field for a nation development &~ current world. ^ # % right now $ countries are adapting ~ curriculum to inculcate ~ essence of this subject & their education system. ^#%, England has been ~ first one to gear its educational approach & this sense + % being followed by $ developed countries.

- ✓ Original Text (Characters + Space + Punctuation) = 371
- ✓ Encoded Text (Characters + Space + Punctuation) = 341
- ✓ Compression Ratio = 341/371 ~= 0.9

**Note:** The characters we use to encode cannot be part of the original text.

### 5.6.2 Run-Length Encoding

In large data streams, it often occurs that a single character may be repeated over and over again in a long sequence.

In run-length encoding, a sequence of repeated characters is replaced by a *flag character*, followed by the repeated character, followed by a single digit that indicates how many times the character is repeated.

Example: CCCCCCC would be encoded as: *C7

*A6*B5*C7*Z8 would be decoded into the following original text:

AAAAAABBBBBCCCCCCCZZZZZZZZ

The original text contains 26 characters, and the encoded string contains 12 characters, giving us a compression ratio in this example of 12/26 or approximately 0.46

Since we are using one character for the repetition count, it seems that we can't encode repetition lengths greater than nine. Instead of interpreting the count character as an ASCII digit, we could interpret it as a binary number.

It is to be noted that:

- The repeated occurrence of the same character is called a run
- The number of repetition is called the length of the run
- The run of any length is represented by three characters

eeeeeee7tnnnnnnnn

@e7t@n8

### 5.6.3 Huffman Encoding

In this technique, fewer bits are assigned to symbols that appear more often and more bits to the symbols that appear less often

Huffman codes using variable-length bit strings to represent each character. Some characters may be represented by five bits, and another few by six bits, and yet another few by seven bits, and so forth.

It is very efficient when occurrence probabilities vary widely

The Huffman codebook from the set of symbols and their occurring probabilities

Two properties:

- ✓ generate compact codes
- ✓ prefix property

Consider the Following Huffman Code:

| Huffman Code | Character |
|---|---|
| 00 | S |
| 01 | N |
| 10 | I |
| 110 | A |
| 111 | R |
| 1010 | T |
| 1011 | M |
| 1111 | O |

The word TRANSMISSIONS would be encoded in binary as

: 1010 111 110 01 00 1011 10 00 00 10 00 01 00

Let's say we use Huffman encoding for the above string, it is 32 bits long.

If we use a fixed size string of let's say 8 bits to represent each character, the binary form of original string would have been : 13 *8 = 104 bits

Compression ratio

: 32/ 104 =~ 0.31

**An important** characteristic of any Huffman encoding is that no bit string used to represent a **character** is the prefix of any other bit string used to represent a character.

# 6.0 Processor Fundamentals – CPU Architecture

## 6.1 Stored Program Concept

This concept states that a program must be in main memory for it to be executed. The instruction is then fetched, decoded and executed one at a time. One should remember that the memory is divided into a set of contiguous locations which is accessed by the processor to read and write data.

## 6.2 Computer Architecture and the Fetch-Execute Cycle

### 6.2.1 Von Neumann Architecture

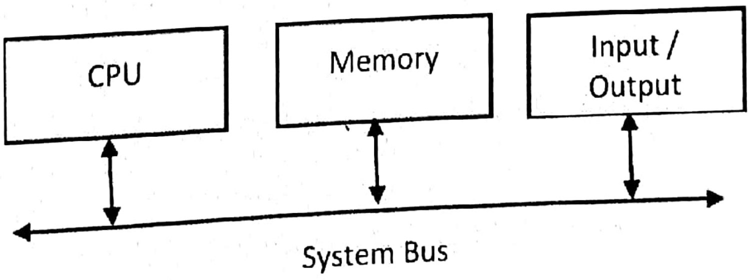
John von Neumann introduced the idea of the stored program and realised that data and programs are indistinguishable and can therefore use the same memory.

Von Neumann architecture key aspects and features are:

- Applies to single processor
- Program consists of a sequence of stored instructions
- Instructions and data are stored in a continuous block of main memory
- instructions are executed in sequence

In a Von Neumann machine:

All data and instructions are stored in the Main Memory. Instructions are sent to the Processor along the System Bus to be executed. Any input and output (such as printing and entering instruction) is performed by I/O devices with the data travelling from the I/O devices to the Processor and Main Memory by means of the System Bus



It follows a linear sequence of **fetch–decode– execute** operations for the sequence of instructions which make up the program. In order to do this, the processor has to use some special registers.

### 6.2.1.1 System Bus

A bus is a connection between different devices. This connection will normally consist of multiple wires along which signals, instructions and data will be carried. In Von Neumann Architecture there is a single bus to manage the connection between the three main components. The System Bus consists of 3 separate buses, each with a specific task that you need to know.

#### 1. Address Bus

It carries the address of the main memory location or input/output device which is about to be used. It also sends address values in only one direction, i.e. from the processor to the memory address register (MAR).

#### 2. Data Bus

Data bus is used to carry the data that needs to be transferred between hardware. The memory data register (MDR) is at one end of the data bus which is bi-directional. (I.e. allows processor instruction set to read values from main memory and store values in memory)

#### 3. Control Bus

Control bus is used to send control signals from the control unit to the other components of the system. A separate wire is dedicated to a particular control signal, for example:

- a completed data transfer (read or write) operation
- interrupt request

### 6.2.1.2 ALU (Arithmetic Logic Unit)

ALU Performs arithmetic (ADD, SUBTRACT), and logical (AND.OR, NOT) comparison operations. It handles all calculations.

### 6.2.1.3 CU (Control Unit)

It controls the flow of instruction from memory to CPU, coordinates the I/O devices of the computer and directs the operation of other components via control signals.

## 6.3 Clock

The processor clock is a timing device that synchronises when the fetch, decode execute cycle runs. A computer may contain several clocks each one used to regulate a particular thing. The CPU does no work until told to—even though data may be on the External Data Bus. You need a buzzer to tell the processor to start. This is referred to as a clock which is actually a stream of pulses.

### 6.3.1 Clock Speed

It is defined as the number of cycles that are performed by the CPU per second. Clock speed is the CPU's maximum speed, not the speed at which the CPU must run. It is used to synchronize eternal and external activity.

One cycle per second = 1 hertz (Hz); 1 million cycles per second = 1 megahertz (MHz)

Every command requires at least two clock cycles. Clock speed is the maximum clock cycles per second. A clock cycle is the time taken by the special wire to charge. The CPU needs at least two clock cycles to act on each command. A cycle is one complete up-and-down segment of the sine wave.

Clock Speeds

- 1 hertz (1 Hz) = 1 cycle per second
- 1 megahertz (1 MHz) = 1 million cycles per second
- 1 gigahertz (1 GHz) = 1 billion cycles per second
- Intel 8088 ran at 4.77 MHz
- Modern CPUs run at 3+ GHz ( core i5,core i7)

## 6.4 Increasing the performance of our computer

There are 3 things that can boost our computer's performance:

✓ **Increase the word length**

- Word Length or word size is the number of bits that processor can process at one time.
- The larger the word size, the larger and precise instructions it can handle.
- Some specific game console even exist with word size 128 bits. E.g. IBM system /370 is considered as 128 bit

✓ **Increase the clock speed**

- With a faster clock speed, the CPU would be forced to perform more instructions per second. .e.g. Clock speed 3 GHz is 3 times faster than 1GHz.
- However, note by increasing the clock speed, the processor uses more current and eventually it gets hotter and can even be damaged.
- Gamers often use the feature of overclocking their processor for processor intensive games but use high quality heat sink for cooling (liquid cooling).

✓ **Increase the bus width**

- It is defined as the number of bits of information that a bus can carry at one time.
- If your processor bus width was 64 bit but your other bus 32 bit it would lead to a bottle neck, hence the larger the bus the faster is our computer.
- Modern processor are 64 bit based while few are still 32 bit.
- e.g. Intel Itanium 2 processor system bus is 128 bit

## 6.5 Ports

These are method through which input or output is performed between the processor and peripheral devices in the computer. Common ports that are used are USB (Universal Serial Bus), RJ (Registered Jack), Firewire etc.

## 6.6 Register

A register is simply a location that can store data. Registers can be thought of in the same way as individual locations in memory in that they store data, although memory locations have no special purpose. The registers are outside the immediate access store and consequently allow faster access to the data they store.

### 6.6.1 List of Registers

| | |
|---|---|
| ✓ Program Counter | **(PC)** - it is also called Sequence Control Register **(SCR)** |
| ✓ Current Instruction Register. | **(CIR)** |
| ✓ Memory Address Register. | **(MAR)** |
| ✓ Memory Data Register. | **(MDR)** – it also called Memory Buffer Register **(MBR)** |
| ✓ Accumulator. | **(ACC)** |
| ✓ Index Register | **(IR)** |

### 6.6.2 Register Purpose and use during the Fetch Execute Cycle

#### Program Counter (PC)

PC Contains the address of the next instruction to be fetched. It passes address to the MAR and is then incremented. The contents are altered to the operand of the instruction if a jump instruction.

#### Current Instruction Register (CIR)

The CIR holds the current instruction. The instruction is divided into the op-code and operand. This register holds the instruction while the op-code is decoded and then sends the address to the MAR.

#### Memory Address Register (MAR)

MAR stores the address of the memory location to be used next. The value or address in the PC is loaded into the MAR to show the address of the instruction to be fetched. The address or operand of the current instruction in the CIR is loaded into the MAR.

#### Index Register (IR)

Index Register modifies the address held in the CIR by the addition of the contents of IR or an integer value. It is used in indexed addressing.

#### Memory Data Register (MDR)

During the fetch stage the contents of the address in the MAR is copied into the MDR. The contents of MDR is the instruction to be used (this is copied to the CIR).During the execute stage (store instruction) the contents of the Accumulator is copied into the MDR. During the execute stage (load/add instruction) the contents of the memory location is copied into the MDR

#### Accumulator

Accumulator hold the data currently being processed. Result of calculation is held in accumulator before being passed to memory unit.

## Stages of the Fetch-Execute Cycle

**Sequence of steps describing the fetching and execution of a single program instruction**

### FETCH STAGE

1. Copy the address that is in the program counter (PC) into the memory address register (MAR).
2. Increment the PC for the next fetch instruction
3. Load the instruction that is in the memory address given by the MAR into the memory data register (MDR).
4. Load the instruction that is now in the MDR into the current instruction register (CIR).

### DECODE STAGE

5. Identify the type of addressing being used by the instruction
    - I. If the address is a direct address, load a copy of the address into the MAR and retrieve the contents of the address.
    - II. If the address is an indexed address, add the address to the contents of the index register (IR) and copy the result to the MAR. Retrieve the contents of this address.
6. Decode the instruction.

### EXECUTE STAGE

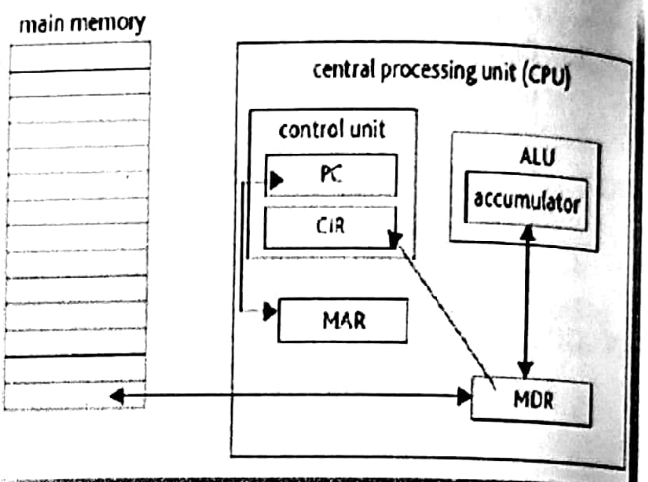
7. If the instruction is a jump instruction then
    - I. Load the address operand into the PC.
    - II. Go to step 1.
8. Execute the instruction.
9. Go to step 1.

## 6.8 Pseudocode Version of Fetch Execute Cycle

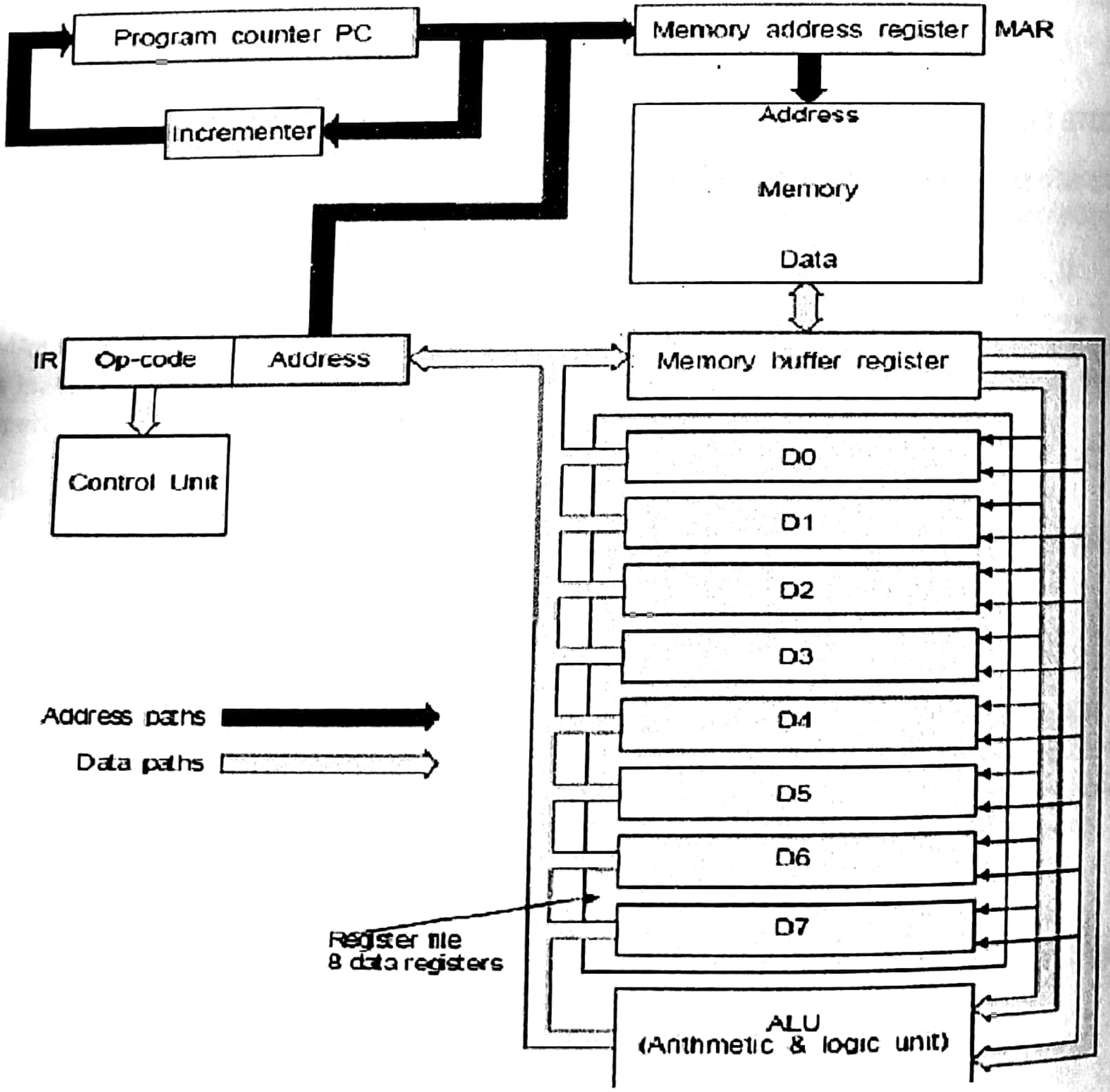
**Note:** ← means assignment operator

```
1.  MAR  ←  [PC]
2.  PC   ←  [PC] + 1
3.  MDR  ←  [[MAR]]
4.  CIR  ←  [MDR]
5.  Decode
6.  Execute
7.  Go to Step 1
```

·Typical Diagram of processor

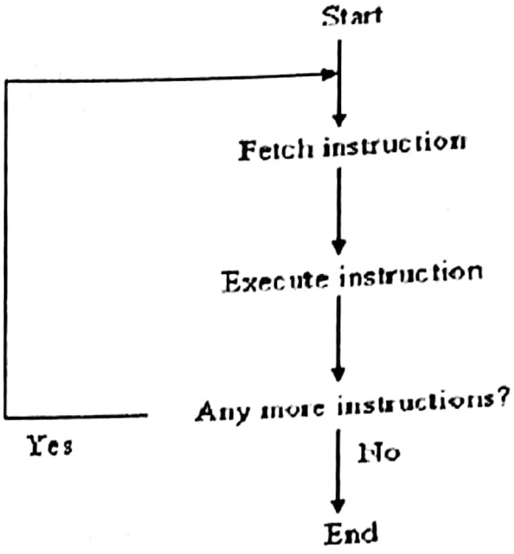


## 6.9 Detailed Diagram of Processor



## 6.10 Interrupts

An interrupt is a signal sent to the processor by devices or programs indicating that they require the attention of the processor.



### 6.10.1 Types of Interrupts

I/O or Hardware interrupt

- ✓ Generated by hardware (I/O devices) to signal a service request (allows resolution of speed mismatch implications) or an error has occurred.
- ✓ E.g.
  - ○ Printer has emptied the buffer and wants it refilled. (I/O)
  - ○ Printer is out of paper or is not connected. (I/O)
  - ○ Close down safely in the event of a power failure from: (HW)
    - An Uninterruptible Power Supply (UPS) which took over when the main electricity was lost but now its rechargeable power supply is running low.
    - The device which measures how much power is still left in a laptop battery informs the processor it is running low.

Timer Interrupt

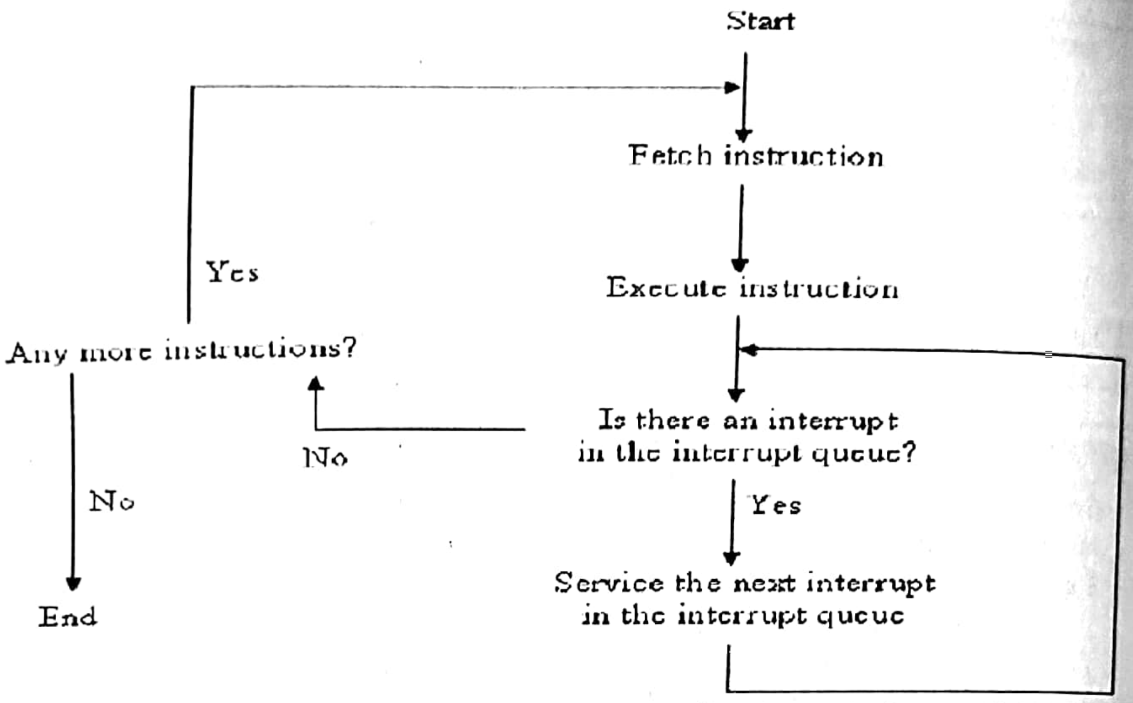
- ✓ Generated at fixed intervals.
- ✓ Allows for display refresh and to control access to processor in multi-access or multi-programming system.

Program interrupt

- ✓ Generated due to an error in a program:
  - Violation of memory use (trying to use part of the memory reserved by the OS for other use).
  - An attempt to execute an invalid instruction (e.g. division by zero).

## 6.11 Handling of Interrupt by OS

**How can the interrupted program be resumed from where it left off when the interrupt has been serviced?**


Start
Fetch instruction
Execute instruction
Yes
Any more instructions?
No
No
Is there an interrupt in the interrupt queue?
Yes
Service the next interrupt in the interrupt queue
End

The contents of all the registers (small, permanent storage locations within the CPU used for a particular purpose) in the processor must be saved so that the processor can use them to service the interrupt but can load their contents back in to resume the job being interrupted once the interrupt has been serviced.

**What happens if the current job is more important than the interrupt?**

Priority of interrupt compared with current job.

**If higher:**

Interrupt is serviced by the processor.

**If lower:**

Place interrupt in a queue. Current job continues with next cycle.

**What happens if some interrupts are more important than others?**

The Interrupt is allocated a position in job queue according to priorities. Now how this priority is decided is called scheduling.

## 6.12 Handling of Interrupts Summary

i. Current cycle is completed

ii. Priority of interrupt compared with current job

iii. If higher:
  - Contents of special registers saved, current job allocated position in job queue according to priorities
  - Interrupt is serviced by OS.
  - On completion values of special registers from original program are loaded and original job restored.

iv. If lower:
  - Interrupt allocated position in job queue according to priorities.
  - Current job continues with next cycle.

# 7.0 Processor Fundamentals – Processor's Instruction Set

## 7.1 Assembly Language

Assembly language is an example of low level language. It is CPU specific and makes direct use of internal registers. It uses Mnemonics to give instruction from its instruction set.

## 7.2 Processor Instruction set

It is the range of instructions that a CPU can execute. They can be grouped according to the king of processing they cover.

| | | |
|---|---|---|
| 1. | Arithmetic | (ADD,INC,DEC) |
| 2. | Data Transfer | (IN,OUT) |
| 3. | Logic | (OR,AND,XOR,NOT) |
| 4. | Jump | (JPE,JPN,JMP) |
| 5. | Compare Instructions (CMP) | |

**Fawad Khan 0321-6386013**

The instruction are made of two main parts which are:

I. **Opcode** - instruction name
II. **Operand** - data or address

E.g. of a Machine instruction

LDM # 69

LDM # 69
**Opcode** **Operand**

LDM #
**Operation** **Addressing mode**

The number of bits allocated for opcode and operand depends on the word size. Some processor have lots of different instructions and a smaller operand **(Intel, AMD -> used in PC)** while others have less instructions and more space for operand **(ARM – used in tablet and mobile phones)**

| Instruction | | Explanation |
|---|---|---|
| OpCode | Operand | |
| LDM | #n | **Immediate addressing.** Load the number n to ACC |
| LDD | <address> | **Direct addressing.** Load the contents of the given address to ACC |
| LDI | <address> | **Indirect addressing.** The address to be used is at the given address. Load the contents of this second address to ACC |
| LDX | <address> | **Indexed addressing.** Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC |
| LDR | <address> | **Immediate addressing.** Load the number n to IX |
| STO | <address> | Store the contents of ACC at the given address |
| ADD | <address> | Add the contents of the given address to the ACC |
| INC | <register> | Add 1 to the contents of the register (ACC or IX) |
| DEC | <register> | Subtract 1 from the contents of the register (ACC or IX) |
| JMP | <address> | Jump to the given address |
| CMP | <address> | Compare the contents of ACC with the contents of <address> |
| CMP | #n | Compare the contents of ACC with number n |
| JPE | <address> | Following a compare instruction, jump to <address> if the compare was True |
| JPN | <address> | Following a compare instruction, jump to <address> if the compare was False |
| IN | | Key in a character and store its ASCII value in ACC |
| OUT | | Output to the screen the character whose ASCII value is stored in ACC |
| END | | Return control to the operating system |

**All questions will assume there is only one general purpose register available (Accumulator)**

# ACC denotes Accumulator
IX denotes Index Register
# denotes immediate addressing

B denotes a binary number, e.g. B01001010
& denotes a hexadecimal number, e.g. &4A

## 7.3 Addressing Mode

Addressing mode is defined as the rule for interpreting or modifying the address field of an instruction.

Effective address is the address of operand produced by the application of the rule for interpreting or modifying the address field of the instruction before the operand is actually referenced.

There are 5 main types of addressing mode that we shall consider:

1. Immediate Addressing
2. Direct Addressing / Absolute Addressing
3. Index Addressing
4. Indirect Addressing
5. Relative Addressing

### Immediate Addressing

In this mode, operand is part of the instruction itself. It implies that the operand = address field

E.g. LDM #25 means the accumulator will hold value 25 (operand)

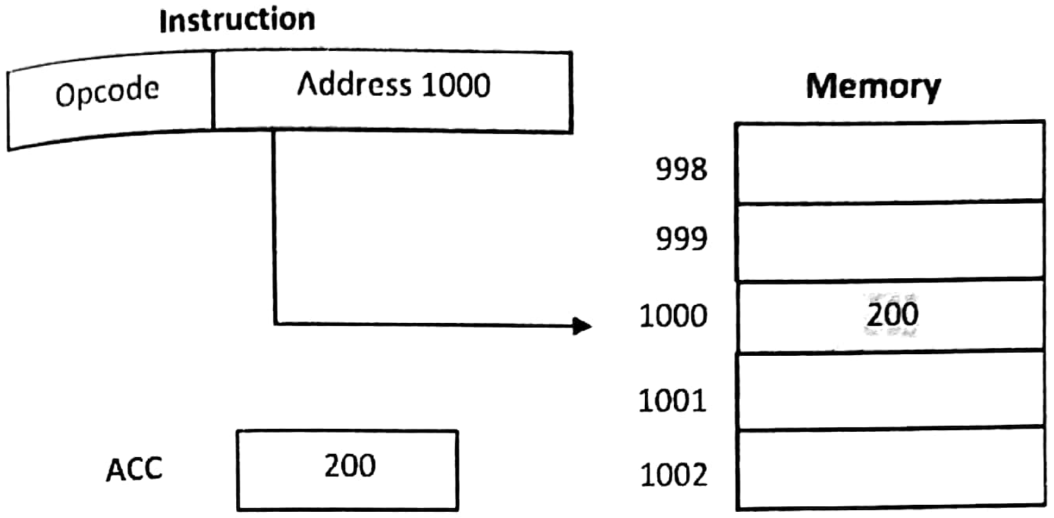
| Advantage | Disadvantage |
|---|---|
| ▪ Since there is no memory reference to fetch data , it is fast | ▪ It has a limited range |
| **Use** : It might be used to simulate constants | |

### Direct Addressing / Absolute Addressing

In this mode, address field contains address of operand. Effective address (EA) = address field (A). E.g. LDD 1000 means that the value 1000 in the address part of the machine code instruction is the address of the data.

| Advantage | Disadvantage |
|---|---|
| ▪ Memory is accessed only once to get the operand.<br>▪ Single memory reference to access data<br>▪ No additional calculations to work out effective address<br>▪ Simple to use | ▪ **Limited address space** i.e. it does not allow access to all memory addresses as there are memory addresses larger than can be held in the address part of an instruction. For e.g. in a standard 32 bit word, 8 bits are used for the instruction code (op-code) and 24 bits are used for the address of the data this allows ONLY $2^{24}$ locations in memory to be addressed. |
| Use : One use might be to simulate variables | |

e.g. LDD 1000 will load the value at address 1000 in accumulator

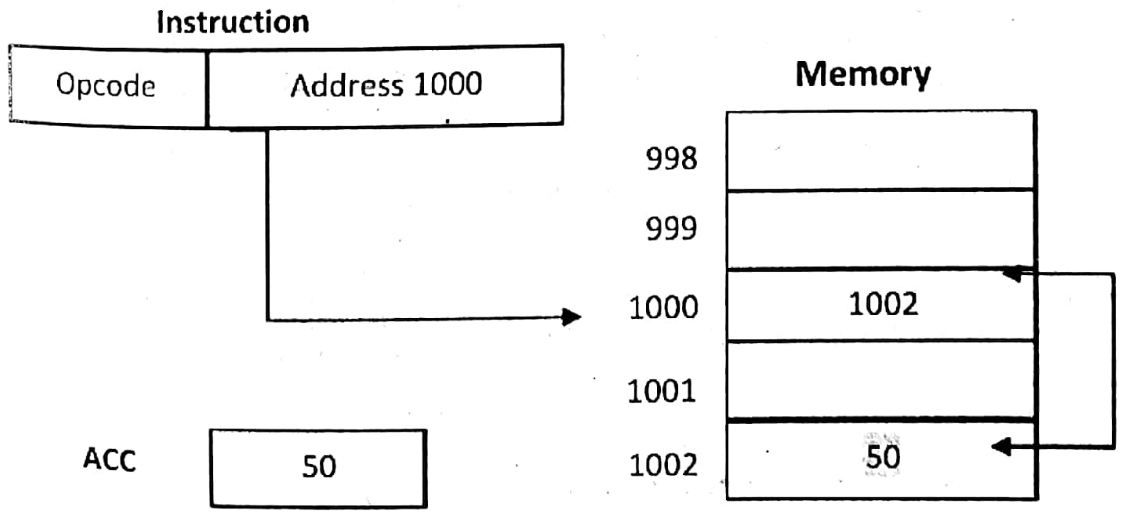


Indirect Addressing

In this addressing mode. The memory cell pointed to by address field contains the address of (pointer to) the operand. Effective address (EA) = address of address .i.e. look into address (A) and look there for operand E.g. LDI 1000

| Advantage | Disadvantage |
|---|---|
| ▪ It offers a large address space of $2^n$ where n is the word length. | ▪ We have to access multiple memory to find operand which makes it slower. |
| Use : It might be used to simulate pointers | |

e.g. LDI 1000 will load the value 50 in ACC



## Indexed Addressing

In this mode, a special register is used to adjust the address part of an instruction (Indexed Register). Effective address (EA) = address + index register. E.g. LDX 998 = LDX (998 + Index Register)

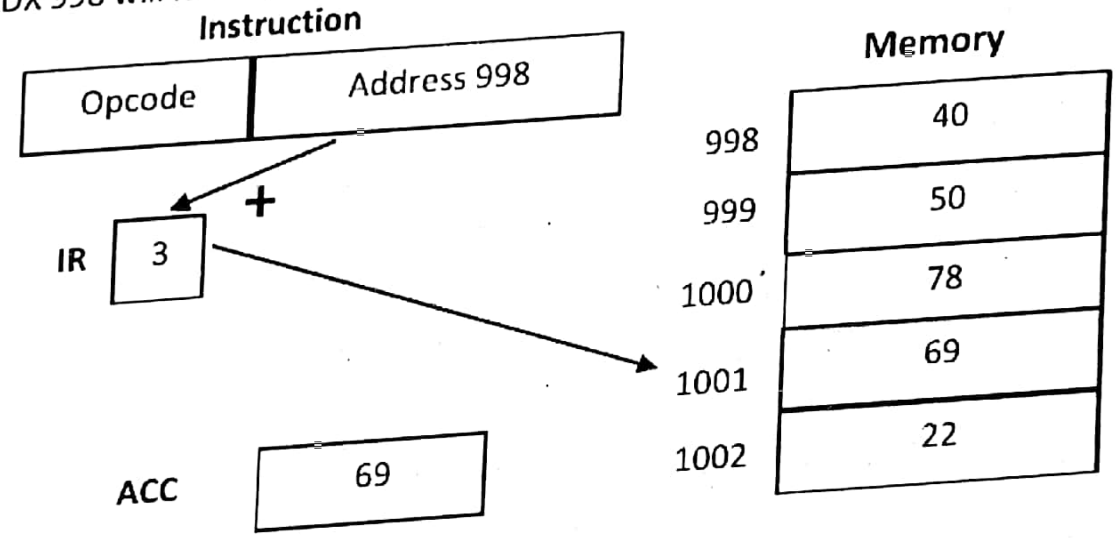
**Advantage**

- ✓ Allows a set of contiguous data locations to be accessed such as an array or compute memory without modifying the instruction. Hence, the address in the instruction doe not change but rather only the content of the Index register.

**Use**

- ✓ It is used to simulate arrays

E.g. LDX 998 will load the value found at (998 + 3) in the accumulator i.e. 69



## Relative Addressing

In this mode, the actual address is the current address + address of PC (Program Counter). It is similar to index addressing except that the PC is used as index register. The address field in the instruction is considered to be a signed integer which is added to the PC to form the address of the operand.

Effective address (EA) = address + Program Counter

**Uses**

- ✓ It might be used for cache usage and locality of reference

e.g. LDA -5

# 8.0 Processor Fundamentals – Assembly Language

## 8.1 Assembly Language

Assembly Language is an example of a LLL. (Low Level Language). It is usually used to program CPU chips and are specific to them. It makes use of registers, mnemonics to program code and uses there different memory modes, and labels to jump from one point to another portion.

| Advantage | Disadvantage |
|---|---|
| ▪ They are very efficient since they are close to machine code<br>▪ It allows excellent control for the processor which is suitable for writing hardware drivers. | ▪ They are difficult to use<br>▪ Programmers need to know the internal architecture of a CPU to code properly |

There are 3 types of statements in assembly language. Usually, one statement should appear on a line.

1. **Executable Instructions**
   - It generates machine code for CPU to execute during run time.
   - It is instructions to tell the processor what to do.
2. **Assembler Directives**
   - It provides information to assembler during translation of programs.
   - It is used to define data, select memory model etc.
   - It should be noted that non – executable directive are not part of instruction set
3. **Macros**
   - These are shorthand notation for a group of statements.
   - It is usually a sequence of instructions, directives or other macros.

Assembly Language instructions have the format:

<Label :> mnemonics <operand> <; comment>

## 8.2 Machine code

Machine code is simply binary instruction such as 1011 1101 0010 1010 is very efficient but is very difficult for human to programs. As such CPU maker provide a set of Mnemonics to use with their processor.

### 8.2.1 Mnemonics

Mnemonics are simply a set of programming instructions that are later translated into machine code by an assembler. The same binary instruction could yield to MOV ACC 32.

In assembly language, mnemonics are the opcode (Operation Code) command such as LDD, ADD etc. A set of assembly instruction set has already been seen above.

## 8.3 Symbolic Addressing

Symbolic addressing is used to use define a symbol for the data item or location in memory instead of an absolute address. Normally, a word is used to represent the actual data.

**Advantage of Symbolic addressing over direct memory references**

- Program in memory is re-locatable ( i.e. we do not particularly care about its absolute location)
- The software becomes more understandable by using symbols

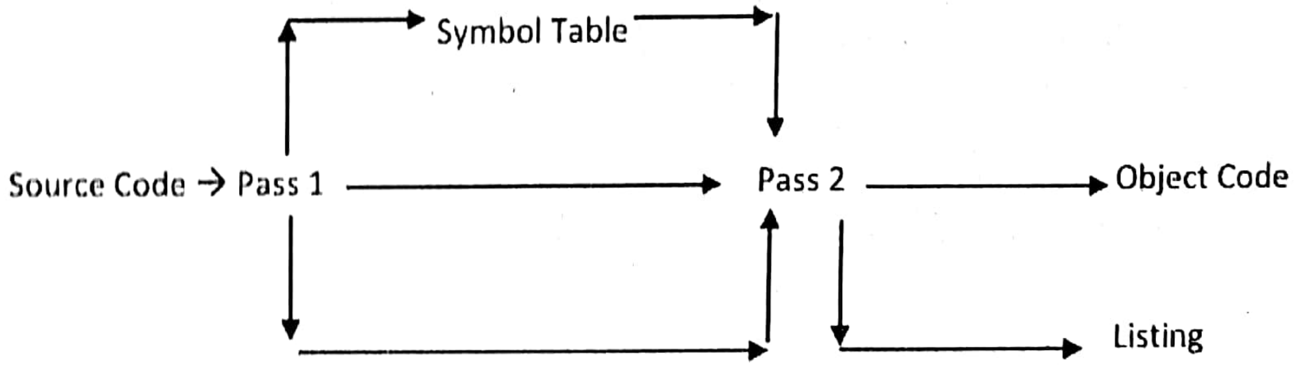
## 8.4 One and two pass assemblers

It is not easy to translate assembly language in one single pass through the ASSEMBLER source code. The problem of forward references arises. The assembler solve this problem by performing two different passes over the user program.

**How it works?**

The goal of a two pass assembler is to draw a symbol table. After the first pass, all user defined identifiers necessary information are recorded in this table. The second pass, allows full assembly to occur by referring to the symbol table whenever it is necessary to determine an address for a named label or the value of a name constant.

During the first pass, errors also are uncovered while in the second pass depends on the first pass to be able to rescan the program.



Assemblers all make heavy use of table.

- The Opcode Translation Table
- The Symbol Table
- The Directive Table
- The String Table

The first pass aim to create a symbol table. The first pass also keeps track of the address entries as it goes through the source code. It then supervise the source handling, lexical and semantic analysis.

The second pass is mainly responsible for code generation while repeating some of source handling and syntactic analysis.

## 8.4 Ethics and Ownership – Computer Ethics

### 8.4.1 Computer Ethics

Ethics is a set of code of conduct laid down by professional bodies so that their member adhere to it. It sets out the standards of behaviour that its members are expected to follow in their professional life.

Computer ethics is needed due to ethical problems aggravated, transformed or generated by computer technology. Computer ethics should be considered as part as the power to trigger development and standards of good practice and conduct in the computing industry and professionals.

**Why do we need ethics?**

The need of a processional code of ethics in computing arise from the fact that software has the potential to do "good" or cause harm or to enable or influence others to do well or cause harm. Every individual has pride in his work and want it to be given recognition and respect. Nowadays software controls many aspects of our lives from safety (Car ABS, Airbags to Air Traffic Control) to Financial (banking, Tax filing) and more.

Many professional organisations have code of professional conduct. They provide guidance for new or young members of the profession who want to behave ethically or those who do not know what is expected of them. There are several organisations for the range of professions included in the computer industry. The main ones are the ACM and IEEE Computer Society. They developed a Code of ethics and Professional Conduct.

One of them is BCS (British Computer Society)

The BCS produced a code of Practice which has four pillars:

- The Public Interest
- Duty to the Relevant Authority
- Duty to the Profession
- Professional Competence and Integrity

The ACM (Association for computing) is one of the body that has its own code of ethics and set of ethical principles approved by the IEEE (Institute of Electrical and Electronic Engineers).

| Advantages | Disadvantages |
|---|---|
| ▪ It sets out the responsibilities and ideals of the profession.<br>▪ It improve the profile of a profession<br>▪ It exerts a de factor regulatory effect which protects both the clients and professionals<br>▪ It provides guidance on acceptable conduct<br>▪ The improves quality and consistency<br>▪ It raises awareness and consciousness of issues | ▪ The standards are obligatory<br>▪ There is the question whether such a code of conduct is desirable or feasible<br>▪ It does not cover all aspects of software development<br>▪ It cannot provide a universal guidance given the heterogeneous nature of the profession<br>▪ It does not provide clear explanation on who determines the violations, how the rules are interpreted and the penalties that exist for violations. |

### 8.4.2 Characteristics of a Code of Ethics

- ✓ These are not simple ethical algorithms that generate ethical decisions
- ✓ Sometimes part of the code may be in conflict or tension with each other or other sources.
- ✓ Hence a good code of ethics provide fundamental principles which require thought rather than blind allegiance.

### 8.4.3 ACM/IEEE Software Engineering Code of Ethics

Software Engineers / Developers contribute directly or by teaching to the analysis, specification, design, development, certification, maintenance and testing of software systems. Due to their roles, software engineers have significant opportunities to do "good" and cause harm. As such, they have to adhere to a Code of Ethics and Professional Practice.

The code contains 8 principles on behaviour and decisions made by software developers or engineers and other software industry actors. Each clauses states some obligations for each individual claiming to be or aspiring to be a software developer. The code expresses the consensus of the profession on ethical issues, it is a means to educate both the public and aspiring professionals about the ethical obligations of all software engineers.

1. Public
   - Software engineers shall act consistently with the public interest

- It means that software developers have to approve software only if it is safe, meets required standards and passes test. The software should not diminish the quality of life, privacy or harm the environment.
- The software developer should disclose any actual or potential danger to the user.
- It should be fair and avoid deception in all statements concerning software.

2. Client and Employer
   - Software engineers shall act in a manner that is in the best interests of their client and employer , consistent with the public interest

- The engineers should be honest about any limitation of their experience and education.
- There is need to keep private and confidential information consistent with the public interest and the law.
- They should not knowingly use software that is obtained or retained either illegally or unethically.

3. Product
   - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

- The developers should strive for high quality, acceptable cost and a reasonable schedule and ensure that significant trade-offs are clear.
- There should be adequate testing, debugging and reviewing of software along with related documents.
- All forms of software maintenance should be treated with same degree of professionalism as new product development.

4. Judgment
   - Software engineers shall maintain integrity and independence in their professional judgement.

- They developers should not engage themselves in deceptive financial practises.
- Conflict of interests that cannot be reasonably avoided or escaped should be disclosed to all concerned parties

5. Management
   - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

- Software engineers should be informed of the standards before they are held to them.
- There should be fair and just remuneration.
- Expression of ethical concerns about a project by someone must not be punishable.

6. **Profession**

   ➢ Software engineers shall advance the integrity and reputation of the profession consistent with public interest

- Everyone should strive to promote public knowledge of software engineering
- They should be accurate in stating the characteristics of software on which they work.
- They should take responsibility for detecting, correcting and reporting errors in software and associated documents on which they worked.

7. **Colleagues**

   ➢ Software engineers shall be fair to and supportive of their colleagues.

- Others work should be given full credit and one must refrain from taking undue credit
- There should be a fair hearing to the opinions, concerns, or complaints of a colleague.
- If a situation arise where it is outside of a software engineer's own area of competence, there is need to call upon the opinions of other professionals who have competence in that area.

8. **Self**

   ➢ Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

- Software Engineers should further their knowledge
- One should improve their ability to produce accurate, informative and well written documentation.
- One should improve their ability to create safe, reliable and useful quality software.

**The principle of Public Interest is central to the code.**

Problems with codes of conduct

1. Not all aspect are covered in every case nor should they.
2. There is questions on whether a list of rules can define a behaviour that is acceptable by everyone
3. There is little penalty for non-compliance to the code.

**Why should we be interested in ethics and professional conduct?**

- ✓ The quality of software produced by software developers is crucial to society ( Lives depend on safety and reliability if many software systems)
- ✓ Apart from the technical capability, the quality of software depend on the ethics and professional conduct of the software engineers that develop them.

### 8.4.5 Difference between a Code of Conduct and Law

A law applies in all and every situation while a code of conduct is relevant in a context such as an office, organisation etc.

## 8.6 Ethics and Ownership.– Ownership of software and data

### 8.6.1 Legislation

These are laws pertaining in one residing country. In the computer related industry, several regulations have been put in place. Some of the most common acts:

- ✓ Health and Safety Act
- ✓ Copyright , Design and Patents Act
- ✓ Computer Misuse Act
- ✓ Data Protection Act

The justification for copyright and ownership is to enable creators of intellectual wealth to financially support themselves and give them a motive to continue publishing their creations.

**Who is responsible for ensure proper protection?**

The WIPO (World Intellectual Property Organisation) is responsible for the protection of rights of owners and creators of intellectual property worldwide. It forms part of the United Nations specialised agency.

### 8.6.2 Data Protection Act – mean to protect individual's data and its use.

Data must be obtained only for one or more specified lawful purpose and shall not be further processed in any manner incompatible with that purpose or purposes.

Principles

This has Eight Principles:

1. Personal data should be obtained and processed fairly and lawfully
2. Personal data can be held only for specified and lawful purpose
3. Personal data should be adequate, relevant and not excessive for the required purpose
4. Personal data should be accurate and kept up to date
5. Personal data should not be processed in the accordance with the rights of the data subject
6. Personal data should not be kept longer than is necessary
7. Appropriate security measures must be taken against unauthorized access
8. Personal data cannot be transferred to countries outside/other countries unless the country has similar legislation as the DPA

Components of DPA

1. **Personal data:** facts and opinions about a living individual
2. **Data Subject:** is the person whom the data is needed for
3. **Data User:** is a person who need to access or use a data as part of their job
4. **Data Controller:** is the person in charge of the organization (not necessarily each time), he decides what data the organization needs to collect and what it would be used for
5. **Data Commissioner:** this is the person who enforces the data protection act. this is the person that organization need to apply in order to gain permission to collect and store personal data

### 8.6.3 Intellectual Property Rights

– is governed by different laws concerned in respective countries and are usually intangible property such as software. The law crosses national borders. There are 4 different main types of Intellectual Property Rights:

1. **Copyright** is concerned with the right to copy something
2. **Patents** are intended to protect inventions (applicable for a certain period of time)
3. **Confidential information** is concerned with information obtained in particular situations and that cannot be passed on
4. **Trade marks** identify the supplier or product of a particular manufacturer

### 8.6.4 Copyright

***Copyright gives the author of any text, including code, rights of protection from copying. Copyright prevents others from copying, issuing or amending software that it covers.***

Only certain types of work are protected by copyright law. The types that concern us here are 'original literary, dramatic, musical or artistic' works. In practice this applies to almost all software. Copyright comes into **existence automatically** and, in the UK, does not need to be registered.

The practical effect of this is that copying software without permission is forbidden. One-off copying can give rise to a civil action leading to the award of damages or a court order to desist. Commercial copying can be a criminal offence

In Mauritius, we have the following act in place among others:

- ✓ The Copyright Act, adopted in 1997
- ✓ The Patent, Industrial Designs and Trademarks Act 2002
- ✓ The Protection Against Unfair Practices (Industrial Property Rights) Act 2002
- ✓ The Layout Designs (Topographies) of Integrated Circuits Act, 2002
- ✓ The Geographical Indications Act, 2002

Rights of Copyright Owner

The certain exclusive rights are:

- ✓ The right to make copies of the work

- The right to issue copies of the work to the public, whether or not they are charged for it.
- The right to adapt the work: This includes translating it

Note: Copyright law does not give the owner of the copyright any power to prevent someone else using or publishing identical material, provided they can show that they did not produce it by copying the copyright work.

### What you can do to a copyright work

- It is not an infringement of copyright to make a backup of a program that you are authorized to use but however, only one such copy is allowed.
- You can 'decompile' a program in order to correct errors in it or for writing interoperable programs
- You can sell your right to use a program in much the same way that you can sell a book you own but however, when you do this, you sell all your rights.

### Copyright Infringement

Firstly, infringement takes place whenever any of the exclusive rights of the copyright owner is breached. It is a matter for the civil courts and the usual remedies are available: a claim for damages or an injunction to refrain from the infringement is the most likely. Secondly infringement occurs when primary infringement occurs in a business or commercial context. In the case of software, this could involve trading in pirated software or it could involve using pirated software within a business.

### 8.6.5 Trade Marks (IN UK done via UK Patent Office)

***A trade mark is any sign capable of being represented graphically which is capable of distinguishing goods or services of one undertaking from those of other undertakings. A trade mark may, in particular, consist of words (including personal names), designs, letters, numerals or the shape of goods or their packaging.***

If a symbol is registered as a trade mark, it is an offence for organisations other than the owner to display the trade mark on their products. By displaying the trade mark when its software is loaded, the owner of the trade mark prevents other organisations from falsely claiming that they have produced the software. In this way, the trade mark owner can protect its reputation.

Trademark protection is only effective in countries in which the trade mark is registered. Trademarks are an effective way of protecting retail package software from piracy e.g. Microsoft

### 8.6.6 Patents (IN UK done via UK Patent Office)

***A patent is a temporary right, granted by the state, enabling an inventor to prevent other people from exploiting his invention without his permission.***

The inventor must apply for the patent to be granted unlike copyright. The grant of a patent allows the person owning it (the patentee) to prevent anyone else from exploiting the invention, even if they have discovered it for themselves.

### What can be patented?

Based on the European Patent Convention the following invention can be patented only if:

- ✓ Is new;
- ✓ Involves an inventive step;
- ✓ Is capable of industrial application;
- ✓ Is not in an area specifically excluded.

### Obtaining a Patent

Applying for a patent can be an expensive and time consuming business. Patents are granted by national patent offices in their respective countries. In practice, there are schemes run by the European Patent Office and the World Intellectual Property Organisation. Since the computing is a global industry, any patents relating to computing need to be taken out in enough countries to make sure that the market in which the invention is not protected is too small to attract a competitor.

## 8.7 Software Licensing

There are two types of software:

- ✓ **Public Domain software** – these are freely available to anyone
- ✓ **Licensed Software** – these are copyrighted and / or patented which is made available for use under some limitation.

**Commercial Licensed Software** – **pay a fee with often binary form only no source code is provided**

**Shareware** – licensed with conditions depending on its usage. It usually has limited usage according to number of time or days.

**Open Source** – software source is provided with or without license fee

**Free software (Software Libre)** – source code is provided with licensing conditions to ensure freedom of use of the software for all users.

### 8.7.1 Commercial Licensed Software

In return for a licensing fee, the user is licensed to use the software subject to terms and conditions. The rights and obligations are clearly stipulated in the license itself. The main contents include:

- ✓ **Typical Rights** - specify if can install a single copy or multiple copy
- ✓ **Typical Restrictions** - may specify the limitations on software platform
- ✓ **Typical Obligations** - ensures that conditions are not breached.

### 8.7.2 Shareware Licensed Software

This form of licensing with individual developers or small companies wishing to attract the attention on their software through free trials to full purchase later.

- ✓ **Typical Rights** – free download and loading the product for personal use. Usual contains some restricted rights
- ✓ **Typical Restrictions** – contains restriction to some features, over how it can be used or even performance related restriction. These tend to be associated with trial editions.
- ✓ **Typical Obligations** - the software has a trigger that often requires the purchase of a commercial license at some point in time. It usually includes number of uses, time software was installed or the non-commercial use.

### 8.7.3 Open Source Software (OSS)

This type of software contains the lease restriction of all

- ✓ **Typical Rights** – User get access to source code and are allowed to do most things without much restriction.
- ✓ **Typical Restrictions** - There are very few rights
- ✓ **Typical Obligations** - These may be obligations such as maintaining copyright attributions, some artistic license such as modifications of software to be offered back to original author for use in the original package.

### 8.7.4 Free Software (Software Libre)

This licensing scheme was proposed by Richard Stallman. The original license (GNU – General Purpose License) ensures that software is available to anyone to use with quid pro quo that no one takes any rights that they don't pass on.

The free software does not necessarily means free of charge. Someone has the right to charge for passing on a free software package. Charges can be made for warranty or support of the product.

- ✓ **Typical Rights** – we have access to the form of the software used to write it such as the high level language source code. If binary form is available, an offer for desired form can be made. We have rights to modify and examine the software and redistribute it as well.
- ✓ **Typical Restrictions** - if the software modify and redistribute it, it must remain licensed under the original free license. Even derivative works must be published under a license compatible with the original one.
- ✓ **Typical Obligations** - all the facts mentioned in the restrictions must be observed.

## 8.8 Measures to restrict access to Data on Web / Internet

- ✓ Use proper licensing scheme (Creative Commons, GNU etc.)
- ✓ Use DRM mechanism ( Digital Rights Management)
- ✓ Use Encryption mechanism to distribute sensitive data

# 9.0 Database and Modelling: Databases

## 9.1 Flat File Database

All data are held in one single file consisting of multiple records. All items are stored sequentially in one table. The same piece of information may be repeated several times. The user interface is dependent on the file structure.

## 9.2 Relational Databases

It comprises of a number of tables linked together via a relations often called foreign keys. Each table stores data about an entity. Each record is known as a tuple and each field is known as an attribute.

| | Advantages | Disadvantages |
|---|---|---|
| **Flat File database** | 1. Easier to human to read<br>2. Fast data retrieval | 1. Slower data entry<br>2. Duplicated data<br>3. Data inconsistency is possible<br>4. Data Dependence<br>5. Little or no access control<br>6. Queries and reports are slow to produce |
| **Relational database** | 1. Less data inconsistency<br>2. Less data redundancy<br>3. Better security<br>4. Program data independence | 1. Slower data retrieval<br>2. Difficult to read original table data |

## 9.3 Comparison between Flat File and Relational database

| | Flat Files | Relational Database |
|---|---|---|
| **Data Storage** | A number of files are used to store data / information | A single software application is used to store the data contained. |
| | The data are duplicated and may become inconsistent ( not all parts of related data have been updated) | Duplication of data is reduced along with the chance of data inconsistency minimized.<br>Good database duplication avoids duplication and data can be accessed via links in the table |
| | Volume of data stored is large due to duplication | Volume of data is considerably reduced since there is less data duplication which in turn lead to faster sorting and searching of data |

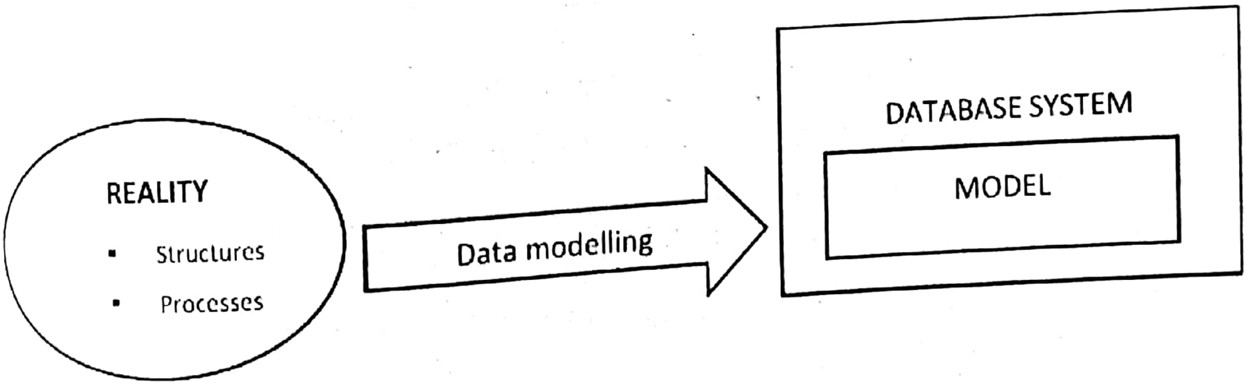
| | | |
|---|---|---|
| Program data independence | If the data structure is modified , the software being used must be coded again | The data structures remains the same even if the tables are altered. Thus, there is no need to alter existing programs due to table design changes. |
| | Views of data are dictated by the distinct files used to control the data and produced by individual departments. Due to this, it is very time consuming. | Queries and reports are easily generated using the wizards or data manipulation language even for novices. |

## 9.4 DBMS (Database Management Systems)

It is a piece of software that allows the manipulation of databases so that users can access the database, maintain and view data. The data are organised into tables, viewed in forms, retrieved using queries and displayed in reports. It usually offers the following facilities:

1. Database design and query design
2. Data Definition Language (**DDL**) to define tables
3. Data Dictionary (**DD**) that describes the tables, relationships and indexes along with the rules for data integrity and validations.
4. Data Manipulation Language (**DML**) which allows for data manipulation using structured query language
5. Backup features
6. Controlling multi user access to data
7. Security and access right features to individual or groups of users

### 9.4.1 Data Modelling



A model represents a perception of structure of reality. The data modelling process fixes a perception of structure of reality and represent the latter. In data modelling process, we select the aspects and we abstract. A data model consists of notation for expressing data structures, integrity constraints and operations.

### 9.4.2 Schema

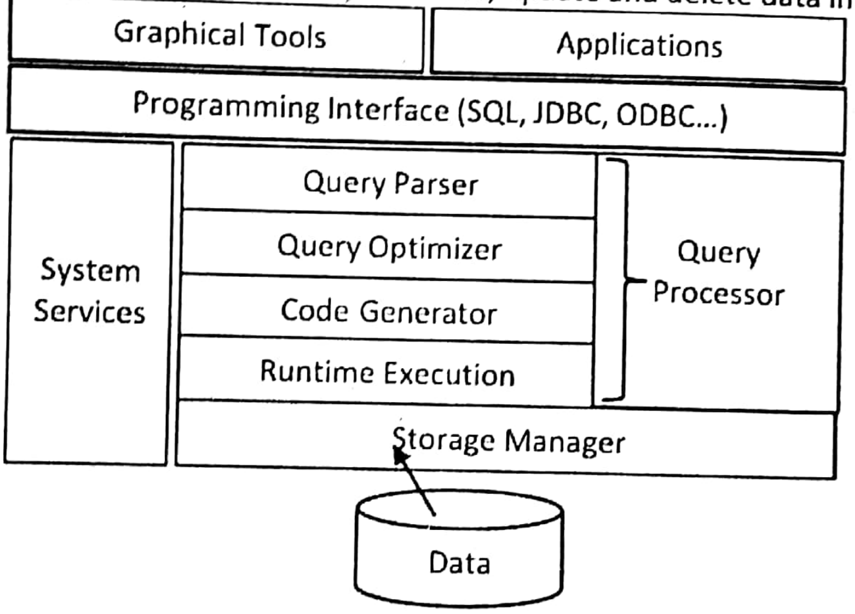
It defines its entities and the relationship among them. Database schema is a descriptive detail of the database, which can be depicted by means of schema diagrams. All these activities are done by database designer to help programmers in order to give some ease of understanding all aspect of database.

### 9.4.3 Logical Database Schema

This defines all logical constraints that need to be applied on data stored. It defines tables, views and integrity constraints etc.

### 9.4.4 Query Processor

A query processor sits between the DBMS interface layer and executes requests to store and retrieve data efficiently and robustly as possible. The query processor handles not only requests to retrieve data, but also handles requests to insert, update and delete data in the database.



### 9.4.5 High Level Language support for databases

- VB.net programmers can use Microsoft ADO.NET framework to connect to access database by using the OLE DB data provider. The OleDbDataAdapter class to obtain rows from the Access database and to insert the rows into a DataSet object.
- PHP can access virtually any database available today. One may use ODBC driver on windows platform.
- Python can use the python DB-API to access a range of databases such as MySQL, Oracle, PostgreSQL, Informix etc.
- Delphi provides the Borland Database Engine to access databases.

## 9.5 Database

A database is a collection of data or information which is held together in an organised or logical way.

**Manual databases** - > school register, telephone directory
**Computerised databases** -> bank databases, search engines, customers database for a business

## Database Management Systems – DBMS

Databases are used to collect and organise information. DBMS are the software packages that allow the creation of databases. Ex – MYSQL, MICROSOFT SQL SERVER, ORACLE, POSTGRESQL, MICROSOFT ACCESS.

### Comparison of Computerised against Manual database

| **COMPUTERISED** | **PAPER BASED** |
|---|---|
| Can hold a vast amount of data | Limited by physical storage space availaible |
| Very fast to find a specific record | Can take a while to manually search through all of the records |
| Can easily search for a specific criteria | Difficult to search for specific criteria |
| Can be used to analyse the data – e.g popular item sales | Very difficult to analyse the data |
| Data can be sorted into ascending or descending order on multiple criteria | Difficult to sort data on more than one criteria |
| Can easily update or amend a record | Changes have to be done manually. Records can look messy if scribbled out |

## 9.6 Database Terminologies

### Record

A record is all of the data or information about one person or one thing. In computerised databases, in each table, each row represents a record.

### Field

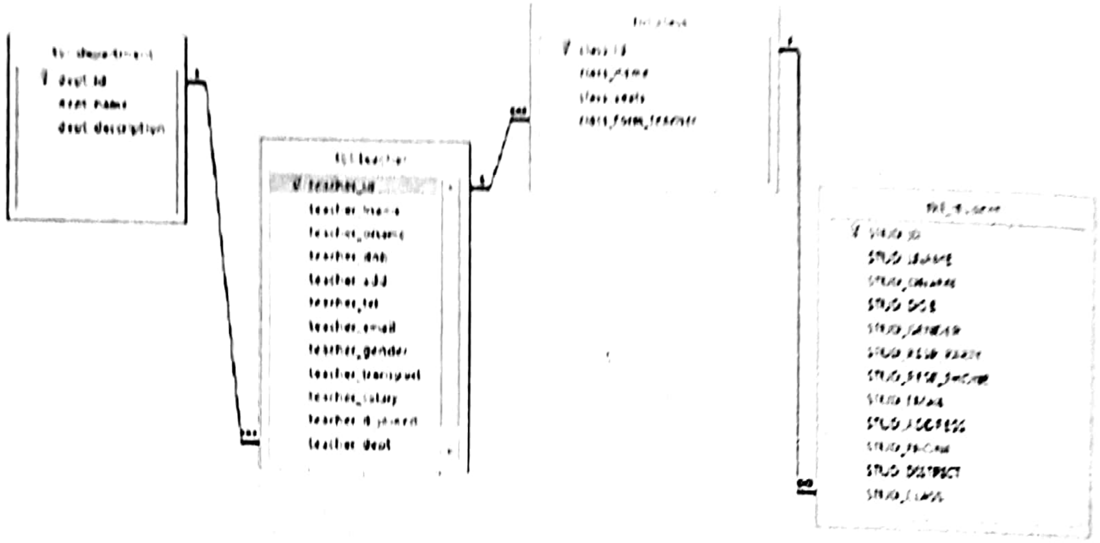
A field is one piece of data or information about a person or thing. It usually has a unique name and has data type associated with it along with a length. In tables, each column in the table represents a field.

### Primary Key or Key field

It is a field that uniquely identifies a record in a database. E.g. student id, national ID

### Foreign Key

A foreign key is the primary key of another table and is used to link tables together to create relationship. In the diagram below , in table tbl_class, the foreign key is class_form_teacher.



### Field Length

It is the number of characters that the field will actually store. It is usually applied to text field type. E.g. for Name, the field length can be 50

### Reports

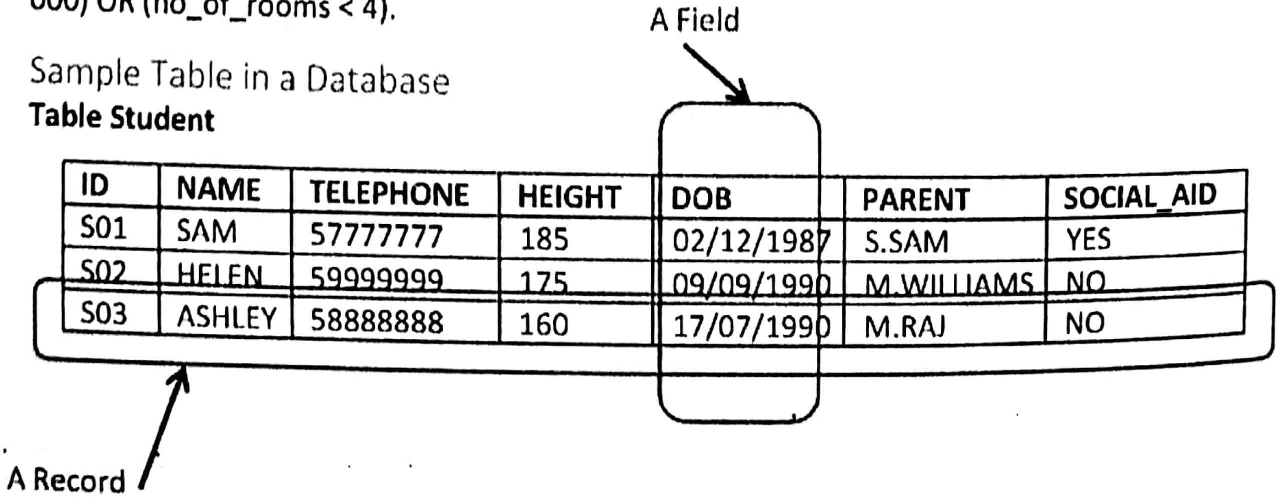
Reports are produced as a result of questions such as "how many times has a type of car broken down" or "which cities have the highest house prices". Reports can be printed out, viewed on the screen or exported (e.g. to a word processor, an email message, etc.).

### Queries

Queries most commonly allow information to be retrieved from tables. Since the information is often spread across several tables, queries allow it to be viewed as one single datasheet. They also allow filtering so only the records required are seen. Queries can be either results seen directly on the screen or the output to another form or report. Example of a query: (house > 200 000) OR (no_of_rooms < 4).

### Sample Table in a Database

**Table Student**


A Field
A Record

| ID | NAME | TELEPHONE | HEIGHT | DOB | PARENT | SOCIAL_AID |
|---|---|---|---|---|---|---|
| S01 | SAM | 57777777 | 185 | 02/12/1987 | S.SAM | YES |
| S02 | HELEN | 59999999 | 175 | 09/09/1990 | M.WILLIAMS | NO |
| S03 | ASHLEY | 58888888 | 160 | 17/07/1990 | M.RAJ | NO |

## In the Above Example

- ✓ Number of Records ( 3) – 3 Rows( S01,S02,S03)
- ✓ Number of Fields (7) – 7 Fields ( ID,NAME,TELEPHONE,HEIGHT,DOB,PARENT,SOCIAL_AID)
- ✓ Primary Key - ID
- ✓ Query -> (HEIGHT >170) AND (SOCIAL_AID="YES")

## 9.7 Data Types

Note: In Different DBMS, the data types may be known differently.

| Data Types | Examples |
|---|---|
| ✓ **Text or Alphanumeric or String or Varchar**<br>✓ Allows you to type in text, numbers and symbols | Name : James<br>Telephone : 02059586320<br>Note: For telephone we use text as a number can be preceded by 0 and it exists as a value |
| ✓ **Date / Time**<br>✓ This restricts data entry to 1-31 for day ( 28 ,30) in appropriate months and 1-12 for month<br>✓ Checks whether a date can actually exist<br>✓ Formats the data into medium, long, short date/time<br>✓ Note : in US ,MM/DD/YYYY and in UK , DD/MM/YYYY<br>✓ For example 12/01/2014 in US Format Means 1st December 2014 while in UK format it means 12 January 2014. | Short Date : 20-Jan-14<br>Long Date : 20 January 2014<br>Medium Date : 20/01/2014<br>Long Time : 23:20:23<br>Medium Time : 05:23 PM<br>Short Time : 17:23 |
| ✓ **Autonumber or Serial**<br>✓ This datatype will automatically increase by 1 as a record is added to the database /table<br>✓ It may be used as an index or even primary key in some cases | Record 1: 1<br>Record 2: 2 |
| ✓ **Currency**<br>✓ This automatically formats the data to have a $ or £ in front of the data and also ensures that there are two decimal places | $9.23<br>£ 69.86 |

| | |
|---|---|
| ✓ **Boolean or Logical or Yes /NO**<br>✓ Often restrict to two choice only that is to either true or false | ON /OFF<br>TRUE /FALSE<br>Yes /No |

## 9.8 Database and Modelling: Relational Database Terminologies

### Entity

- An entity is defined as an object, person, event or 'thing' about which data can be recorded. It has a set of attributes (property / characteristic).
- An association or link connects two entities.
- E.g. Customer is an entity and may have attributes such as name, address, gender etc.

### Tuple

- The record in a table is known as a tuple in RDBMS

### Attribute

- The field in a table is known as attribute in RDBMS

### Referential integrity

Referential integrity

- A referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples in the two relations.
- Example : if a foreign key in a table is defined as a particular type say number, in the table where the latter is being used as primary key, they should have same data type and must exist in that table(relation).

### Candidate Key

A candidate key is another key in a relation schema or table which is unique and could have been used as a primary key. Often database designer choose the most appropriate field to be the key field from the candidate keys available.

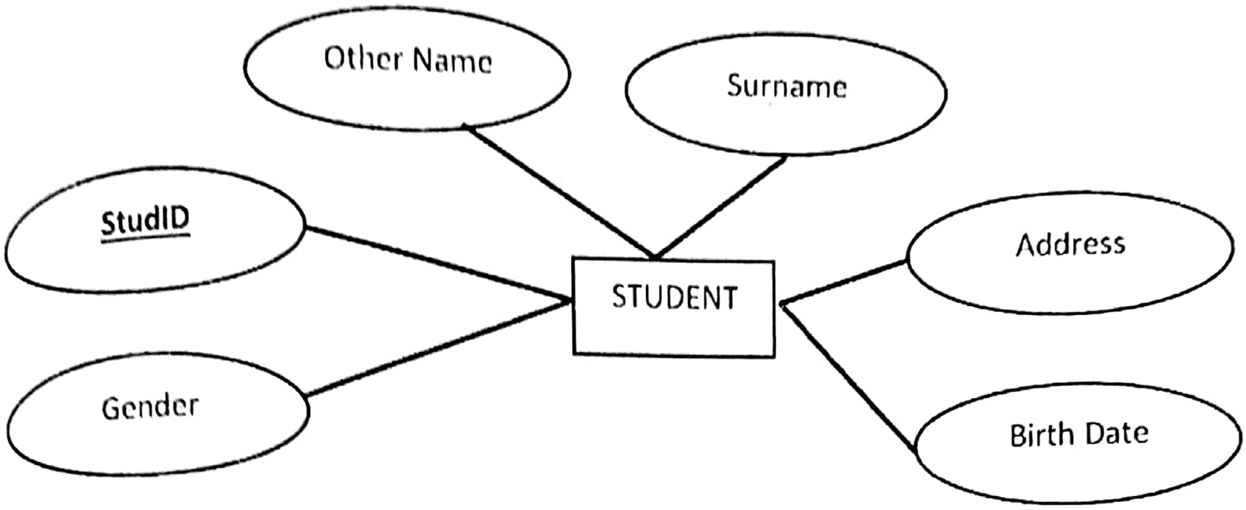
**Example:** in a particular table, a student might have a student code as primary key but we could have use his/her National ID card which represents another candidate key.

### Secondary Key

A secondary key is defined as an attribute which is used to access the data in some way other than by using the defined primary key. Normally secondary key are used as indexes to facilitate and enhance searching/sorting of records.

## 9.9 Entity Relationship Diagram (E-R diagram)

**E-R diagrams** illustrates the relationships between entities in the database and attributes of each via diagrams. The entity is generally represented in boxes while the attributes in oval shape.



### Example of a Data Model

Note: the relationship depends on your interpretation of the context. You should consider the rule that pertain to that context. However, because of distinct interpretation, Cambridge already gives the different relationship in the question itself so that student don't make confusion.

To decide which type of relationship, we must consider the relationship of the entity at a time [at a particular moment].

Consider a School wishing to computerise its actual manual system to store data about its student.

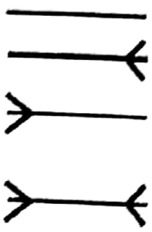
- A teacher can teach only one subject at a time but a subject can be taught by many teachers at a time. [Many – One]
- A teacher attend a class at a time and a class can is attended by a teacher at a time. [One to one]
- An HOD is responsible for a department and a department can only have one HOD at a time. [ One to One ]
- A class consists of many students and a student is attends one class at a time. [One to Many]
- A student can take many subjects and a subject can be taken by many students [many to many]
- A teacher can be in only one department but a department consist of many teachers. [ one to many]

### Entities and their relationships in this example data model above

Note: Entities have to be written in singular and not in plural.

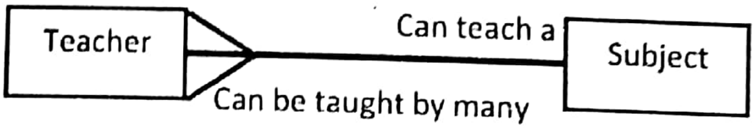
## 9.10 Entity Relationship types and Symbols

1. One to One which is represented by
2. One to Many which is represented by
3. Many to One which is represented by
4. Many to Many which is represented by

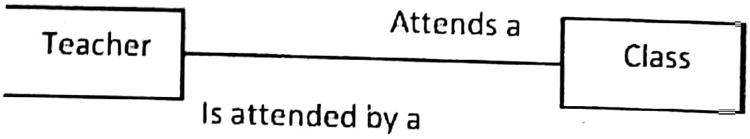


**ER Diagram for Model above**

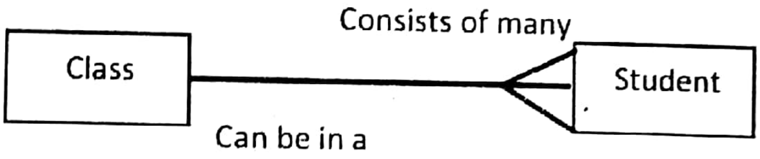
A teacher can teach only one subject at a time but a subject can be taught by many teachers at a time. [Many – One]



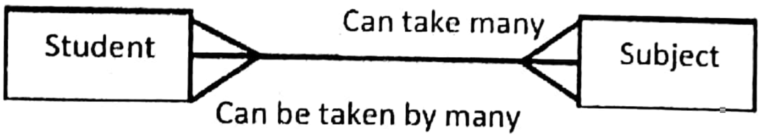
A teacher attend a class at a time and a class can is attended by a teacher at a time. [One to one]



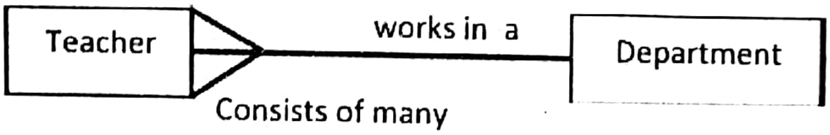
A class consists of many students and a student is attends one class at a time. [One to Many]



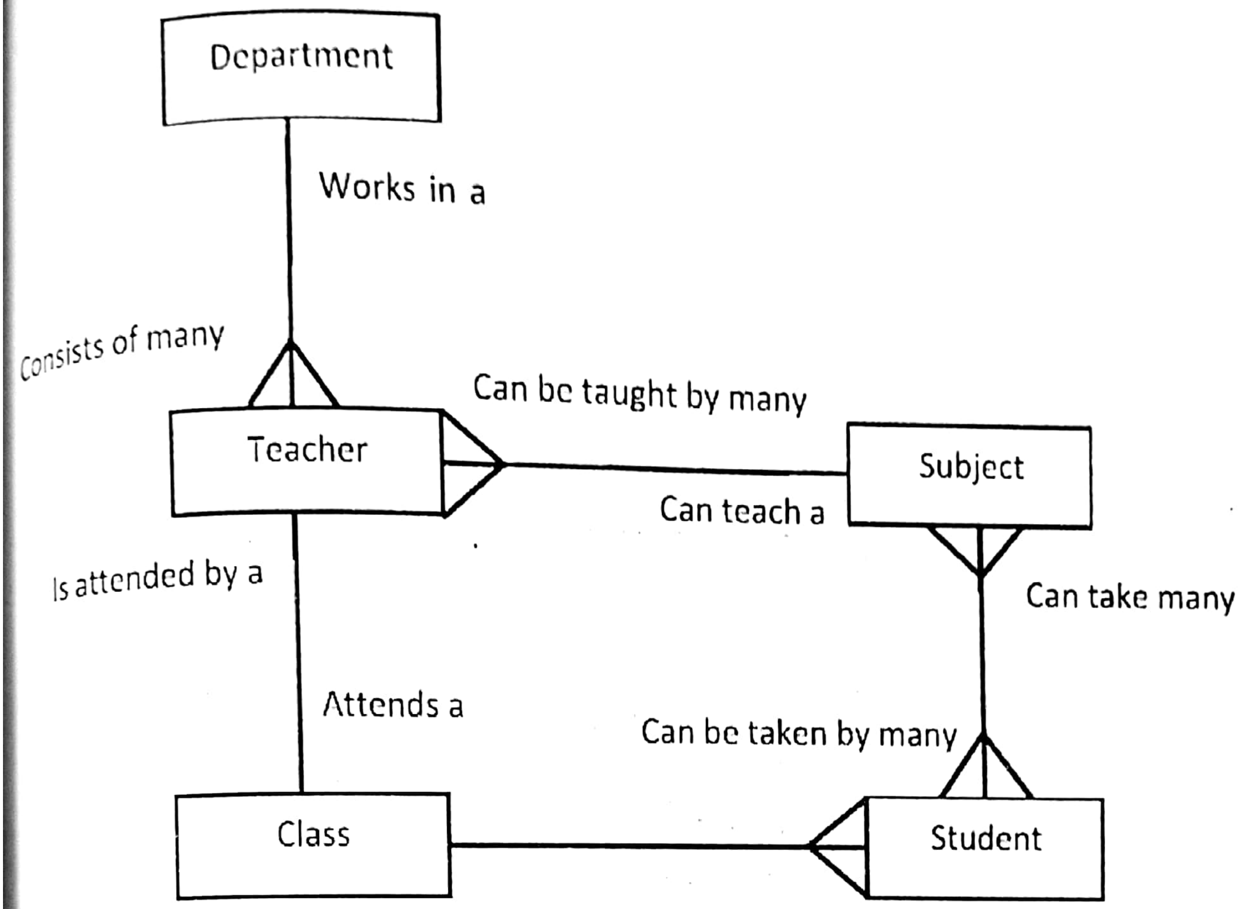
A student can take many subjects and a subject can be taken by many students [many to many]



A teacher can be in only one department but a department consist of many teachers. [ one to many]

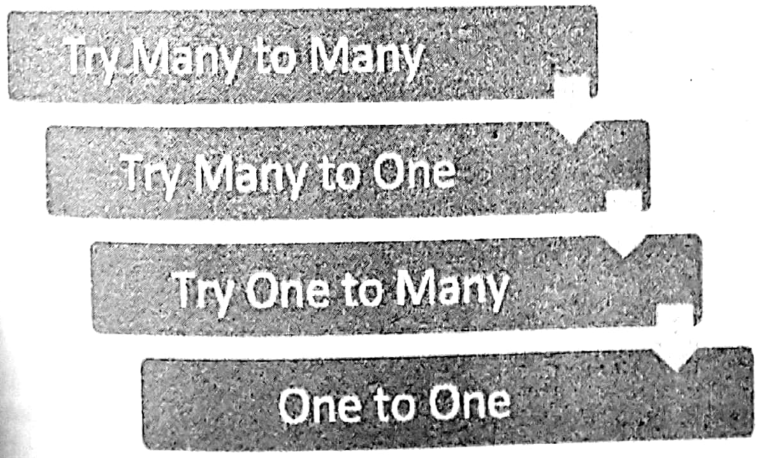


Putting it all together we obtain:



## Steps in forming relationships

To avoid confusion, the following strategy may be adopted when deciding which type of relationship and always study the context and remember at a particular instance in time. Different interpretation may lead to different design.!!!



## 9.11 Database Normalisation

***It is defined as a technique that is used for designing relational database tables so as to minimise duplication of data which can eventually lead to inconsistencies and the latter leading to the loss of data integrity.***

Normalisation consists of analysing relations to meet increasingly more stringent normal forms leading to progressively better groupings of attributes. Normal forms are specified in terms of function dependencies identified by the database designer.

### UNF (Un-normalised Normal Form)

1. 1st Normal Form – a relation is said to be in 1st normal form if it does not have a repeated group of attributes
2. 2nd Normal Form – a relation is said to be in 2nd normal form if it is in 1st normal form and there are no non key field's attributes which are dependent on only part of the primary key.
3. 3rd Normal Form – a relation is said to be in 3rd normal form if it is in 2nd normal form and there are no dependencies between non-key attributes.

### Normalisation Process

Consider the following receipt

**Nand Chicken Eater Ltd**

**Royal Road Venise**

**To : Navin Cassanova**
**Venise**
**Italy**

**Inv No : 769**
**Date : 10/12/2014**

| **Item No** | **Description** | **Quantity** |
|---|---|---|
| 1 | Crispy Strips | 3 |
| 2 | Box Master | 2 |
| 3 | Zinger Regular | 2 |
| 4 | Virgin Mojito | 2 |

Un-normalised Form - **RECEIPT** (Num, CusName, City, country, (ItemID, Description))

RECEIPT is the name of the relation while Num, CusName, City, Country, ProdID and Description are attributes.

Note: ItemID, Description are inside () because they form a repeating group. Hence, customers can order more items or fewer items)

Table in UNF

| Num | CusName | City | Country | ItemID | Description |
|---|---|---|---|---|---|
| 769 | Navin Cassanova | Venice | Italy | 1 | Crispy Strips |
| 769 | Navin Cassanova | Venice | Italy | 2 | Box Master |
| 769 | Navin Cassanova | Venice | Italy | 3 | Zinger Regular |
| 769 | Navin Cassanova | Venice | Italy | 4 | Virgin Mojito |
| 666 | M.Ahmed | Cairo | Egypt | 1 | Crispy Strips |
| 666 | M.Ahmed | Cairo | Egypt | 3 | Zinger Regular |
| 022 | John Smith | London | England | 4 | Virgin Mojito |

## 1st Normal Form

- The table is completed and the primary key is identify to make each tuple unique
- Underline attributes shows primary key
- Num + ItemID = Primary Key (Composite Key)
- **RECEIPT** (Num, CusName, City, Country, ItemID, Description)

| Num | CusName | City | Country | ItemID | Description |
|---|---|---|---|---|---|
| 769 | Navin Cassanova | Venice | Italy | 1 | Crispy Strips |
| 769 | Navin Cassanova | Venice | Italy | 2 | Box Master |
| 769 | Navin Cassanova | Venice | Italy | 3 | Zinger Regular |
| 769 | Navin Cassanova | Venice | Italy | 4 | Virgin Mojito |
| 666 | M.Ahmed | Cairo | Egypt | 1 | Crispy Strips |
| 666 | M.Ahmed | Cairo | Egypt | 3 | Zinger Regular |
| 022 | John Smith | London | England | 4 | Virgin Mojito |

Note: The details of every customer for each item on the receipt are duplicated.

## 2nd Norma Form

- All the incomplete dependencies have been removed.
- CusName ,City and Country depend only on Num and not on ItemID
- Num determines CusName, City ,Country
- Description only depends on ItemID, it does not depend on Num
- ItemID determines Description
- Relationship showing which items have been bought by which customer is lost, thus to keep this connection :
- Num ,ItemID ( Dummy functional dependency ) relationship is created
- Hence , relations in 2nd NF are :

**RECEIPT** (Num, CusName, City, Country)

**ITEM** (ItemID, Description)

**RECEIPT_ITEM (Num, ItemID)**

Note: RECEIPT_ITEM needs a compound key because a receipt may contain several items and similar items may be found on several receipts.

### 3rd Normal Form

- All non –key elements are fully dependent on the primary key.
- Country depends on City not directly on Num
- Customer Name depends on customer number
- Hence , relations in 3rd NF

**RECEIPT** (**Num**, CusName, City)

**CITY_COUNTRY** (**City**, Country)

**ITEM** (**ItemID**, Description)

**RECEIPT_ITEM (Num, ItemID)**

# 9.12 Database and Modelling: Data Definition Language (DDL) and Data Manipulation Language (DML)

## SQL

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in relational database. SQL is the standard language for Relation Database System. All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix, PostgreSQL and SQL Server use SQL as standard database language.

## Why SQL?

- ✓ Allows users to access data in relational database management systems.
- ✓ Allows users to describe the data.
- ✓ Allows users to define the data in database and manipulate that data.
- ✓ Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- ✓ Allows users to create and drop databases and tables.
- ✓ Allows users to create view, stored procedure, functions in a database.
- ✓ Allows users to set permissions on tables, procedures and views

## SQL Commands

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into groups based on their nature:

### DDL – Data Definition Language

| Command | Description |
|---|---|
| CREATE | Creates a New Table ,a view of a table, or other object in database |
| ALTER | Modifies an existing database object such as a table |
| DROP | Deletes an entire table ,a view of a table or other object in the database |

### DML – Data Manipulation Language

| | |
|---|---|
| INSERT | Creates a record |
| UPDATE | Modifies records |
| DELETE | Delete records |

### DCL – Data Control Language

| | |
|---|---|
| GRANT | Gives a privilege to user |
| REVOKE | Takes back privilege granted from user |

### DQL – Data Query Language

| | |
|---|---|
| SELECT | Retrieves certain records from one or more tables |

## 9.13 SQL Demonstration using WAMP server (Windows Apache MySQL PHP)

The following software is a freeware that can be obtained from:

http://www.wampserver.com/en/

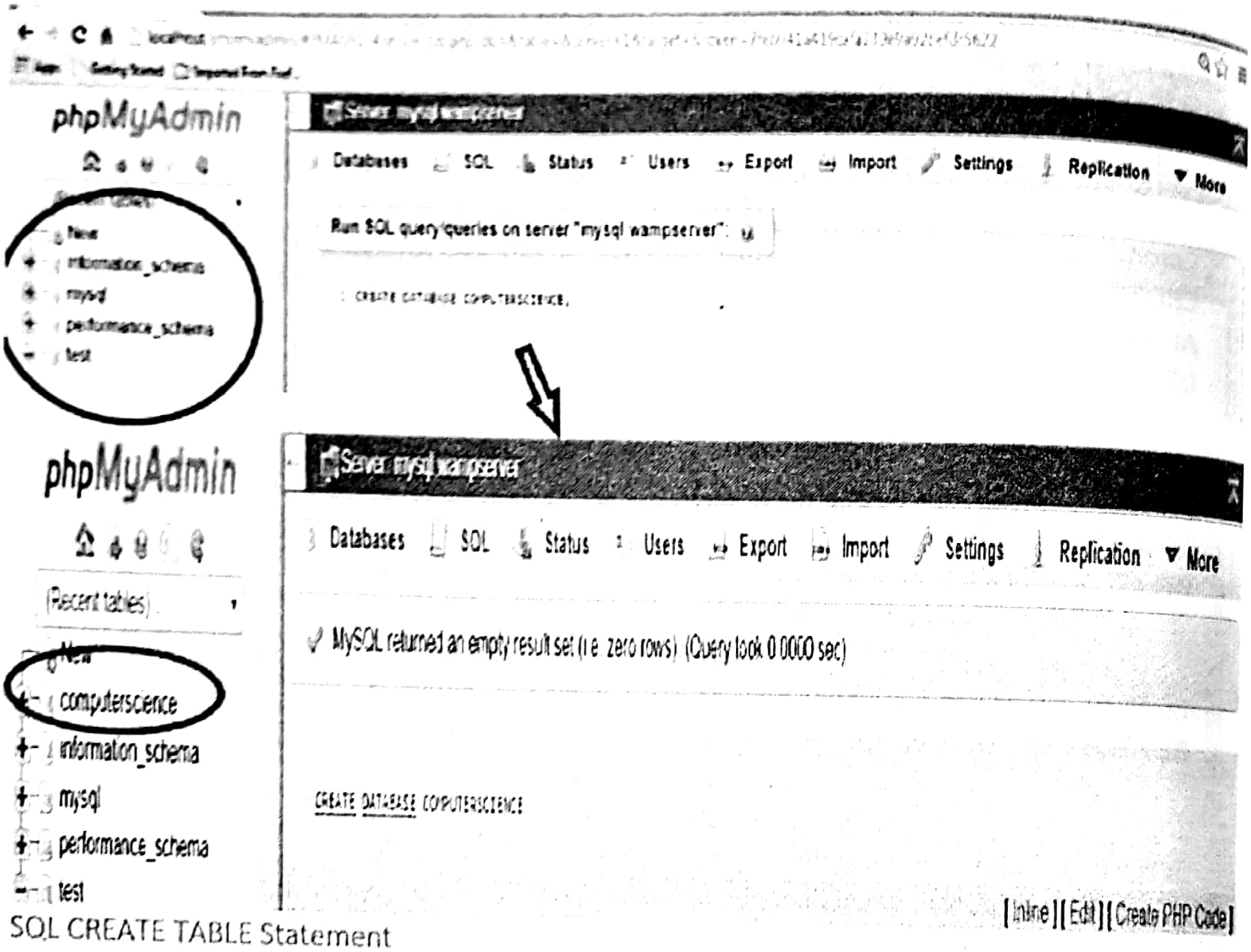
- All SQL statements start with any keywords such as **SELECT, INSERT, UPDATE, DELETE,** ALTER, **DROP, CREATE, USE, SHOW** and finishes with a semicolon (;)
- Note SQL is case insensitive meaning even if you write in capital or small letters it means the same except for table name
- The following is a sample database that we are going to create and manipulate using SQL statements

### SQL CREATE Database statement

Format : **CREATE DATABASE** database_name;

Example : **CREATE DATABASE** ComputerScience;

Explanation : The above command creating a database called ComputerScience as shown below.



Run SQL query/queries on server "mysql wampserver":

```
CREATE DATABASE COMPUTERSCIENCE;
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0000 sec)

```
CREATE DATABASE COMPUTERSCIENCE
```

SQL CREATE TABLE Statement

**Format** : CREATE TABLE table_name(

Column1 datatype1 ,

Column 2 datatype2,

.......

Column N datatype

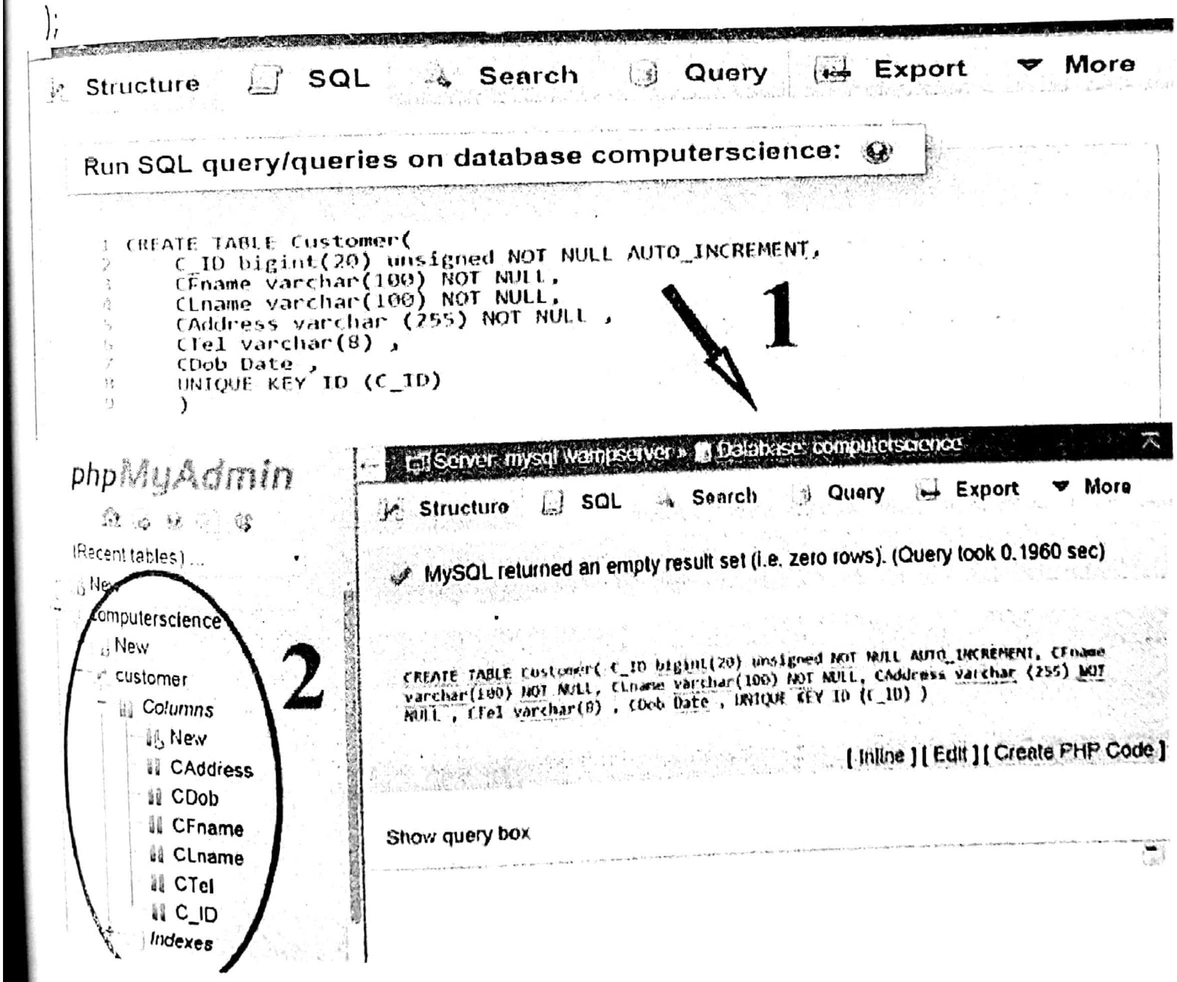
)

**Explanation** : The create table command is simply creating a table in the database we created with the respective field names and data type specified.

Example: let us create the following 3 tables that we would use as example throughout:

## Create Table Customer

```
CREATE TABLE Customer(
C_ID bigint(20) unsigned NOT NULL AUTO_INCREMENT,
CFname varchar(100) NOT NULL,
CLname varchar(100) NOT NULL,
CAddress varchar (255) NOT NULL ,
CTel varchar(8) ,
CDob Date ,
UNIQUE KEY ID (C_ID)
);
```



Run SQL query/queries on database computerscience:

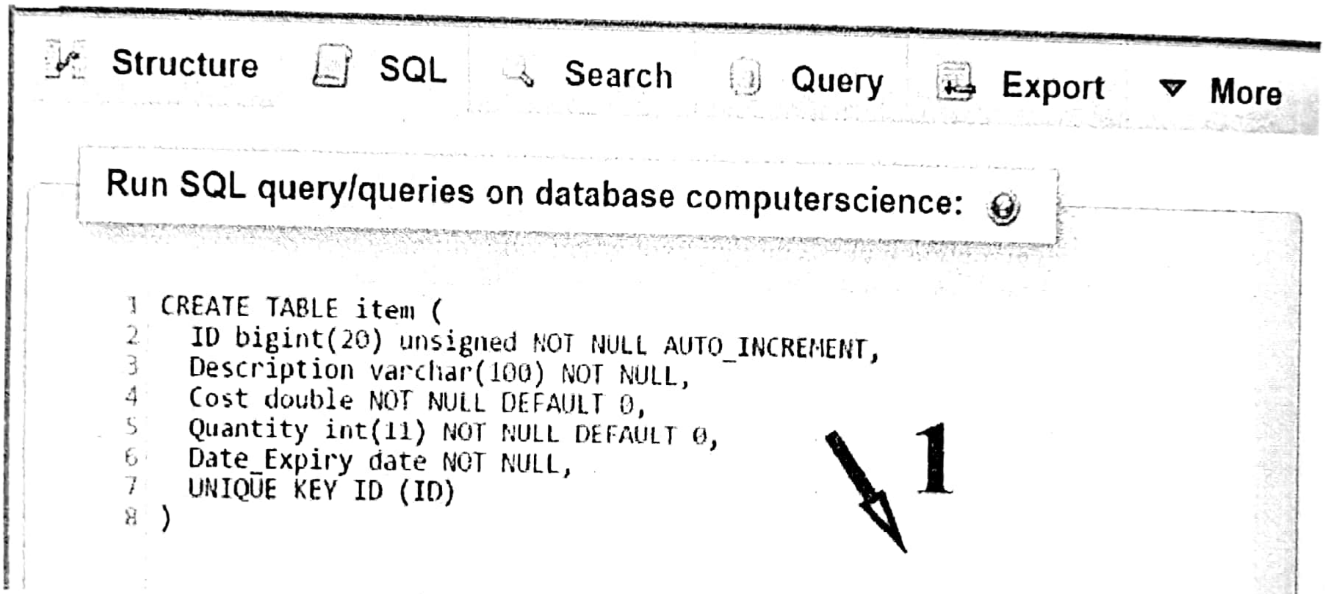
```
CREATE TABLE Customer(
    C_ID bigint(20) unsigned NOT NULL AUTO_INCREMENT,
    CFname varchar(100) NOT NULL,
    CLname varchar(100) NOT NULL,
    CAddress varchar (255) NOT NULL ,
    CTel varchar(8) ,
    CDob Date ,
    UNIQUE KEY ID (C_ID)
    )
```

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1960 sec)

```
CREATE TABLE Customer( C_ID bigint(20) unsigned NOT NULL AUTO_INCREMENT, CFname
varchar(100) NOT NULL, CLname varchar(100) NOT NULL, CAddress varchar (255) NOT
NULL , CTel varchar(8) , CDob Date , UNIQUE KEY ID (C_ID) )
```

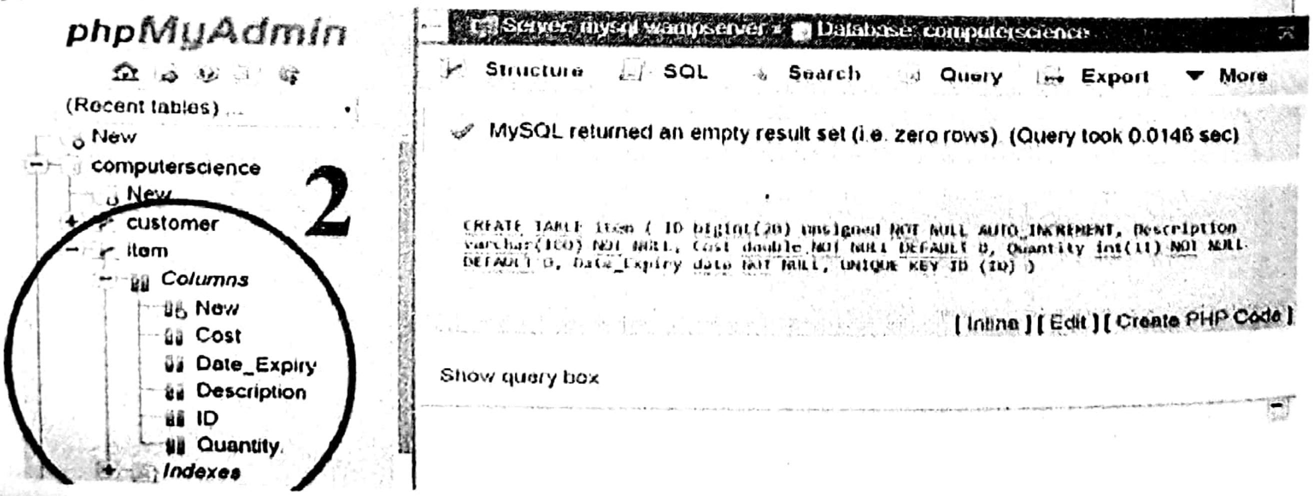
## Create Table Item

```
CREATE TABLE item (
 ID bigint(20) unsigned NOT NULL AUTO_INCREMENT,
 Description varchar(100) NOT NULL,
 Cost double NOT NULL DEFAULT 0,
 Quantity int(11) NOT NULL DEFAULT 0,
 Date_Expiry date NOT NULL,
 UNIQUE KEY ID (ID)
);
```



Run SQL query/queries on database computerscience:

```
CREATE TABLE item (
  ID bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  Description varchar(100) NOT NULL,
  Cost double NOT NULL DEFAULT 0,
  Quantity int(11) NOT NULL DEFAULT 0,
  Date_Expiry date NOT NULL,
  UNIQUE KEY ID (ID)
)
```

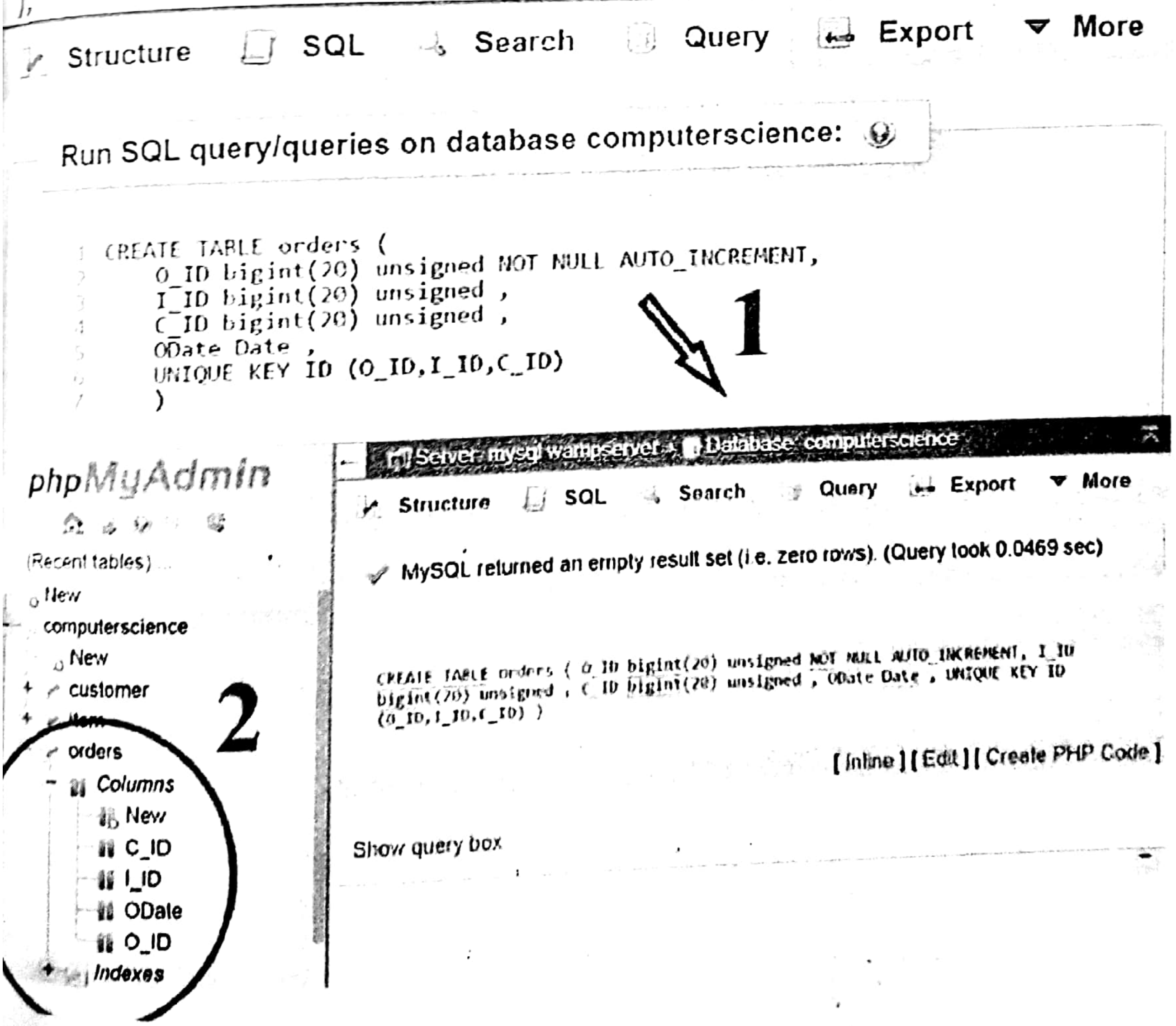


MySQL returned an empty result set (i.e. zero rows). (Query took 0.0146 sec)

```
CREATE TABLE item ( ID bigint(20) unsigned NOT NULL AUTO_INCREMENT, Description
varchar(100) NOT NULL, Cost double NOT NULL DEFAULT 0, Quantity int(11) NOT NULL
DEFAULT 0, Date_Expiry date NOT NULL, UNIQUE KEY ID (ID) )
```

## Create Table Orders

```
CREATE TABLE orders (
O_ID bigint(20) unsigned NOT NULL AUTO_INCREMENT,
I_ID bigint(20) unsigned ,
C_ID bigint(20) unsigned ,
ODate Date ,
UNIQUE KEY ID (O_ID,I_ID,C_ID)
);
```



Run SQL query/queries on database computerscience:

```
CREATE TABLE orders (
    O_ID bigint(20) unsigned NOT NULL AUTO_INCREMENT,
    I_ID bigint(20) unsigned ,
    C_ID bigint(20) unsigned ,
    ODate Date ,
    UNIQUE KEY ID (O_ID,I_ID,C_ID)
    )
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0469 sec)

```
CREATE TABLE orders ( O_ID bigint(20) unsigned NOT NULL AUTO_INCREMENT, I_ID
bigint(20) unsigned , C_ID bigint(20) unsigned , ODate Date , UNIQUE KEY ID
(O_ID,I_ID,C_ID) )
```

## SQL ALTER TABLE

The ALTER command is used to modify the structure of a table usually.

**Format** : **ALTER TABLE table_name ADD column_name data_type;**

- **ALTER TABLE** item **ADD** Origin varchar(100);

**Explanation:** The command above simply modify the table item and adds a new column called Origin with data type varchar and maximum length 100 characters.

- **ALTER TABLE** orders **ADD** Qty int(11)

**Format** : **ALTER TABLE table_name DROP column_name data_type;**

- **ALTER TABLE** item **DROP** Quantity;

**Explanation:** the above command is being used to delete the column called **Quantity** from the table **item**.

**Format** : **ALTER TABLE** item **CHANGE** Quantity Qty int (11);

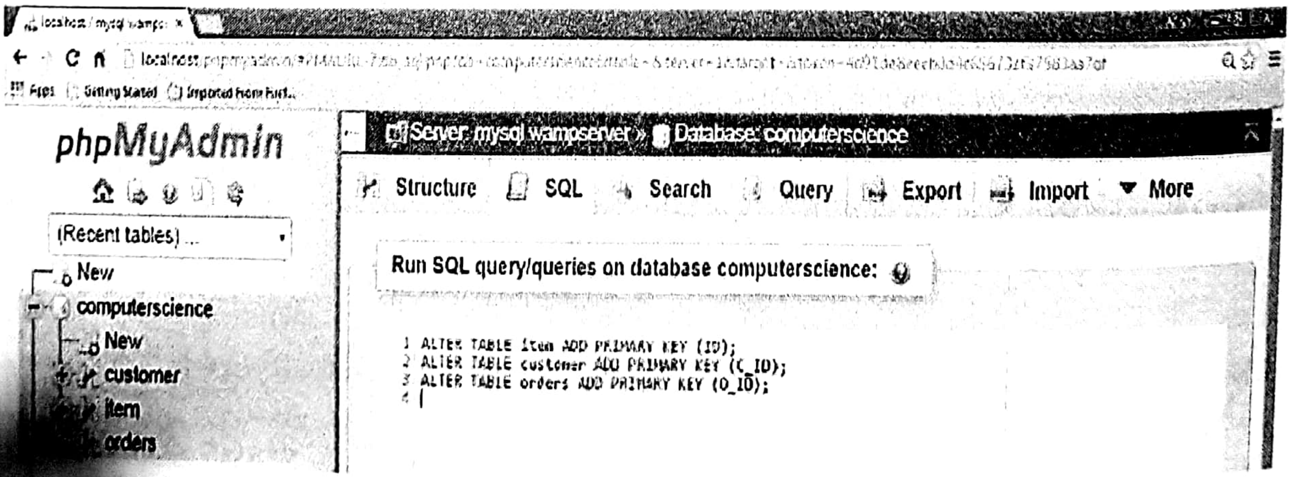
**Explanation:** The command above is simply renaming the table **item** column **Quantity** to a new column name entitled Qty with data type integer.

## SQL TO ADD Primary Key or Foreign Key to a table

**Primary Key**

- **ALTER TABLE** item **ADD PRIMARY KEY** (ID);
- **ALTER TABLE** customer **ADD PRIMARY KEY** (C_ID);
- **ALTER TABLE** orders **ADD PRIMARY KEY** (O_ID);

**Explanation:** the above command are simply adding a constraint that is a primary key to specified table.



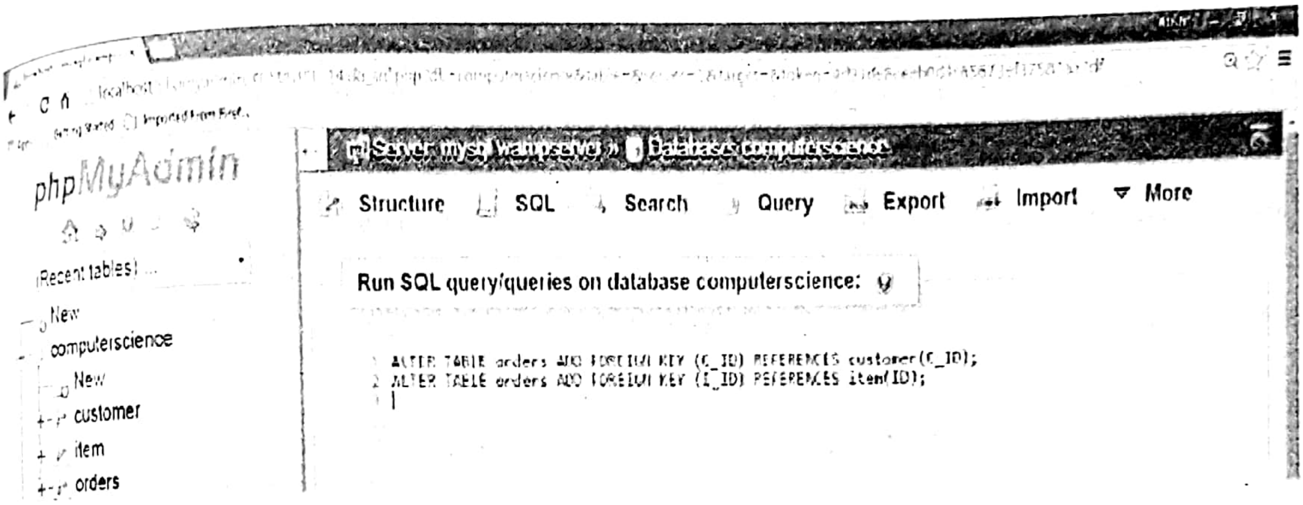
Run SQL query/queries on database computerscience:

```
ALTER TABLE item ADD PRIMARY KEY (ID);
ALTER TABLE customer ADD PRIMARY KEY (C_ID);
ALTER TABLE orders ADD PRIMARY KEY (O_ID);
```

### Foreign Key

- **ALTER TABLE** orders **ADD FOREIGN KEY** (C_ID) **REFERENCES** customer(C_ID);
- **ALTER TABLE** orders **ADD FOREIGN KEY** (I_ID) **REFERENCES** item(ID);

Explanation: The above two commands are adding the constraint of foreign key to the respective table and linking them with the table they should be connected.



Run SQL query/queries on database computerscience:

```
ALTER TABLE orders ADD FOREIGN KEY (C_ID) REFERENCES customer(C_ID);
ALTER TABLE orders ADD FOREIGN KEY (I_ID) REFERENCES item(ID);
```

### SQL DATA Maintenance

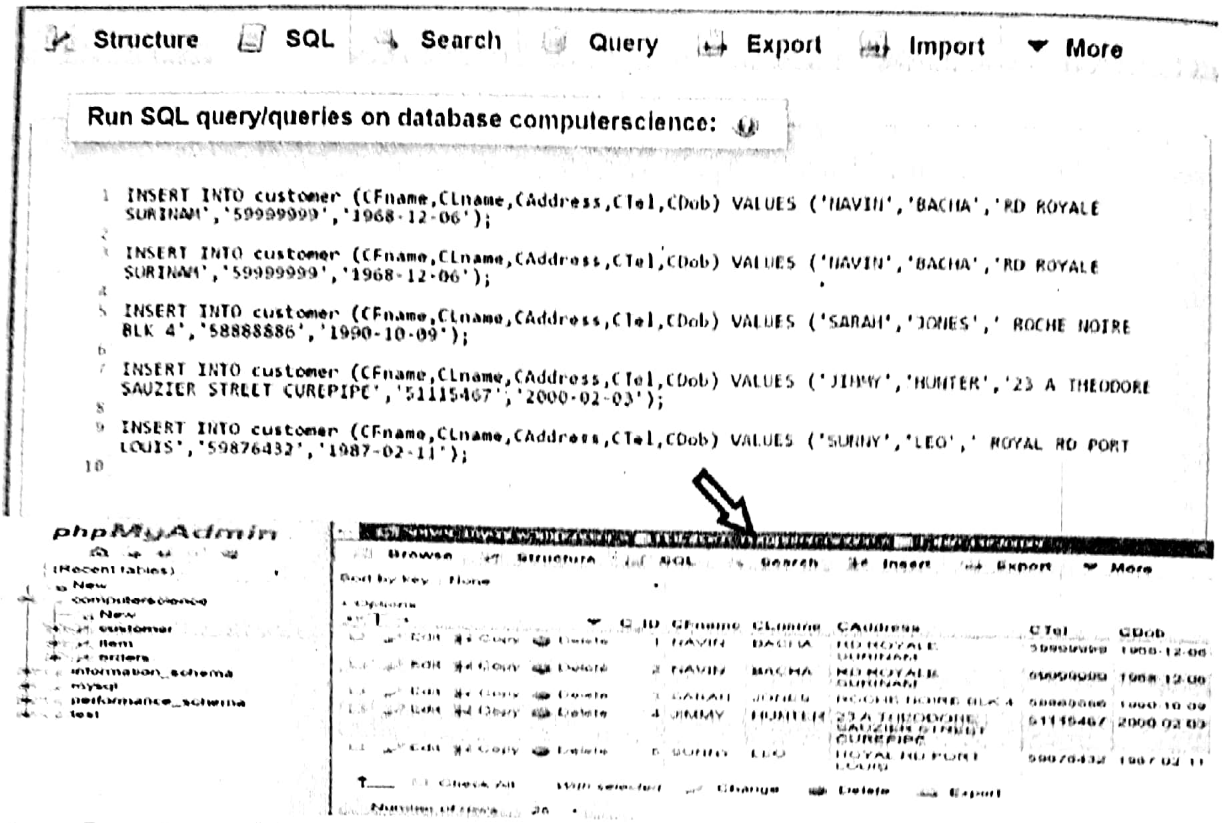
### SQL INSERT INTO Statement

Format: **INSERT INTO** table_name(column1, column2,.....,columnN) **VALUES** (value1, value2,.....,valueN);

Explanation: The INSERT INTO command simply adds a record to a specified table with respective field data given in that sequence.

**Insert Data into table customer**

- **INSERT INTO** customer (CFname,CLname,CAddress,CTel,CDob) **VALUES** ('NAVIN','BACHA','RD ROYALE SURINAM','59999999','1968-12-06');
- **INSERT INTO** customer (CFname,CLname,CAddress,CTel,CDob) **VALUES** ('NAVIN','BACHA','RD ROYALE SURINAM','59999999','1968-12-06');
- **INSERT INTO** customer (CFname,CLname,CAddress,CTel,CDob) **VALUES** ('SARAH','JONES',' ROCHE NOIRE BLK 4','58888886','1990-10-09');
- **INSERT INTO** customer (CFname,CLname,CAddress,CTel,CDob) **VALUES** ('JIMMY','HUNTER','23 A THEODORE SAUZIER STREET CUREPIPE','51115467','2000-02-03');
- **INSERT INTO** customer (CFname,CLname,CAddress,CTel,CDob) **VALUES** ('SUNNY','LEO',' ROYAL RD PORT LOUIS','59876432','1987-02-11');

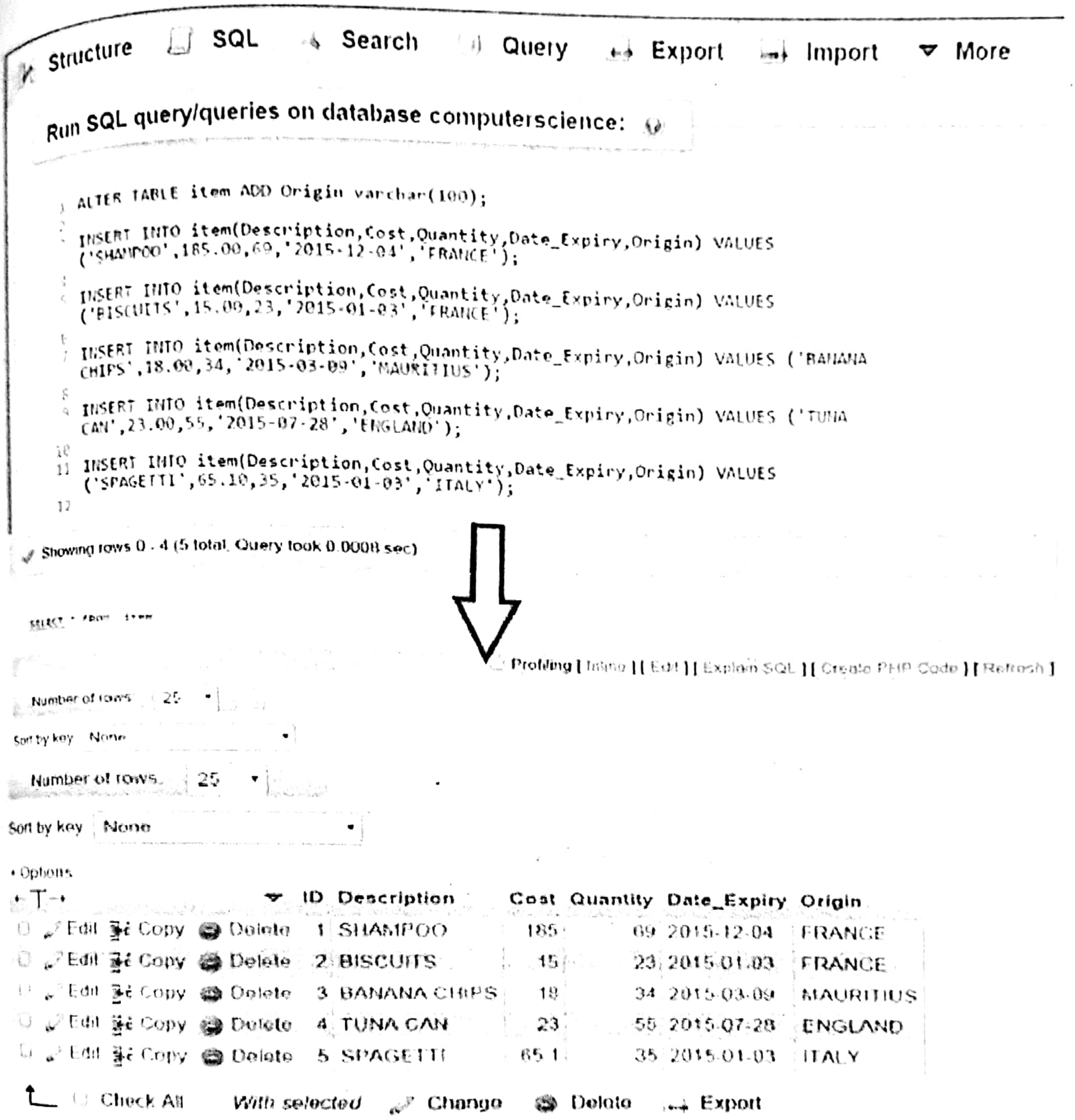


Run SQL query/queries on database computerscience:

```
INSERT INTO customer (CFname,CLname,CAddress,CTel,CDob) VALUES ('NAVIN','BACHA','RD ROYALE
SURINAM','59999999','1968-12-06');

INSERT INTO customer (CFname,CLname,CAddress,CTel,CDob) VALUES ('NAVIN','BACHA','RD ROYALE
SURINAM','59999999','1968-12-06');

INSERT INTO customer (CFname,CLname,CAddress,CTel,CDob) VALUES ('SARAH','JONES',' ROCHE NOIRE
BLK 4','58888886','1990-10-09');

INSERT INTO customer (CFname,CLname,CAddress,CTel,CDob) VALUES ('JIMMY','HUNTER','23 A THEODORE
SAUZIER STREET CUREPIPE','51115467','2000-02-03');

INSERT INTO customer (CFname,CLname,CAddress,CTel,CDob) VALUES ('SUNNY','LEO',' ROYAL RD PORT
LOUIS','59876432','1987-02-11');
```

**Insert Data into table item**

- **INSERT INTO** item(Description,Cost,Quantity,Date_Expiry,Origin) **VALUES** ('SHAMPOO',185.00,69,'2015-12-04','FRANCE')
- **INSERT INTO** item(Description,Cost,Quantity,Date_Expiry,Origin) **VALUES** ('BISCUITS',15.00,23,'2015-01-03','FRANCE')
- **INSERT INTO** item(Description,Cost,Quantity,Date_Expiry,Origin) **VALUES** ('BANANA CHIPS',18.00,34,'2015-03-09','MAURITIUS')
- **INSERT INTO** item(Description,Cost,Quantity,Date_Expiry,Origin) **VALUES** ('TUNA CAN',23.00,55,'2015-07-28','ENGLAND')
- **INSERT INTO** item(Description,Cost,Quantity,Date_Expiry,Origin) **VALUES** ('SPAGETTI',65.10,35,'2015-01-03','ITALY')



Run SQL query/queries on database computerscience:

```
ALTER TABLE item ADD Origin varchar(100);

INSERT INTO item(Description,Cost,Quantity,Date_Expiry,Origin) VALUES
('SHAMPOO',185.00,69,'2015-12-04','FRANCE');

INSERT INTO item(Description,Cost,Quantity,Date_Expiry,Origin) VALUES
('BISCUITS',15.00,23,'2015-01-03','FRANCE');

INSERT INTO item(Description,Cost,Quantity,Date_Expiry,Origin) VALUES ('BANANA
CHIPS',18.00,34,'2015-03-09','MAURITIUS');

INSERT INTO item(Description,Cost,Quantity,Date_Expiry,Origin) VALUES ('TUNA
CAN',23.00,55,'2015-07-28','ENGLAND');

INSERT INTO item(Description,Cost,Quantity,Date_Expiry,Origin) VALUES
('SPAGETTI',65.10,35,'2015-01-03','ITALY');
```

✓ Showing rows 0 - 4 (5 total, Query took 0.0008 sec)

```
SELECT * FROM item
```

Number of rows: 25

Sort by key: None

Number of rows: 25

Sort by key: None

| ID | Description | Cost | Quantity | Date_Expiry | Origin |
|---|---|---|---|---|---|
| 1 | SHAMPOO | 185 | 69 | 2015-12-04 | FRANCE |
| 2 | BISCUITS | 15 | 23 | 2015-01-03 | FRANCE |
| 3 | BANANA CHIPS | 18 | 34 | 2015-03-09 | MAURITIUS |
| 4 | TUNA CAN | 23 | 55 | 2015-07-28 | ENGLAND |
| 5 | SPAGETTI | 65.1 | 35 | 2015-01-03 | ITALY |

## Insert Data into table orders

- **INSERT INTO** orders(I_ID,C_ID,ODate,Qty) **VALUES** (1,5,'2015-02-03',3);
- **INSERT INTO** orders(I_ID,C_ID,ODate,Qty) **VALUES** (2,4,'2015-04-02',5);
- **INSERT INTO** orders(I_ID,C_ID,ODate,Qty) **VALUES** (3,3,'2015-05-27',6);
- **INSERT INTO** orders(I_ID,C_ID,ODate,Qty) **VALUES** (4,2,'2015-12-30',2);
- **INSERT INTO** orders(I_ID,C_ID,ODate,Qty) **VALUES** (5,1,'2015-06-06',7);
- **INSERT INTO** orders(I_ID,C_ID,ODate,Qty) **VALUES** (1,5,'2015-04-09',3);
- **INSERT INTO** orders(I_ID,C_ID,ODate,Qty) **VALUES** (4,3,'2015-03-31',5);

# 10.0 Hardware – Logic Gates

## 10.1 Logic Gates

***A gate is an electronic device that produces a result based on one or more input values. In reality, gates consist of one to six transistors, but digital designers think of them as a single unit Integrated circuits contain collections of gates suited to a particular purpose***

Logic gates process signals which represent true or false. These are the basic components that make up digital logic. Normally the Positive supply voltage represents true (1) and (0) represents false.

***Gates are identified by their functions and their name is in capital letters. Gates have two or more inputs, except NOT gate which has only 1 input. All gates usually have one Output.***

Usually letters **A, B, C, D, and E** and so on are used as label to inputs and **Q or F** is labelled as output.

Some gate symbols have a circle on their output which means that their function includes inverting the output. It is equivalent to feeding the output through a NOT gate.

**Truth Table**

***A truth table is a good way to show the function of a logic gate. It shows the output states for every possible combination of input states. The symbol 0 (false) and 1 (true) are usually used in truth tables.***

**Note:** The number of inputs in a gate yields to $2^n$ possibilities where n is number of inputs. The example truth table shown below shows the input and output of an AND gate.( n = 2)

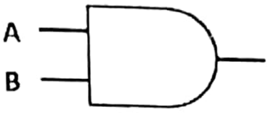
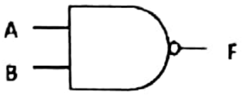
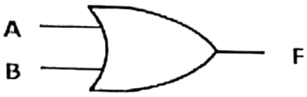
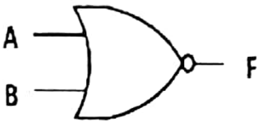
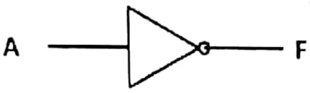
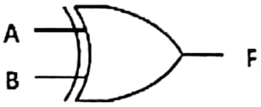
| Input A | Input B | Output Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## 10.1 Types of Logic Gates

- ✓ AND Gate
- ✓ NAND Gate
- ✓ OR Gate
- ✓ NOR Gate
- ✓ NOT Gate
- ✓ XOR

**Note:** NAND & NOR Gate are known as universal gate and they can represent the logic of any other gates by multiple combination.

## 10.2 Logic Gate Symbols

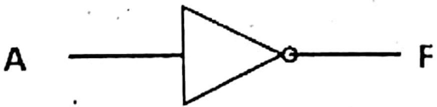
| GATE | BOOLEAN EXPRESSION | SYMBOL | GATE | BOOLEAN EXPRESSION | SYMBOL |
|---|---|---|---|---|---|
| AND | $F = A \bullet B$ |  | NAND | $F = \overline{A \bullet B}$ |  |
| OR | $F = A + B$ |  | NOR | $F = \overline{A + B}$ |  |
| NOT | $F = \overline{A}$ |  | XOR | $F = A \oplus B$ |  |

## 10.3 NOT GATE

***The output F is true when the input A is NOT true***; the output is the inverse of the input.

F = NOT (A)

A NOT gate can only have one input and is also called an inverter. (A "**o**" is found on the output)

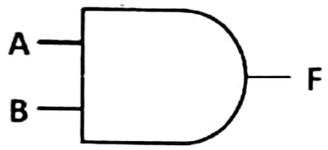


Truth Table for NOT Gate (Inverter)

| A | F ( Ā) |
|---|---|
| 0 | 1 |
| 1 | 0 |

## 10.4 AND Gate with 2 inputs

***The output F is true only if input A and B are true. F = A AND B.*** An AND gate can have two or more inputs; its output is true only if all its inputs are true.



AND gate is similar to Multiplication. Tips to Remember for exam, let's say the following trick

- **A** : You wake up Early
- **B** : You have done your home work
- **F** : You will go to school

| A | B | $F = A \bullet B$ |
|---|---|---|
| FALSE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| TRUE | FALSE | FALSE |
| TRUE | TRUE | TRUE |

Replace ALL FALSE BY 0 and TRUE by 1. You will go to school only if you have wake up early AND you have done your homework. Both Conditions have to be true then you will have true as result

Truth Table for 2 Inputs AND Gate

Number of Inputs (n) => 2, so we have $2^2$ possibilities

| A | B | $F = A \bullet B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| **1** | **1** | **1** |

## 10.5 NAND Gate

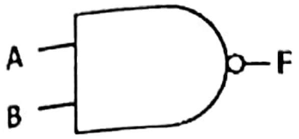
This is an AND gate with the output inverted, as shown by the 'o' on the output.

***The output is true if input A AND input B are NOT both true: F = NOT (A AND B).***

A NAND gate can have two or more inputs; its output is true if NOT all inputs are true. It is simply the output of a AND Gate followed by that of a NOT Gate

### Truth Table for 2 Inputs NAND Gate

Number of Inputs (n) => 2, so we have $2^2$ possibilities



| A | B | $F = \overline{A \cdot B}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

### Binary OR Rule

- ✓ 0 + 0 = 0
- ✓ 0 + 1 = 1
- ✓ 1 + 0 = 1
- ✓ 1+ 1 = 1
- ✓ 1 +1 +1 = 1

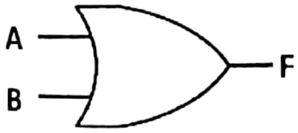
### 10.6 OR Gate with 2 inputs

***The output F is true if input A OR input B is true (or both of them are true). F = A OR B***

An OR gate can have two or more inputs, its output is true if at least one input is true. OR gate is similar to binary addition

**TIP:** IF you see a 1 in any input, using OR Gate, the output will always be one since one input is true.

Number of Inputs (n) => 2, so we have $2^2$ possibilities



| A | B | $F = A + B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

### 10.7 NOR Gate with 2 input

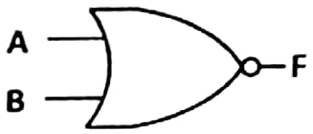
This is an OR gate with the output inverted, as shown by the 'o' on the output. The output Q is true if both inputs A and B is false. Q = NOT (A OR B)

A NOR gate can have two or more inputs; its output is true if inputs are true.

**Tip:** Simply inverse the output of an OR gate

Truth Table for 2 Inputs NOR Gate

Number of Inputs (n) => 2, so we have $2^2$ possibilities



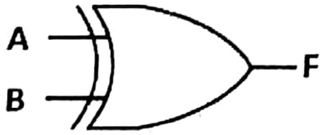
| A | B | F ( $\overline{A + B}$ ) |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

## 10.8 EX-OR Gate (XOR Gate)

The **XOR gate** (sometimes **EOR gate**, or **EXOR gate**) is a digital logic gate that implements an exclusive or; that is, a true output (1) results if one, and only one, of the inputs to the gate is true (1).

If both inputs are false (0) and both are true (1), a false output (0) results. A way to remember XOR is "one or the other but not both". It can only have 2 inputs.

Number of Inputs (n) => 2, so we have $2^2$ possibilities



| A | B | F = A ⊕ B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**SOP form of XOR Gate =** $\overline{A}B + A\overline{B}$ (Discussed later in Paper 3)

## 10.9 Universal Gate

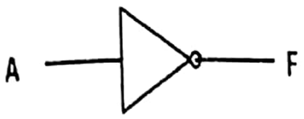
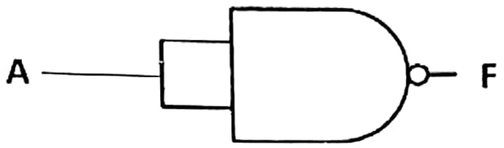
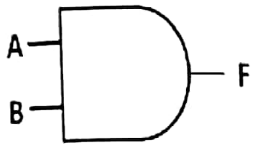
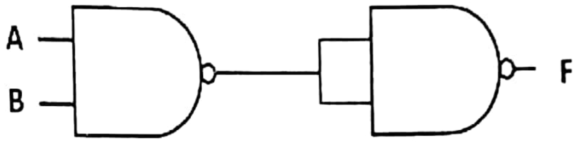
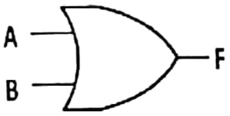
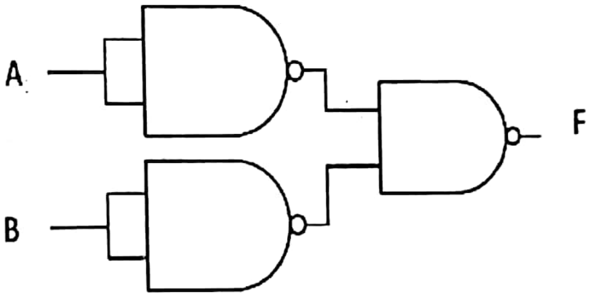
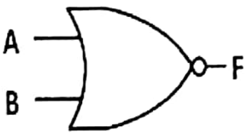
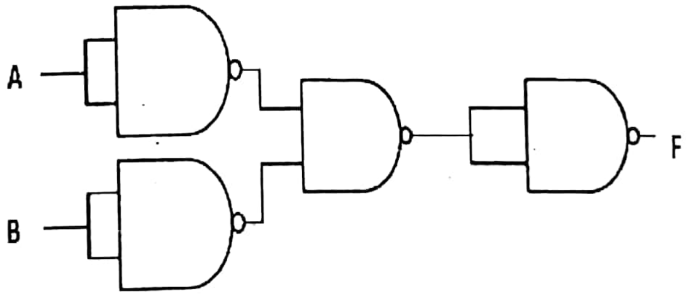
A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates.

In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families. In fact, an AND gate is typically implemented as a NAND gate followed by an inverter not the other way around!!

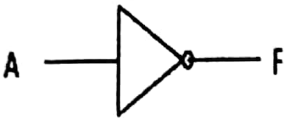
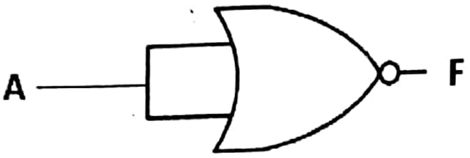
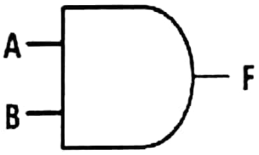
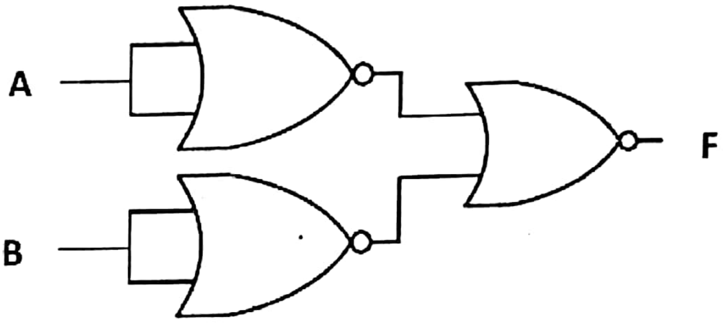
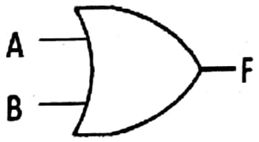
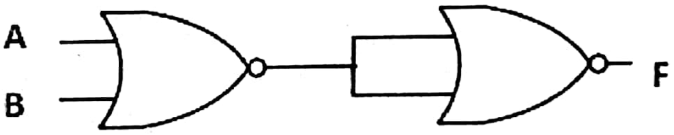
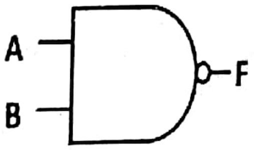
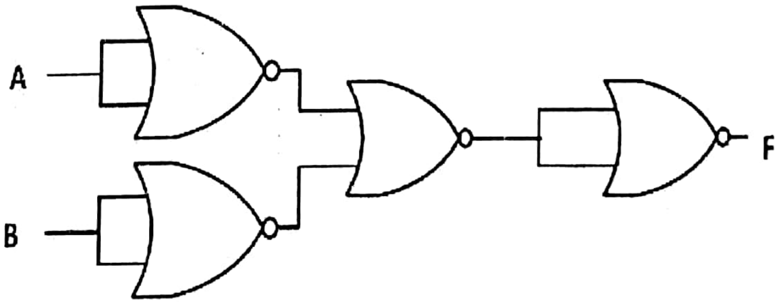
Likewise, an OR gate is typically implemented as a NOR gate followed by an inverter not the other way around!!

### 10.9.1 Logic Gate Equivalent using NAND Gate

Note: There may be several representation possible using different combinations.

| | Gate | Equivalent in NAND Gates |
|---|---|---|
| NOT |  |  |
| AND |  |  |
| OR |  |  |
| NOR |  |  |

### 10.9.2 Logic Gate Equivalent using NOR Gate

| | Gate | Equivalent in NOR Gate |
|---|---|---|
| **NOT** |  |  |
| **AND** |  |  |
| **OR** |  |  |
| **NAND** |  |  |

### 10.10 Combinations of Logic Gates

It is possible to combine logic gates together to produce more complex logic networks.

We shall consider two scenario which are possible normally

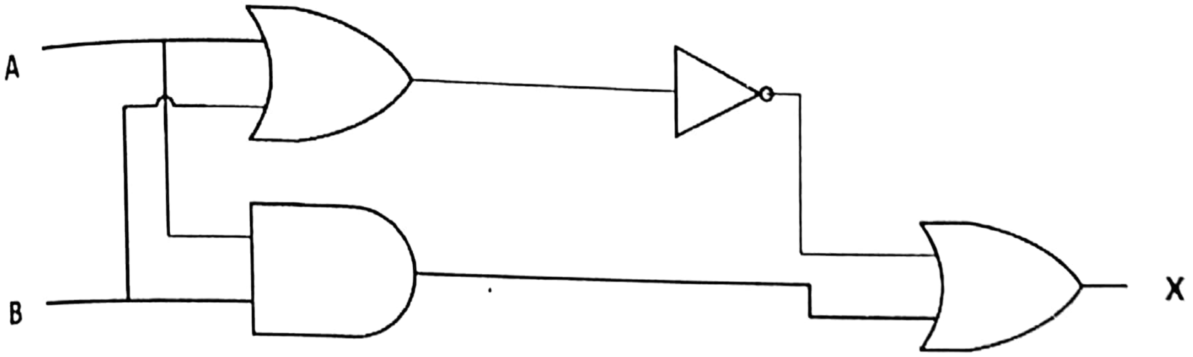
a) Drawing the truth table from a given logic network
b) Designing a logic network for a given problem and then testing it by drawing the truth table.

## Procedure to produce truth table

1. Find the Number of Inputs in the combined circuit (n)
2. Create Truth table to represent the number of input possibilities
3. Label Interim gates with Letters Preceding the same level
4. Arrange your letters in such a way to proceed along with inputs for your truth table
5. Write the letters gate function for verification purpose and to avoid errors or confusion
6. Draw Final Truth Table

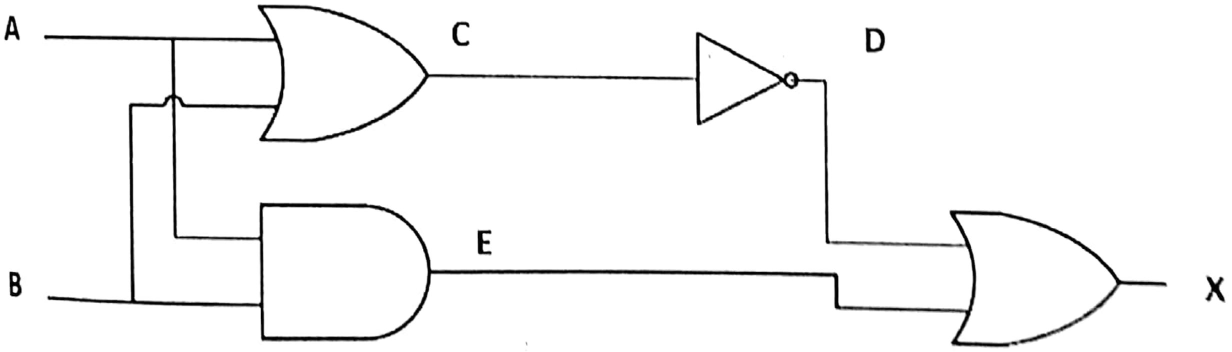
## Example 1 – Combined Circuit with 2 inputs

Consider the Example Given Below



**Step 1**: Number of Inputs in this case is A, B => 2, therefore, No of Possibilities -> $2^2$

**Step 2**: Label the interim gates with letters sequentially so as to draw your truth table easily (C, D, and E)



**Step 3**: Draw Truth Table with appropriate headings

| A | B | C | D | E | X |
|---|---|---|---|---|---|
| 0 | 0 | | | | |
| 0 | 1 | | | | |
| 1 | 0 | | | | |
| 1 | 1 | | | | |

**Step 4**: Write the Gate Function output from each intermediate letters you have added

I. C = A **OR** B
II. D = **NOT** C
III. E = A **AND** B
IV. X = D **OR** E

**Step 5**: Construct the Truth Table according to logic gate operations learnt previously and thei truth table behaviours

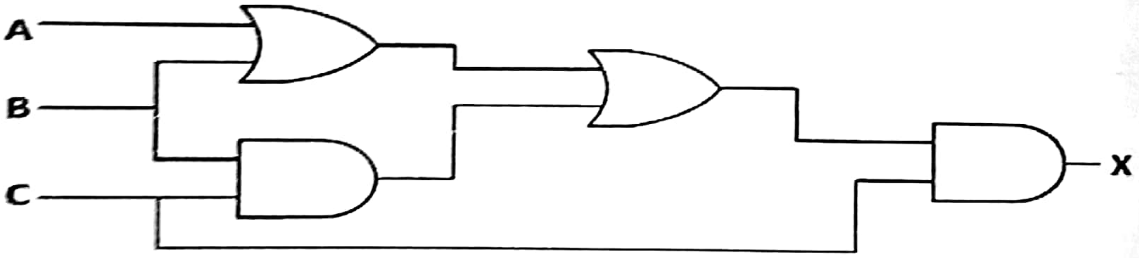
| A | B | C | D | E | X |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

**Step 6**: List only Final truth Table

| A | B | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

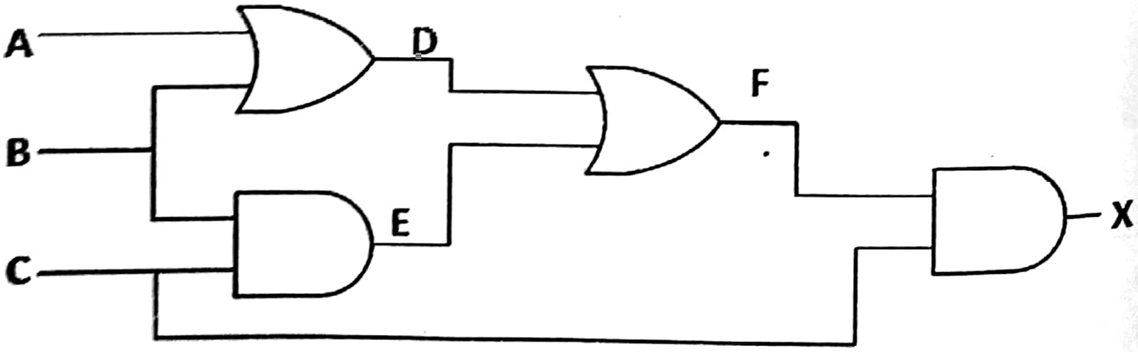
### Example 2 – Combined Circuit with 3 inputs

Consider the Example Given Below



**Step 1**: Number of Inputs in this case is A, B and C => 3, therefore, No of Possibilities -> $2^3$

**Step 2**: Label the interim gates with letters sequentially so as to draw your truth table easily (D, E and F)



**Step 3**: Draw Truth Table with appropriate headings

| A | B | C | D | E | F | X |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | | | |
| 0 | 0 | 1 | | | | |
| 0 | 1 | 0 | | | | |
| 0 | 1 | 1 | | | | |
| 1 | 0 | 0 | | | | |
| 1 | 0 | 1 | | | | |
| 1 | 1 | 0 | | | | |
| 1 | 1 | 1 | | | | |

**Step 4**: Write the Gate Function output from each intermediate letters you have added

I. D = A **OR** B
II. E = B **AND** C
III. F = D **OR** E
IV. X = F **AND** C

**Step 5**: Construct the Truth Table according to logic gate operations learnt previously and their truth table behaviours

| A | B | C | D | E | F | X |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Step 6**: List only Final truth Table

| A | B | C | X |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

### Example 3: Combined Circuit from Description Given

An alarm, X, gives a signal (i.e. X = 1) when a car fuel injection system gives certain faul conditions.

The inputs are:

| input | Binary value | Condition |
|---|---|---|
| P | 0 | pressure < 5 bar |
|  | 1 | pressure >= 5 bar |
| Q | 0 | revs > 8000 rpm |
|  | 1 | revs <= 8000 rpm |
| R | 0 | temp > 120 °C |
|  | 1 | temp <= 120 °C |

The alarm returns a value of 1 if:

**Either** (i) pressure < 5 bar AND revs > 8000 rpm

**Or** (ii) revs <= 8000 rpm AND temp > 120 °C

**Procedure:**

1. Count the Number of Inputs
2. Read Instruction and Convert to appropriate statement interpretation by referring to the table given
3. Express statement in terms of given gate
4. Note: These questions usually ask the user to construct the logic gate circuit using given
5. Draw Logic Gate Diagram and Label properly

**Step 1:**

- ✓ No of Inputs : P,Q,R => 3 , therefore , No of Possibilities = $2^3$

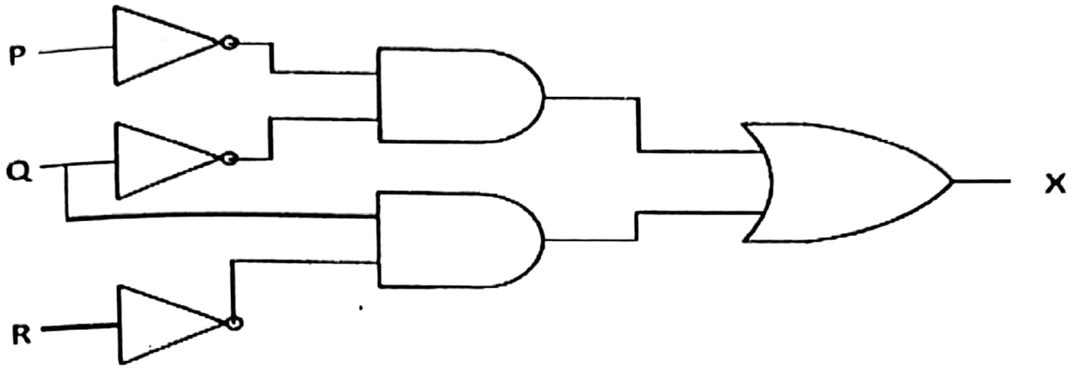
**Step 2:**

- ✓ **Statement 1** : pressure < 5 bar AND revs > 8000 rpm
- ✓ Referring to the table given for Statement 1 we get: (P = 0 AND Q = 0)
- ✓ **Statement 2** : revs <= 8000 rpm AND temp > 120 °C
- ✓ Referring to the table given for Statement 2 we get: (Q = 1 AND R = 0)
- ✓ From the question statement definition, it says alarm X runs if either statement 1 or 2 is true
- ✓ Hence: X = 1 IF Statement 1 OR Statement 2
- ✓ X = 1 if (P = 0 **AND** Q = 0) **OR** (Q = 1 **AND** R = 0)

We usually convert the Zero output to NOT one since it is easier to simplify circuits

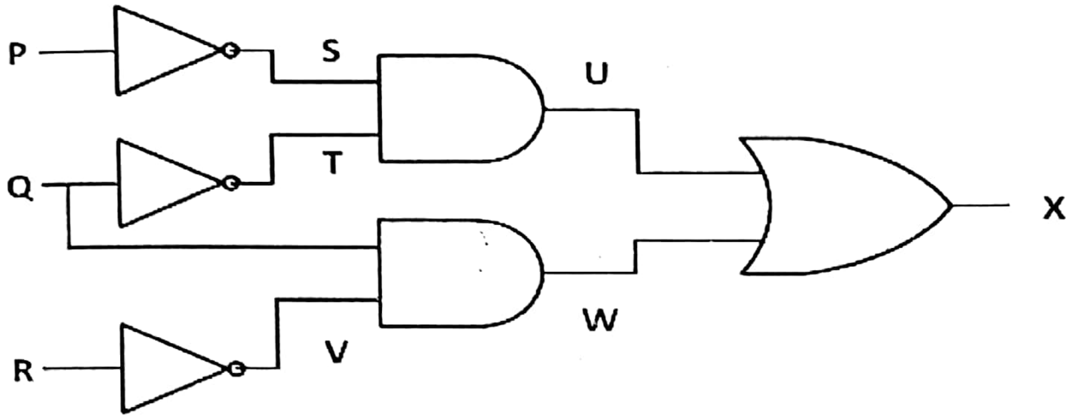
- ✓ Hence , X = 1 if (P = NOT 1 **AND** Q = NOT 1) **OR** (Q = 1 **AND** R = NOT 1)

✓ Therefore, we draw the diagram as shown below using traditional symbols



✓ Finally we construct the Truth table repeating steps with learnt previously

✓ We label the diagram with appropriate letters



✓ We write the letters logic gate function they represent

I. S = NOT P , T = NOT Q
II. U = S AND T , V = NOT R
III. W = Q AND V , X = U OR W

✓ Construct truth table accordingly

| P | Q | R | S | T | U | V | W | X |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

# 11.0 Communication and Internet Technologies – Network

## 11.1 Computer Network

***A computer network** is defined as an interconnection of devices together to share information and services.*

There are several types of services and data that can be shared on a network. These include but are not limited to documents, email, database, hardware (printers, faxes), telephony, videoconferencing and others.

**Main Components of Network**

| | |
|---|---|
| ✓ **Devices** | - these are used to communicate with each other |
| ✓ **Medium** | - refers to how devices are connected together (Wired / Wireless) |
| ✓ **Rules** | - these govern how messages flow across the network |
| ✓ **Messages** | - these refer to the information that travels over the medium |

## 11.2 Network Architectures

These refer to the way devices are connected and the way host connects to the network.

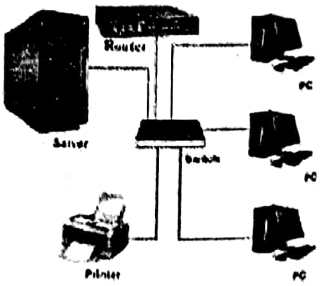
A pc or host is any device connected to a network. Sometimes it is also called a node. A host can have several functions which include:

| | |
|---|---|
| ✓ Function as a client | – request for data |
| ✓ Function as a server | – provide data |
| ✓ Function as peer | – request and provide data |

Based on these varying functions, several network architectures have been developed including the 3 main below:

- ✓ Client Server Model
- ✓ Peer to Peer Model
- ✓ Mainframe / Terminal Model

### 11.2.1 Client Server Networks



This is the most common network infrastructure used in corporate environment. Clients request data and services stored on servers. The servers hold data and provide services. Network operating system (NOS) on server manage resources. The NOS provides:

- Network directory services
- Network security, monitoring, and auditing features
- Architectural framework (APIs) to support server-based applications

Examples of client server model are video sharing website, shared printing service in a university or school

### 11.2.2 Client/Server Authentication

The client OS sends login information to directory server. The directory server is responsible for user authentication. During the login process, client communicates with NOS on server.

| Advantage | Disadvantage |
|---|---|
| ▪ Simplifies management as it is centralised.<br>▪ Easier to secure and back up<br>▪ Scales larger than peer-to-peer network | ▪ It introduces a single point of failure. I.e. if server goes down, whole network stops to function. |

### 11.2.3 Peer to Peer Model

In this type of networking model, all hosts on the network can both request and provide data and services.

#### Peer to Peer authentication

User account usually includes username, password, permissions etc. and exists on a single computer.

| Advantage | Disadvantage |
|---|---|
| ▪ Simple file and resource sharing | ▪ Difficult to maintain security<br>▪ Difficult to manage and backup data |

## 11.3 Network Equipment

#### Modems

A **modem** (**modulator-demodulator**) is a device that modulates an analogue carrier signal to encode digital information, and also demodulates such a carrier signal to decode the transmitted information. The goal is to produce a signal that can be transmitted easily and decoded to reproduce the original digital data.

Modems can be used over any means of transmitting analogue signals, from light emitting diodes to radio. The most familiar example is a voice band modem that turns the digital data of a personal computer into modulated electrical signals in the voice frequency range of a telephone channel.

These signals can be transmitted over telephone lines and demodulated by another modem at the receiver side to recover the digital data. Modems are generally classified by the amount of data they can send in a given unit of time, usually expressed in bits per second (bit/s, or bps). Modems can alternatively be classified by their symbol rate, measured in baud.

The *baud* unit denotes symbols per second, or the number of times per second the modem sends a new signal.

## Bridge

A bridge reads the outermost section of data on the data packet, to tell where the message is going. It reduces the traffic on other network segments, since it does not send all packets. Bridges can be programmed to reject packets from particular networks. Bridging occurs at the data link layer of the OSI model, which means the bridge cannot read IP addresses, but only the outermost hardware address of the packet. In our case the bridge can read the Ethernet data which gives the hardware address of the destination address, not the IP address. Bridges forward all broadcast messages. Only a special bridge called a translation bridge will allow two networks of different architectures to be connected. Bridges do not normally allow connection of networks with different architectures.

## Repeater

A repeater connects two segments of your network cable. It retimes and regenerates the signals to proper amplitudes and sends them to the other segments. When talking about, Ethernet topology, you are probably talking about using a hub as a repeater. Repeaters require a small amount of time to regenerate the signal. This can cause a propagation delay which can affect network communication when there are several repeaters in a row. Many network architectures limit the number of repeaters that can be used in a row. Repeaters work only at the physical layer of the OSI network model.

## Server

A server is a computer on a network that provides a resource to the other machines on the network. There are several different types of server on networks, distinguished from each other by the service that they provide on the network. Typical servers are the file server and the printer server. They offer services exactly as their names suggest and are there to provide a centralised resource.

## Types of Servers

**Mail Server** move and store mail over corporate networks (via LANs and WANs) and across the Internet.

**Proxy servers** sit between a client program (typically a Web browser) and an external server (typically another server on the Web) to filter requests, improve performance, and share connections.

**A Web server** serves static content to a Web browser by loading a file from a disk and serving it across the network to a user's Web browser. This entire exchange is mediated by the browser and server talking to each other using HTTP.

**An application server** occupies a large chunk of computing territory between databases and the end user and they often connect the two.

### Gateway

A gateway is used to connect a LAN to a WAN (possibly the Internet). It provides a connection along with ensuring that material passed from one to the other is appropriate and it also monitors the usage of the connection. The gateway deals with distinct communication protocols between the LAN and the other networks that it connects.

### Network Interface Card

Often abbreviated as NIC, an expansion board you insert into a computer so the computer can be connected to a network. Most NICs are designed for a particular type of network, protocol, and media, although some can serve multiple networks.

### Switch

A switch is a device that filters and forwards packets between LANs segments for which it was intended for. Switches operate at the data link layer (layer 2) and sometimes the network layer (layer 3) of the OSI Reference Model and therefore support any packet protocol. LANs that use switches to join segments are called switched LANs. Since it forward packet to required destination only, it optimise network traffic by setting up a temporary dedicated circuit between and sender and recipient devices. As soon as the transfer is completed, the circuit cease to exist.

E.g. Cisco Switches, 3com Switch, D-Link Switch

### Router

***A router is a device that forwards data packets along networks. A router is connected to at least two networks, commonly two LANs or WANs or a LAN and its ISP's network.*** Routers are located at gateways, the places where two or more networks connect. It chooses the best path to send data from one point to another E.G Popular Routers: Cisco Routers, D-link, Linksys, 3Com

***Routing is the process of sending a packet of information from one network to another network.*** Routers use headers and forwarding tables to determine the best path for forwarding the packets, and they use protocols such as ICMP to communicate with each other and configure the best route between any two hosts. Router configuration ensures that data packets are 'routed' to the correct destination with minimum loss and disruption.

### WAP

It is a device that allows wireless clients (devices) to connect to a wireless network using WIFI standards. The WAPs often serve as a gateway between the wired and wireless networks. Wireless client are centrally connected via a **wireless access point (WAP).** This is referred to as an **infrastructure** connection, or a Basic Service Set (BSS).

Wireless clients must associate with a WAP before data can be forwarded.

**In environments** where a single WAP does not provide sufficient coverage, multiple WAPs can **be linked** as part of an **Extended Service Set (ESS). Most** WAP support multiple wireless **connection** to one wired connection.

## 11.4 WWW (World Wide Web)

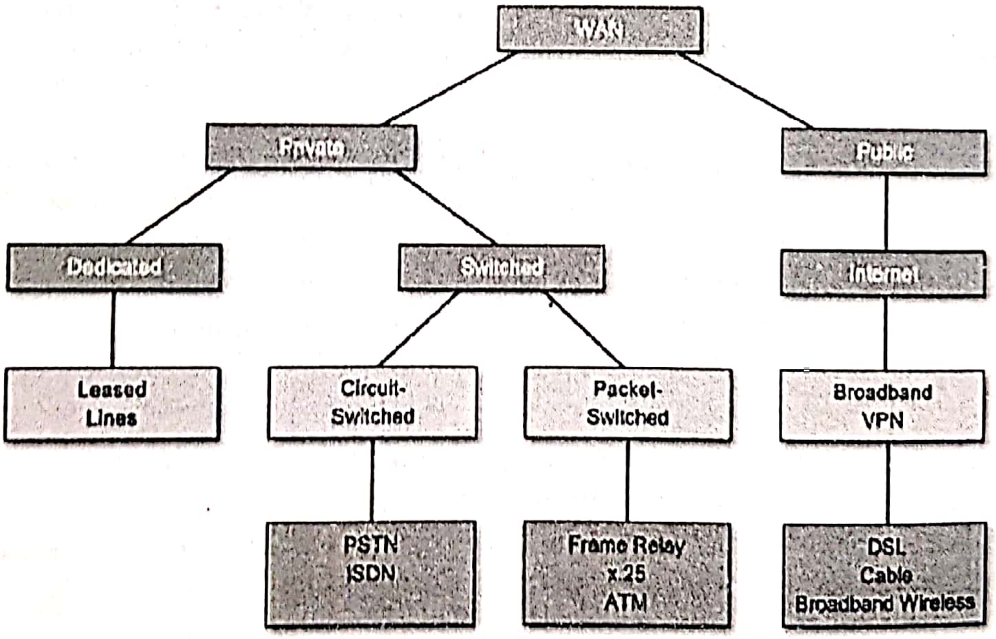
It is defined as a system of interlinked hypertext documents that can be accessed via the internet using a browser. It was invented by British computer Scientist, Sir Tim Berners Lee in 1989.

## 11.5 Communication systems to support the Internet

| WAN Connection | Description |
|---|---|
| Dial Up Networking | Uses a modem to connect through regular analogue phone line |
| Digital Subscriber Line | High speed connections made over regular analogue phone lines |
| Cable | Connections made over the same lines that carry cable television signals |
| Satellite | Connections made by sending signals to and receiving signals from satellite in orbit around the earth |
| Wireless | Connections made through infrared or radio waves. Note connection can also be made over cellular telephone networks or via satellites |
| Cellular | Connections made through a cell phone or laptop's cellular network PC card on a cellular phone network. |

## 11.6 WAN Connection Types

WAN Link Connection Options



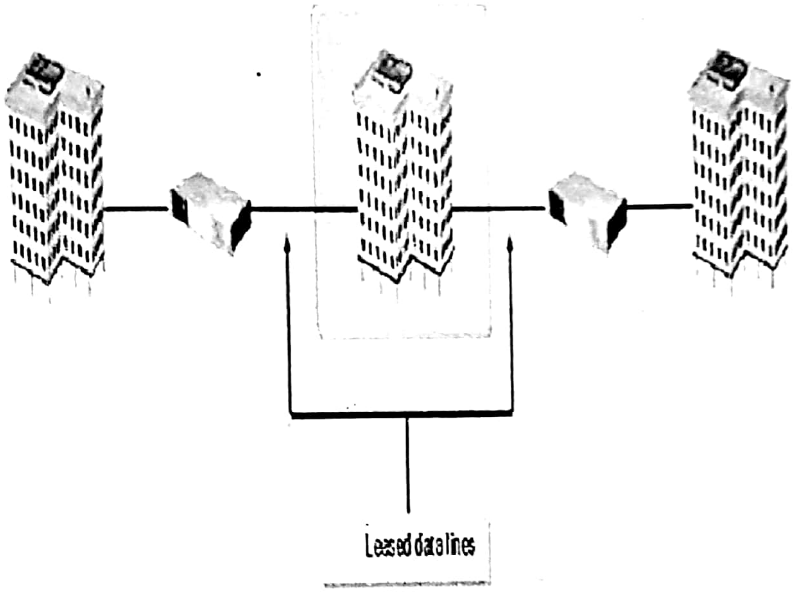
WANs are generally grouped into 3 separate connection types:

1. Point to Point technologies
2. Circuit switched technologies
3. Packet switched technologies

### 11.6.1 Point to Point

These type of technologies are often called dedicated or leased lines and is among the most expensive form of WAN technology. The lines are leased from an Internet Service Provider who guarantees bandwidth from one location to another. The cost is often determined by the distance and amount of bandwidth required. Point to Point connection requires no call set up and is always one.

E.g. T1 Lines, T3 Lines & E Lines



### 11.6.2 Circuit Switched

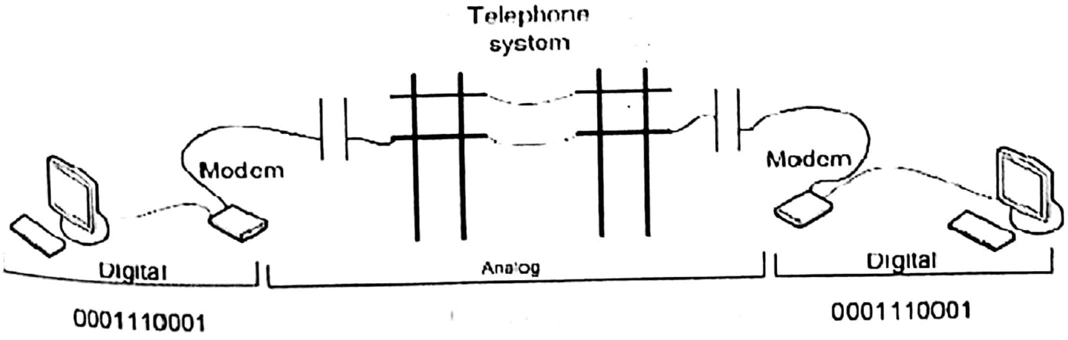
These type of technologies require that a connection is setup before data transmission. (Call set up). The session is destroyed once data transfer is complete and is often known as On Demand Circuit. Its price is generally low compared to point to point. E.g. Dial-Up, ISDN

### PSTN (POTS)

**PSTN** – Public Switched Telephone Networks

**POTS** - Plain old telephone System

The Public Switched Telephone Network uses a dial-up system over telephone lines. The connection is not continuous. The data and phone share the same line and allows only one to be used as a time. Such system usually has a max data speed of 56 Kbps and provide remote access.



### 11.6.3 Packet Switched Technologies

These type of technologies often shares a common infrastructure between all provider's subscribers. Hence, bandwidth is not guaranteed but is rather allocated on best effort basis. Normally such technologies does not suit applications which require consistent bandwidth but its price is less expensive than that of point to point. E.g. of packet switched technologies include: Frame Relay and X25.

The packet switched network moves data through the network packet by packet. Each packet takes the best route available at any given time rather that an established route.

### 11.6.4 Cellular Networks

These types of networks use are very complex. It requires the use of a central access point which is a cell tower connected to a main hub. There are two common major cell standards used such as GSM and CDMA.

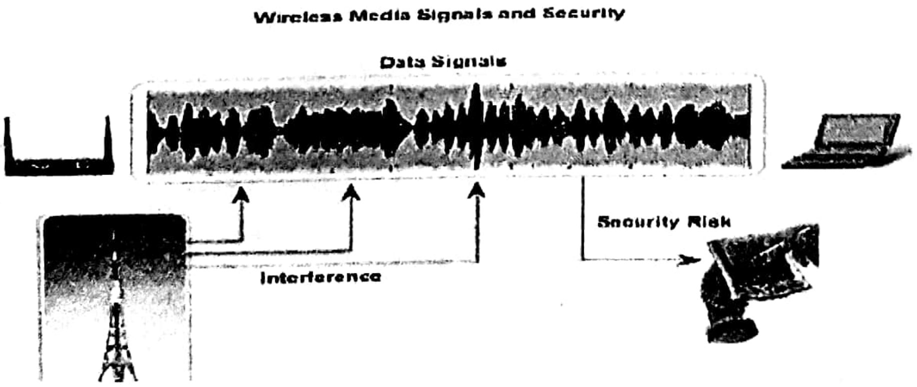
## 11.7 Network Transmission Technologies Pros and Cons

Microwaves

| Advantage | Disadvantage |
|---|---|
| ▪ Theses waves pass through the ionosphere and hence are suitable for satelite to earth transmission.<br>▪ It can be altered to carry many signals at one time including data, televisions pictures and voice messages. | ▪ It is absorbed easily by nature<br>▪ We need special aerials to receive them |

Radio Waves

Radio frequencies are radiated into the air via an antenna which creates radio waves. The higher the frequency, the higher is the data rate but they have a shorter range.



| Advantage | Disadvantage |
|---|---|
| - It can travel the earth since some are reflected off the ionosphere.<br>- Carry carry a message instantaneously over a wide area.<br>- Aerials to receive them are simpler than for microwaves | - Radio waves does not cover all area<br>- Radio waves are susceptible to infererence and noise<br>- Radio waves has a range of frequencies that can be accessed by existing devices is limited and hence there is lot of competiton amidst companies for the use of the frequencies |

### Satellite

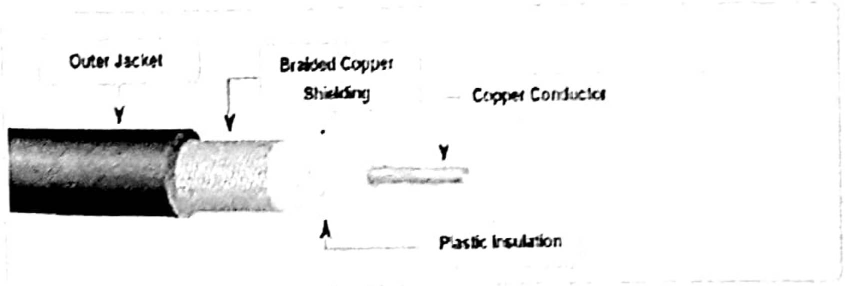
Instead of using cables, signals are transmitted through air whereby a service providers beams a microwave signal from a dish on the ground to an orbiting satellite. The latter then sends the signal back down to the receivers. Usually receivers are small satellite dishes but can also be portable modems or phones. This if often referred to as point to multi point because the satellite can provide signal to a number of receivers.

| Advantage | Disadvantage |
|---|---|
| - Can be used anywhere in the world where satellite signal is available | - Installation can be tricky<br>- Requires Line of Sight<br>- Latency can be a problem<br>- Connections are pretty slow compared to wired cabling |

### Coaxial cable

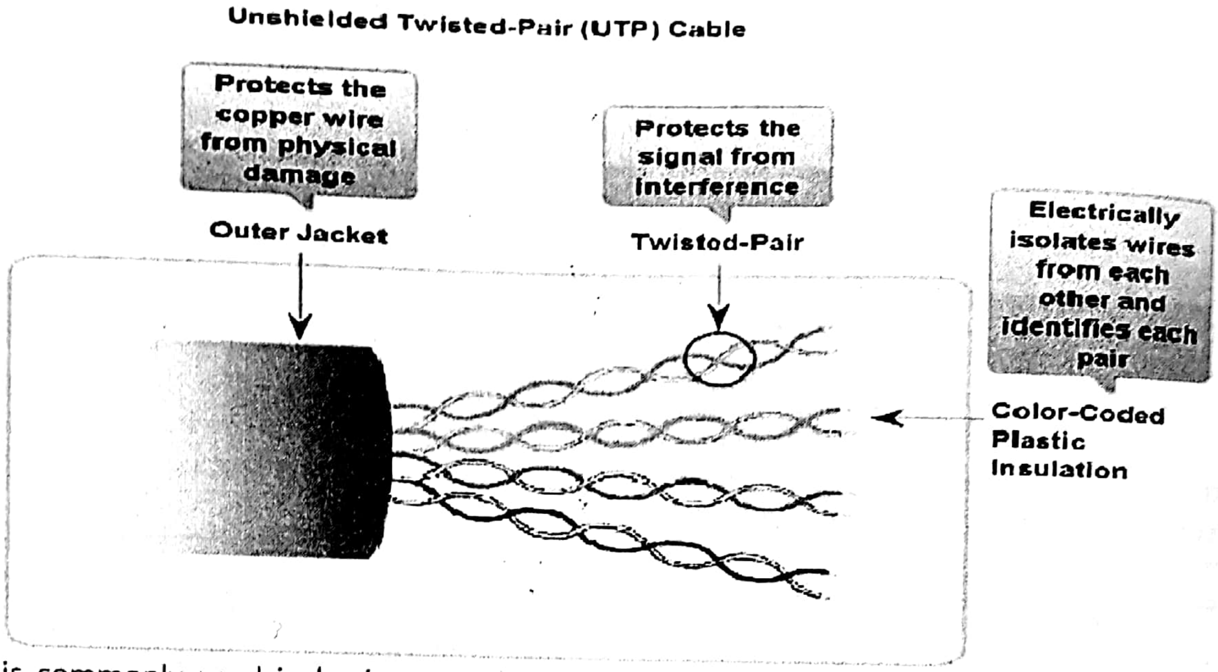
This is usually found in older networks. It is available in various specifications that are rated according to the RG Type system. The distance and cost should be considered when selecting coax cable.

**Coaxial Cable Design**

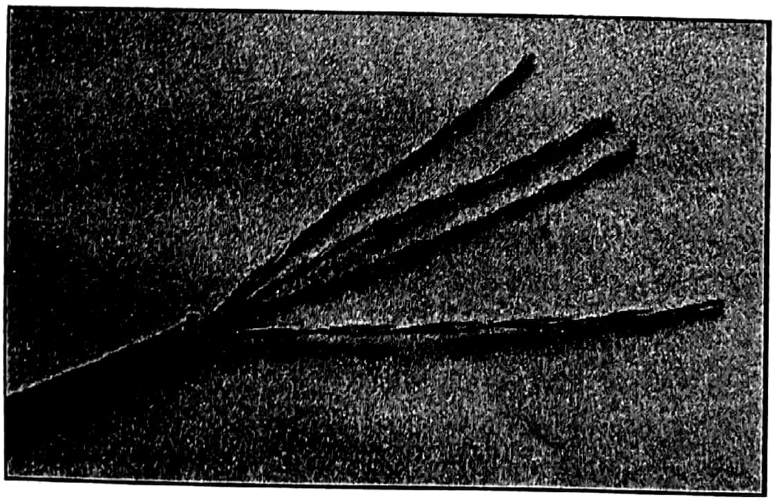


| Advantage | Disadvantage |
|---|---|
| ▪ Less susceptible to EMI interference | ▪ Low bandwidth carrying capacity |

Twisted pair cable

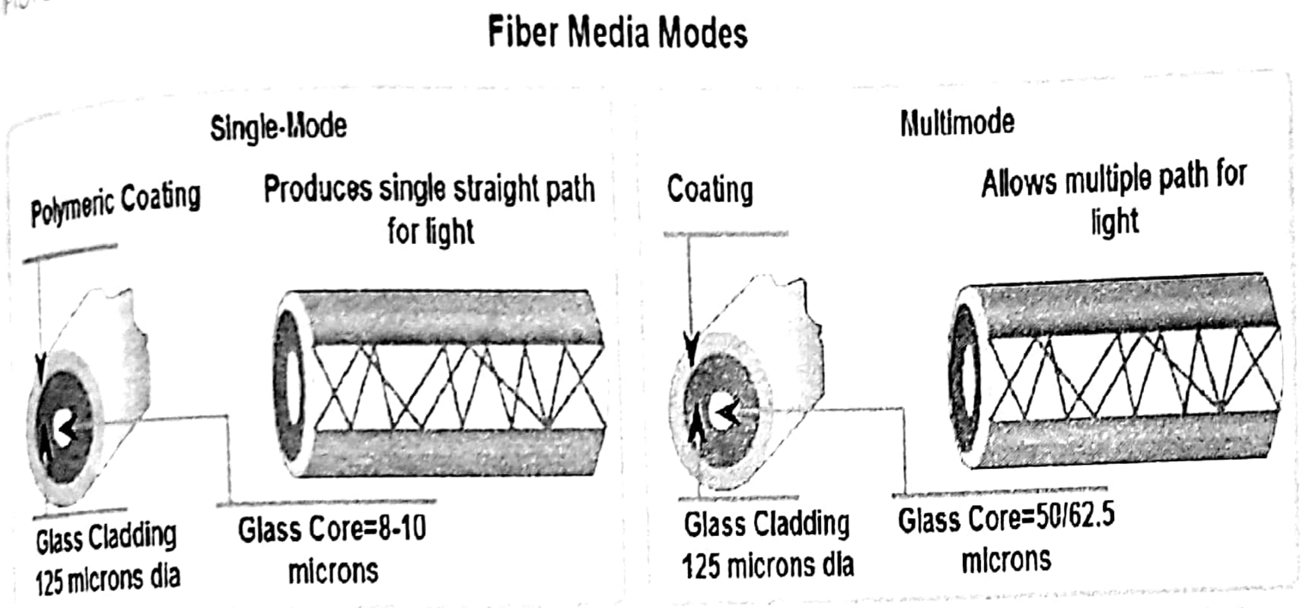

Unshielded Twisted-Pair (UTP) Cable
Protects the copper wire from physical damage
Outer Jacket
Protects the signal from interference
Twisted-Pair
Electrically isolates wires from each other and identifies each pair
Color-Coded Plastic Insulation

It is commonly used in business application to deliver both data and voice. The cable has different category that specifies its maximum length and speed. There are two types which are unshielded twisted pair (UTP) and shielded (STP) which are commonly used. Normally it has four pairs of wires with each pair twisted around d each other to cancel out Electromagnetic Interference.



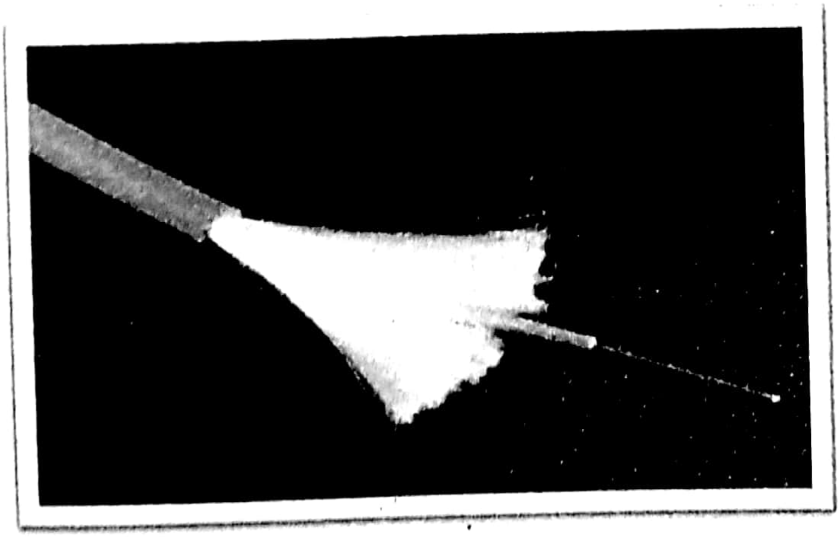
| Advantage | Disadvantage |
|---|---|
| - Cheap compared to other types of cabling<br>- Has limited bandwidth carrying capacity | - Susceptible to interference |

## Fibre Optic


Fiber Media Modes
Single-Mode
Polymeric Coating
Produces single straight path for light
Glass Cladding 125 microns dia
Glass Core=8-10 microns
Multimode
Coating
Allows multiple path for light
Glass Cladding 125 microns dia
Glass Core=50/62.5 microns

This type of cable carries light based data. The actual cable is made up of strands of glass or plastic and is not thicker than a human hair. It has two modes which actually refers to the bundles of light that enter the fibre optic cable:

i. Single mode – is used for wide area networks and other long distance
ii. Multimode – is used for local area networks and short cable runs



| Advantage | Disadvantage |
|---|---|
| ▪ Fastest cable media<br>▪ Less interference and less signal degradation<br>▪ Higher carrying capacity compared to others<br>▪ More secured since impossible to tap light<br>▪ Thinner and lighter weight | ▪ Most expensive out of all cables<br>▪ Cannot be bend around tight corners |

## 11.8 Bit Streaming

***Streaming is defined as the process of delivering media constantly by a provider using telecommunication networks.*** It can be done in two main fashion which are live streaming or on-demand streaming. This technology has emerged due to faster internet connection, better compression algorithm for media delivery and faster and more powerful processors.

### 11.8.1 Live Streaming

Live streaming is defined as the process of delivering media content live over the Internet over a network where subscribers access the content in real time from a media publisher without saving the file to a hard disk at one time only. E.g. YouTube, Football match streaming (ESPN.com)

### 11.8.2 On Demand Streaming

It is defined as the process of delivering media to the user by means of the file being saved to a hard disk and then played from that location and can be available for an extended amount of time. On demand streaming is also known as progressive streaming. E.g. On demand video by MYT service in Mauritius, NETFLIX

### 11.8.3 Bandwidth

Due to the video quality size and file transmission size, a high broadband speed is recommended specially for videos depending on the resolution type. Audio streams are compressed using audio codec such as MP3, ACC or Vorbis. Video streams are compressed using video codec such as VP8 or H.264.

The bandwidth allows smooth streaming of video online at viewable quality. If a large video has resolution 640 * 480 it is usually streamed at 400 -700 kbps even if some have higher bit rate streams since producers have to pay for their bandwidth.

If a video resolution is 320 * 240, it means there are 76800 pixels in each frame while if 640 * 480 is used it has 307200 pixels. Hence compression need to be applied to achieve same data rate which obviously means a reduction in quality. That is why an internet connection of 300 kbps should produce excellent quality at 320 * 240 but would look disastrous at 640 * 480.

# 12.0 Communication and Internet Technologies – IP Addressing

## 12.1 IP Address

***An IP address is a unique address for a device on a network. Every host needs to have a unique IP address. It can be configured manually by a network administrator or automatically via a DHCP server.*** There are two main version of IP address that exist:

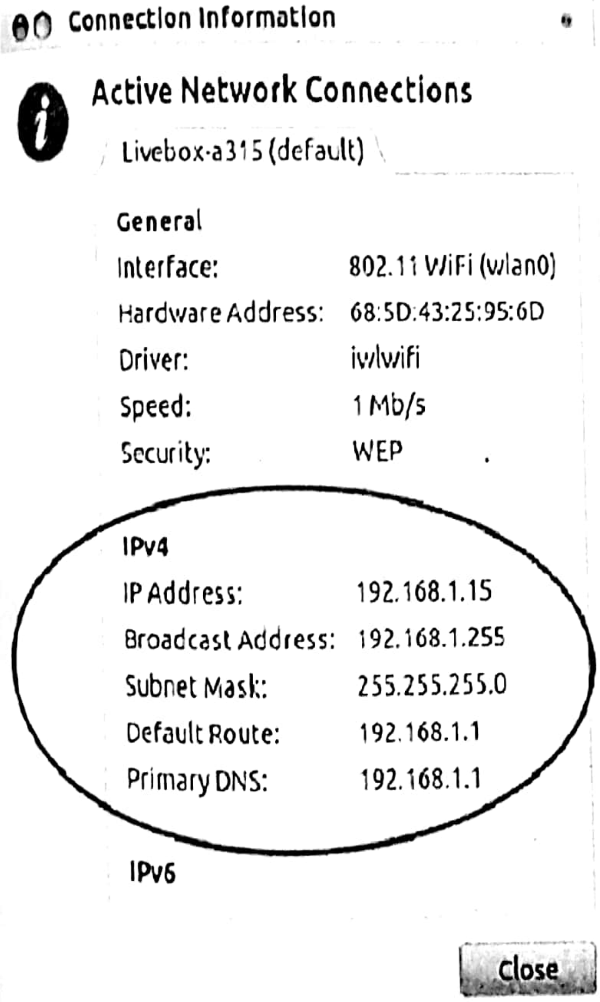
- ✓ IPV4 – 32 bit - quad dotted decimal number between 0-255
- ✓ IPV6 – 128 bit - hexadecimal

### 12.1.1 IPV4

An IPV4 address is made up of a network and Host ID. It uses 8 bits for each octet. Hence an IP address like 192.168.1.9 will be represented in binary as follows:

1100 0000. 1010 1000. 0000 0001. 0000 1001

A subnet mask help identify the network part of an IP address and which part makes up the host.



**Active Network Connections**

Livebox-a315 (default)

**General**

Interface: 802.11 WiFi (wlan0)
Hardware Address: 68:5D:43:25:95:6D
Driver: iwlwifi
Speed: 1 Mb/s
Security: WEP

**IPv4**

IP Address: 192.168.1.15
Broadcast Address: 192.168.1.255
Subnet Mask: 255.255.255.0
Default Route: 192.168.1.1
Primary DNS: 192.168.1.1

**IPv6**

### 12.1.2 Classes of IP Address

| Class | First Octet | Default Subnet Mask | Comments | Possible Number of Host | IP Address Representation |
|---|---|---|---|---|---|
| A | 1-126 | 255.0.0.0 | For very large networks | 16777214 | N.H.H.H |
| B | 128-191 | 255.255.0.0 | For medium size networks | 65534 | N.N.H.H |
| C | 192-223 | 255.255.255.0 | For smaller networks with fewer host | 254 | N.N.N.H |
| D | 224-239 | N/A | Reserved for multicasts | | |
| E | 240-255 | N/A | Reserved for testing | | |

**Note:**

- ✓ N represent the network part and H is the host part.
- ✓ 127 is reserved for localhost and loopback address.

### 12.1.3 Public IP Addresses

A public IP address is needed for any device to be routed on the Internet. The master holder is IANA which assigns it to regional registries and then to ISPs. ISPs allocate them to organisations and individual users. Public IP address is strictly controlled to ensure duplicate addresses are not allowed.

| Class | Public IP Ranges |
|---|---|
| A | 1.0.0.0 – 9.255.255.255 , 11.0.0.0 – 126.255.255.255 |
| B | 128.0.0.0 – 172.15.255.255 , 172.32.0.0 - 192.255.255.255 |
| C | 192.0.0.0 – 192.167.255.255 , 192.169.0.0 – 223.255.255.255 |

### 12.1.4 Private IP Addresses

A private IP address is only intended for use within an organisation and cannot be routed on the internet.

| Class | Private Address Range |
|---|---|
| A | 10.0.0.0 – 10.255.255.255 |
| B | 172.16.0.0 – 172.31.255.255 |
| C | 192.168.0.0 – 192.168.255 |

***NAT (Network Address Translation) is used to translate between private addresses and public address. It is a temporary solution to address the shortage of IP address.*** It can even be used as a security mechanism. Due to the depletion of public IP address, hence IPV6 was introduced.

## 12.2 IPV6

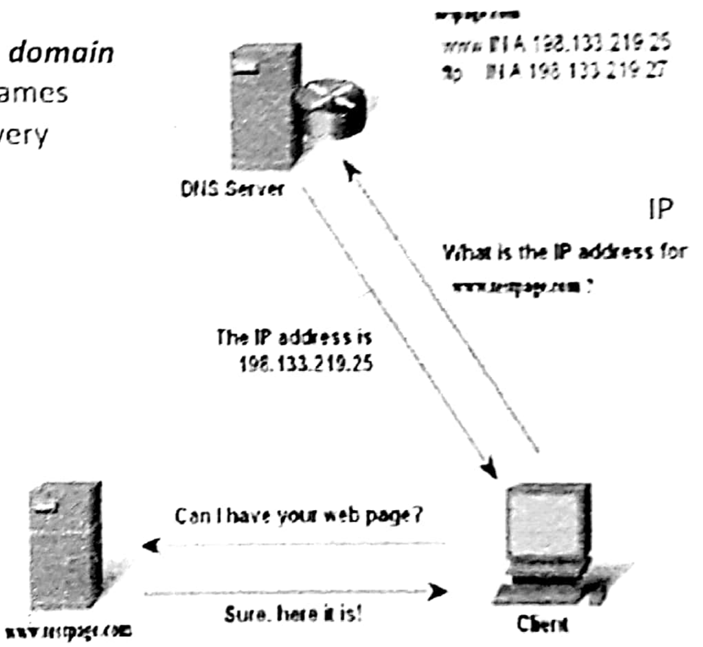
IPV6 uses 128 bits to represent an IP address. It is approximately around 3.7 * $10^{21}$ addresses.
IPv6 Features offers the following features:

- ✓ Increased address space and scalability
- ✓ Simplified configuration
- ✓ Integrated security
- ✓ Compatibility with IPv4

An example of an IPv6 address: 2541:1523:26B1:CC14:0123:1111:2222:4444 Less than 20% of the IPv6 address space has been designated for use currently allowing for enormous substantial growth.

## 12.3 Domain Name Service

***A DNS is an internet service that translates domain names into IP addresses.*** Because domain names are alphabetic, they are easier to remember. Every time someone use a domain name, the DNS service must translate it to the corresponding address. If one DNS server doesn't know how to translate a particular domain name, it asks another one and so on till the correct IP is returned.


www IN A 198.133.219.25
ftp IN A 198.133.219.27
DNS Server
IP
What is the IP address for www.testpage.com ?
The IP address is 198.133.219.25
Can I have your web page?
Sure, here it is!
www.testpage.com
Client

DNS uses a hierarchical system of databases to resolve names and IP addresses of know hosts within networks and across the Internet.

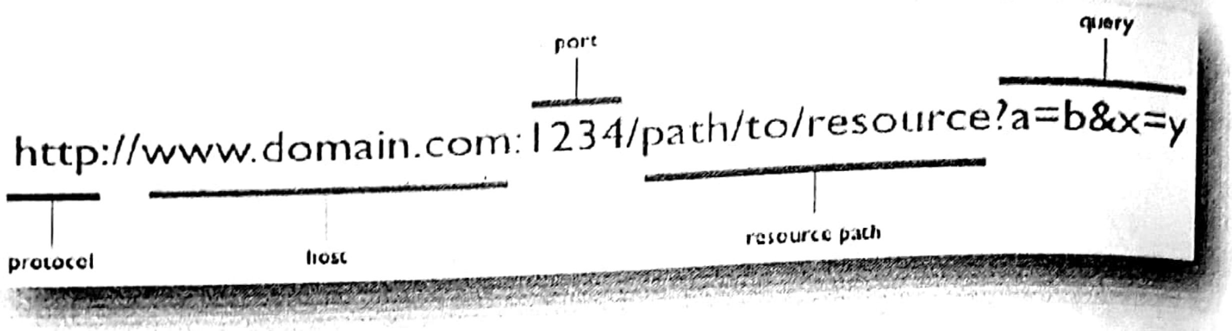
www.google.com = 209.85.225.104

represents a Fully Quantified Domain Name (FQDN) :

- ✓ .com represents the top level domain
- ✓ .google represents a secondary level domain
- ✓ www represents a host computer in the .google.com domain

Other top level domains include .org, .net and .gov. Top level domains can also include country code such as .mu, .za etc.

## 12.4 URL (Uniform Resource Locator)



An URL functions much like a postal or email address. Just as postal and email addresses list a name and specific location, a URL, or web address, indicates where the host computer is located, the location of the website on the host, and the name of the web page and the file type of each document, among other information.

***It is the global address of documents and other resources on the World Wide Web.*** The first part of the address indicates what protocol to use, and the second part specifies the IP address or the domain name where the resource is located. For example http://www.bcs.org.uk

# 13.0 Communication and Internet Technologies – Client and Server Side Scripting

## 13.1 Script

A script is defined simply as a set of instructions which provide change into an application. In terms of web pages, script is used for almost all functions in browsers (client side) and even at the server (server side).

### 13.1.1 Client Side scripting

– writing code used by browsers to render Web pages and to respond to users' interactions. It reduces the overload of the server to validate data as it relies on the user's computer. E.g. of client side scripting language is JavaScript.

### 13.1.2 Server Side Scripting

– writing code used to generate the code consumed by browsers. E.g. of server side scripting language is PHP & ASP.NET. Server side scripting provides user with powerful data manipulation tools and personalisation aspects.

Note: Server side script cannot be seen or copied by the user.

**Process**

1. User clicks on the hyperlink and the current web page is requested from the appropriate server.
2. The server interprets the code in the script found in the page.
3. The page content is then adjusted to what the user has requested by passing data around.
4. The final page with adjustment made is then sent to the user browser for result display.

## 13.2 Sequence of Events for client server interaction

1. User clicks on the hyperlink and the current web page is requested from the appropriate server.
2. The server looks for the web page and finds it.
3. The page found contents' are transmitted to the client computer.
4. The client computer processes the JavaScript code.
5. The page is displayed on the client computer.

## 13.3 JavaScript

JavaScript is a scripting language most often used for client side web development (User's browser). The main reason JavaScript was designed was to add interactivity to HTML pages. The JavaScript code consists of lines of executable computer code and is usually embedded directly into HTML pages.

It has to be noted that JavaScript is an interpreted language and some browsers may behave differently upon a JavaScript program.

## JavaScript uses

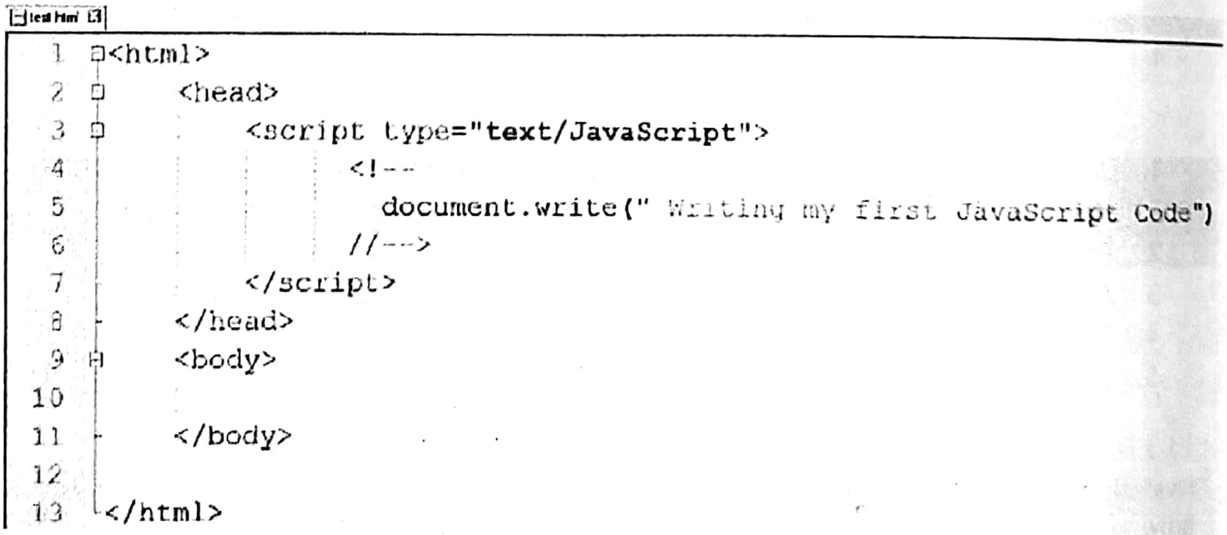
- It provides HTML designers a programming tool to write snippets of code to embed into their HTML pages.
- It allows the insertion of dynamic text into an HTML page.
- It can react to events and allow the running of codes upon clicking on particular sections.
- It can read and write HTML elements contents.
- It can be used to validate data on forms before data is submitted to a server.
- It can be used to detect the visitor's browser.
- It can be used to create cookies to store and retrieve information on the visitor's computer.

## Benefits of JavaScript

- **Usability** – we can modify a page without having to post back to the server which allows faster user interaction.
- **Efficiency** – it can make small, quick changes to the page without waiting for the server
- **Event driven** – it an respond to user actions like clicks and key presses

## JavaScript File

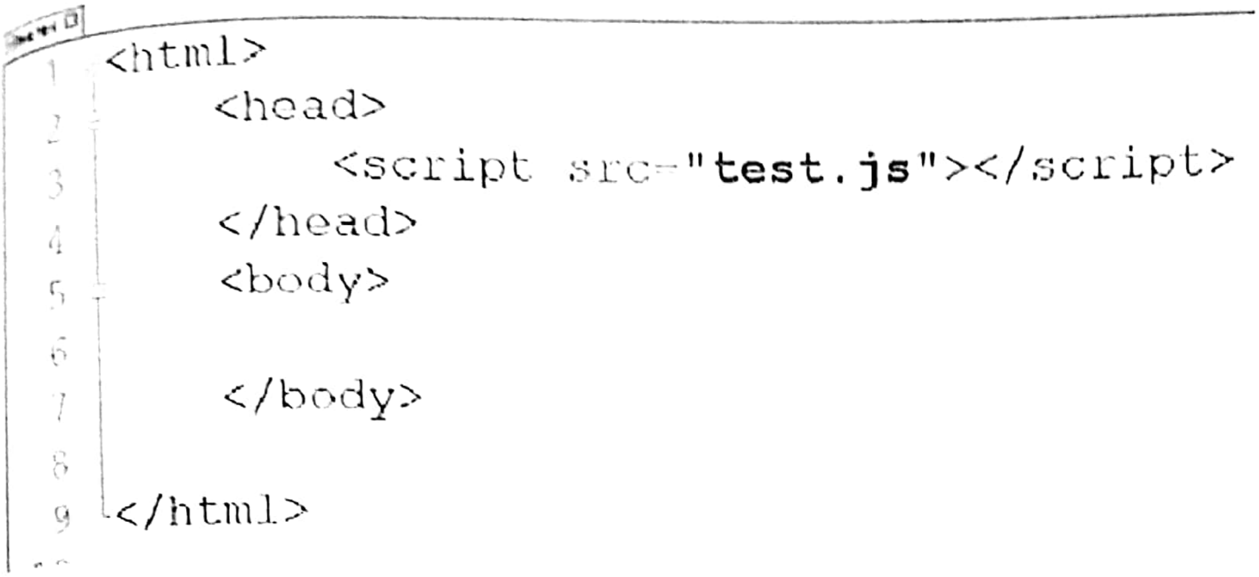
The script should be placed in the HTML page's head and script code is usually stored in a separate .js file. It can however be placed directly in the HTML file's body or head but this is a BAD style as one should try to separate content, presentation and behaviour.



```
<html>
        <head>
              <script type="text/JavaScript">
                     <!--
                        document.write(" Writing my first JavaScript Code")
                     //-->
              </script>
        </head>
        <body>

        </body>

</html>
```

## Using External JavaScript

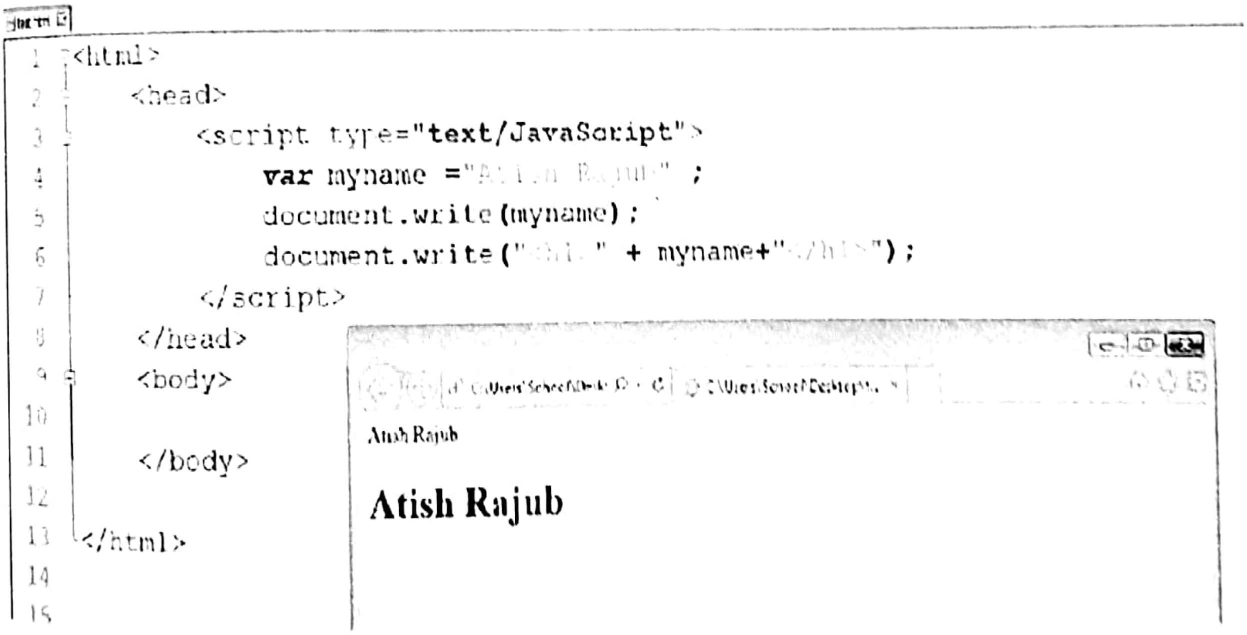
To run same JavaScript on several pages without needing to write the same script on every page, we use an external file and the file extension is .js file.



```
<html>
    <head>
        <script src="test.js"></script>
    </head>
    <body>

    </body>

</html>
```

## JavaScript Variable

Variables are declared in javascript using **var** keyword.



```
<html>
    <head>
        <script type="text/JavaScript">
            var myname ="Atish Rajub" ;
            document.write(myname);
            document.write("<h1>" + myname+"</h1>");
        </script>
    </head>
    <body>

    </body>

</html>
```

Atish Rajub

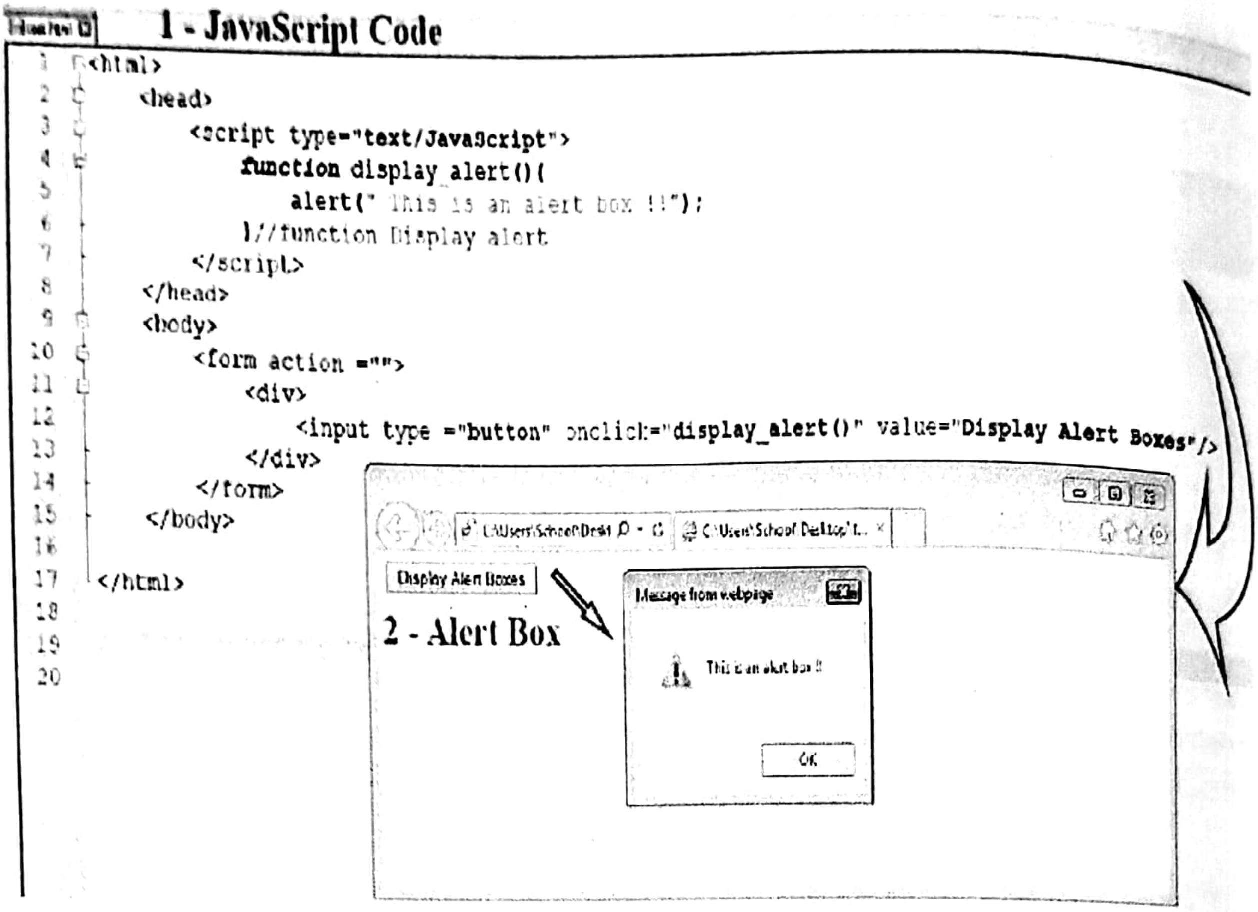
# Atish Rajub

## POPup Boxes

**Difference between Alert, Confirm and Prompt Box.**

**Alert Box** is used to display information to the user and the user has to click ok to proceed.

**Syntax:** alert("Your Text Goes Here");



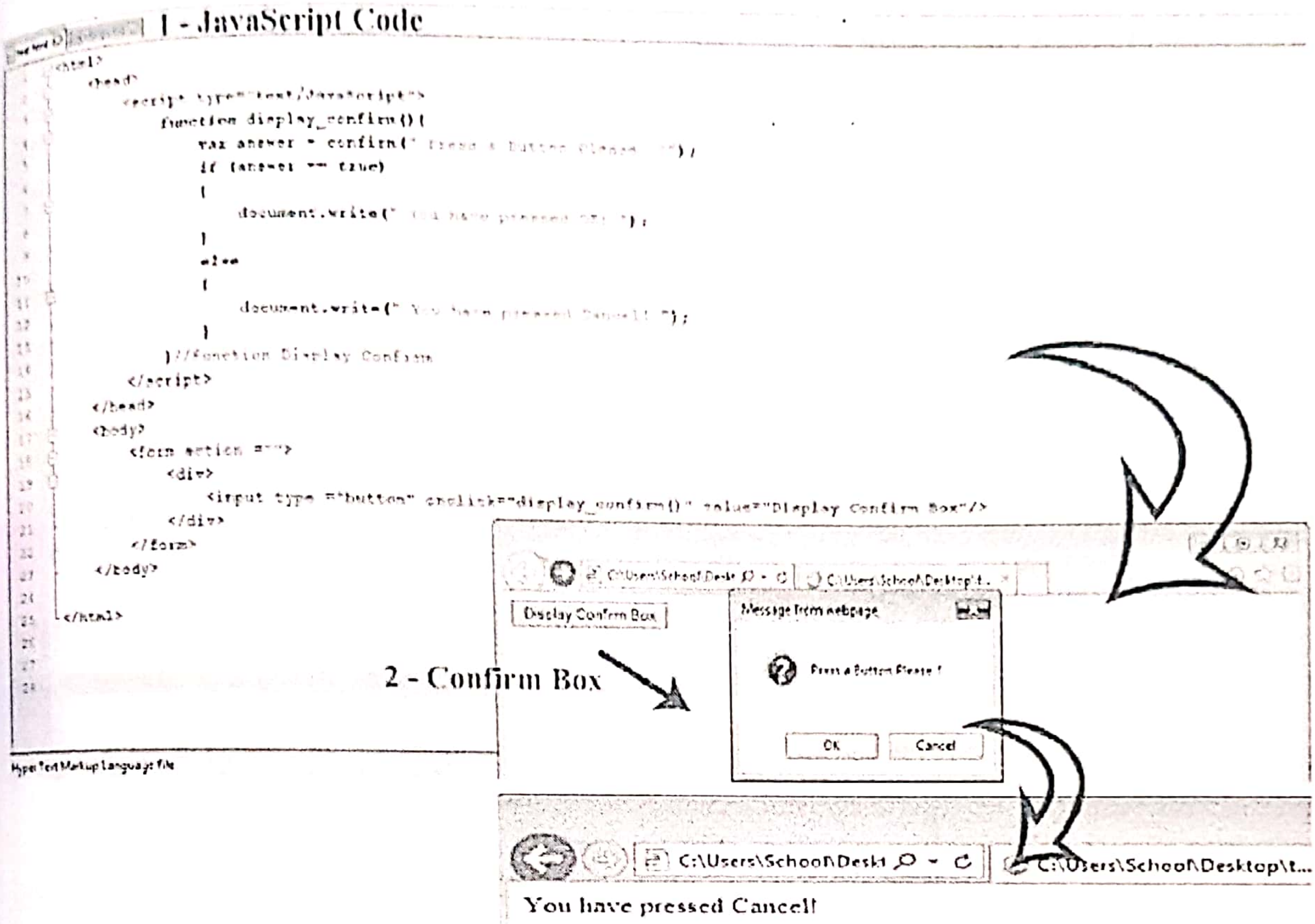
```
<html>
    <head>
        <script type="text/JavaScript">
            function display_alert(){
                alert(" This is an alert box !!");
            }//function Display alert
        </script>
    </head>
    <body>
        <form action ="">
            <div>
                <input type ="button" onclick="display_alert()" value="Display Alert Boxes"/>
            </div>
        </form>
    </body>

</html>
```

This is an alert box !!

## Confirm Box

Confirm Box is used if the user has to verify or accept something. It has two options which are OK or Cancel.

**Syntax:** confirm("Your Text Goes here");



```
<html>
    <head>
        <script type="text/javascript">
            function display_confirm(){
                var answer = confirm(" Press a Button Please ! ");
                if (answer == true)
                {
                    document.write(" You have pressed OK! ");
                }
                else
                {
                    document.write(" You have pressed Cancel! ");
                }
            }//function Display Confirm
        </script>
    </head>
    <body>
        <form action ="">
            <div>
                <input type ="button" onclick="display_confirm()" value="Display Confirm Box"/>
            </div>
        </form>
    </body>
</html>
```

Press a Button Please !

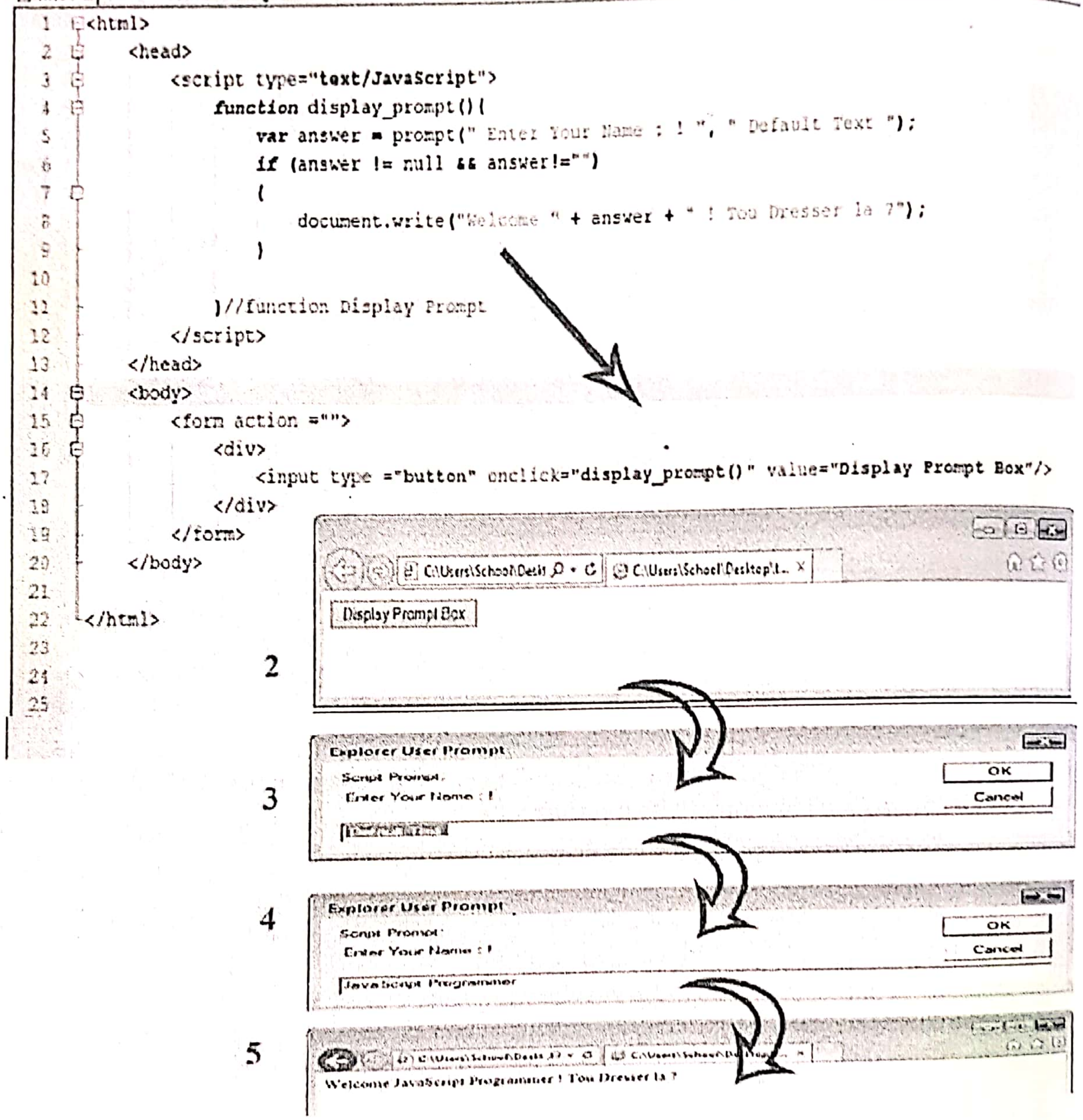
You have pressed Cancel!

## Prompt Box

Prompt Box is used to take input from the user before entering a page for example. If user press OK, the box returns the input otherwise it returns null.

**Syntax :** prompt(" Your Text Goes here" , " Your Default Value");

### 1- JavaScript Code



```
<html>
    <head>
        <script type="text/JavaScript">
            function display_prompt(){
                var answer = prompt(" Enter Your Name : ! ", " Default Text ");
                if (answer != null && answer!="")
                {
                    document.write("Welcome " + answer + " ! You Dresser la ?");
                }

            }//function Display Prompt
        </script>
    </head>
    <body>
        <form action ="">
            <div>
                <input type ="button" onclick="display_prompt()" value="Display Prompt Box"/>
            </div>
        </form>
    </body>

</html>
```

Explorer User Prompt

Script Prompt:

Enter Your Name : !

JavaScript Programmer

Welcome JavaScript Programmer ! You Dresser la ?

## 13.4 PHP (PHP Hypertext Processor)

It is a server side scripting language that is often used to make web pages dynamic. It can perform the following tasks:

- Depending on different context, it can provide distinct contents
- It interfaces with other services such as database, email server and others.
- It also authenticate users.
- It is used heavily in the processing of form information from html pages or PHP dynamic pages.

## PHP Sequence of Events

Scenario 1: Client Browser -> it request an HTML file (often static content) and the server just sends that file.

Scenario 2: Client Browser -> it requests a PHP file (dynamic content) and the server has to read it, process the script code inside it and then sends result across the network. The PHP script produces output as the response sent back.

## Advantages of PHP

- It is free and is open source and hence many developer use it for free of charge.
- It is compatible by most popular web servers such as Apache, IIS, etc.
- It is very simple but support a wide range of functionality
- It is readily available by default on most commercial web hosts.

## PHP File characteristics

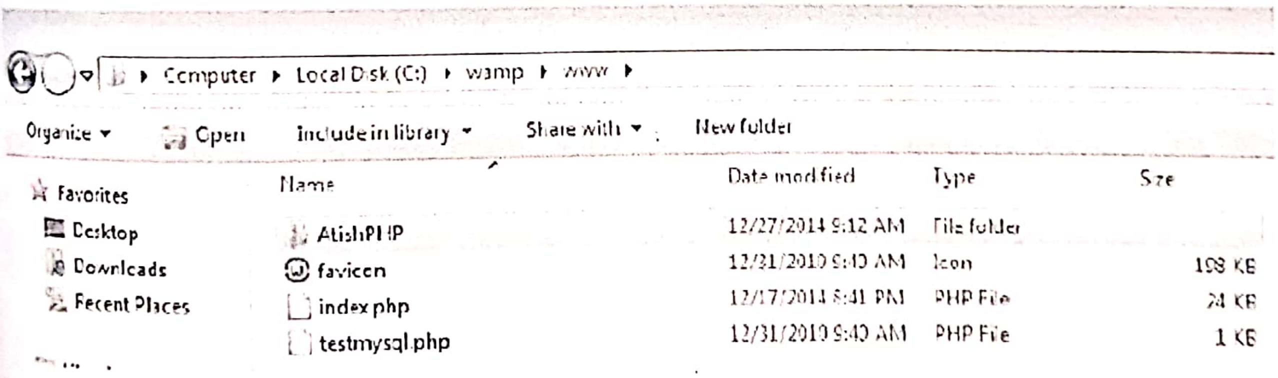
- All PHP page extensions end with .PHP usually.
- A block or file of PHP code begins with **<?php** and ends with **?>**
- PHP statements, codes, function declarations etc. appear between these endpoints.

## Database

Often PHP is combined with a database server (PHP + MYSQL) which is also open source and very popular amidst web developers.

## Steps to run PHP

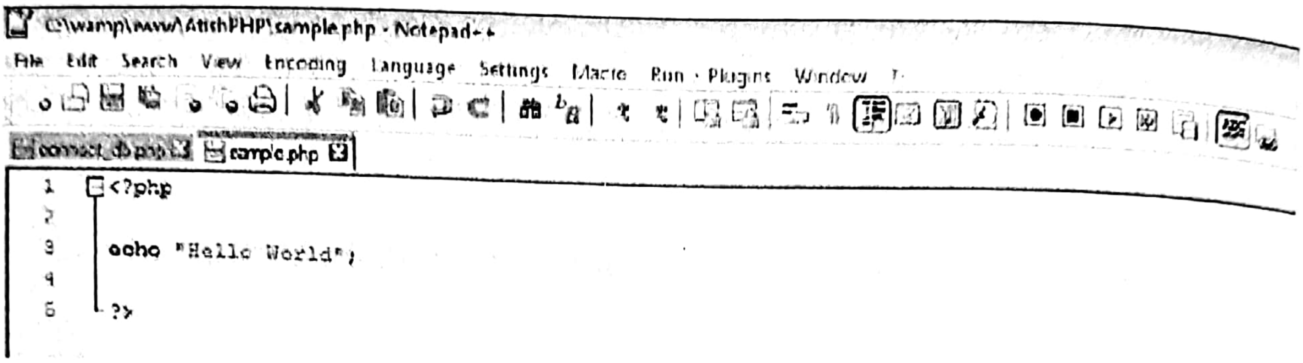
1. Install and Run WAMP server.
2. Go to directory where WAMP is installed and create a folder as shown below.



| Name | Date modified | Type | Size |
|---|---|---|---|
| AtishPHP | 12/27/2014 9:12 AM | File folder | |
| favicon | 12/31/2010 9:40 AM | Icon | 198 KB |
| index.php | 12/17/2014 8:41 PM | PHP File | 24 KB |
| testmysql.php | 12/31/2010 9:40 AM | PHP File | 1 KB |

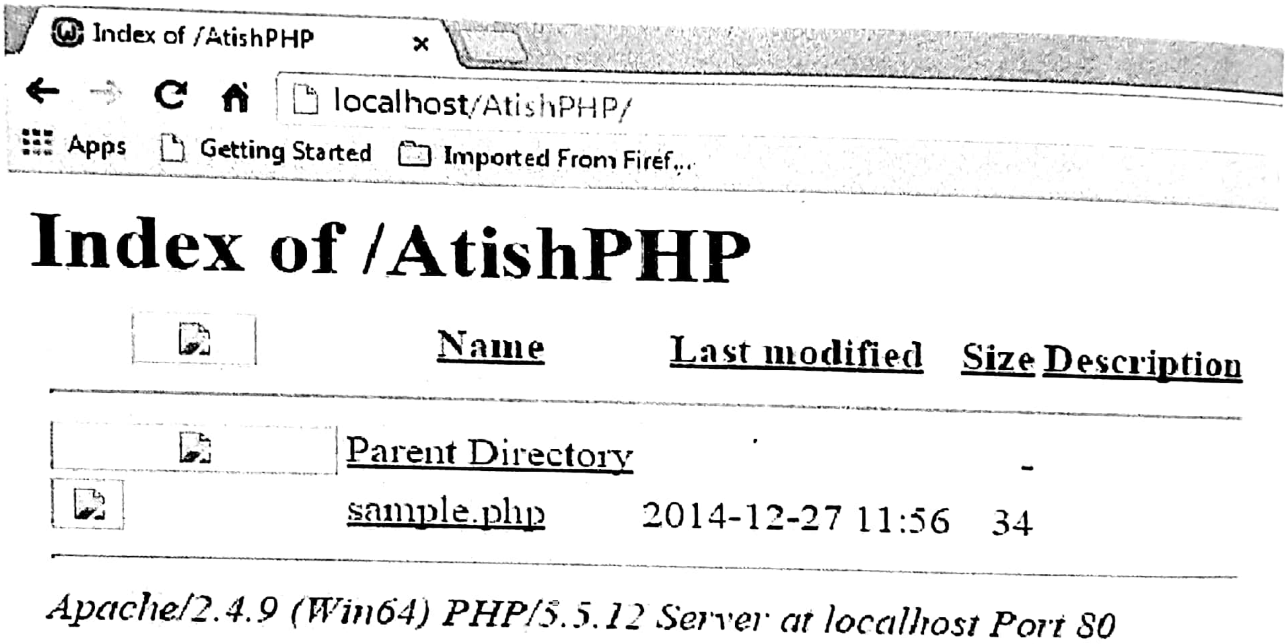
3. Now using a text editor (Notepad, Notepad++) write simple PHP code as shown below and save it as sample.php in the folder created:

Example 1


C:\wamp\www\AtishPHP\sample.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
connect_db.php
sample.php

```
<?php

echo "Hello World";

?>
```

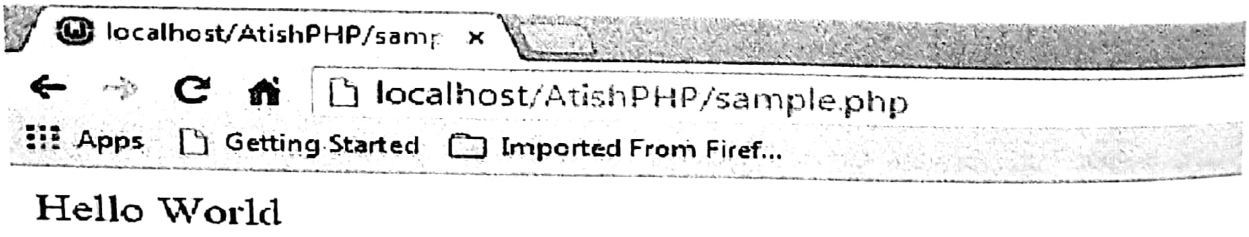
4. Go to your browser and type the url to access the PHP page you have created to see it run.



# Index of /AtishPHP

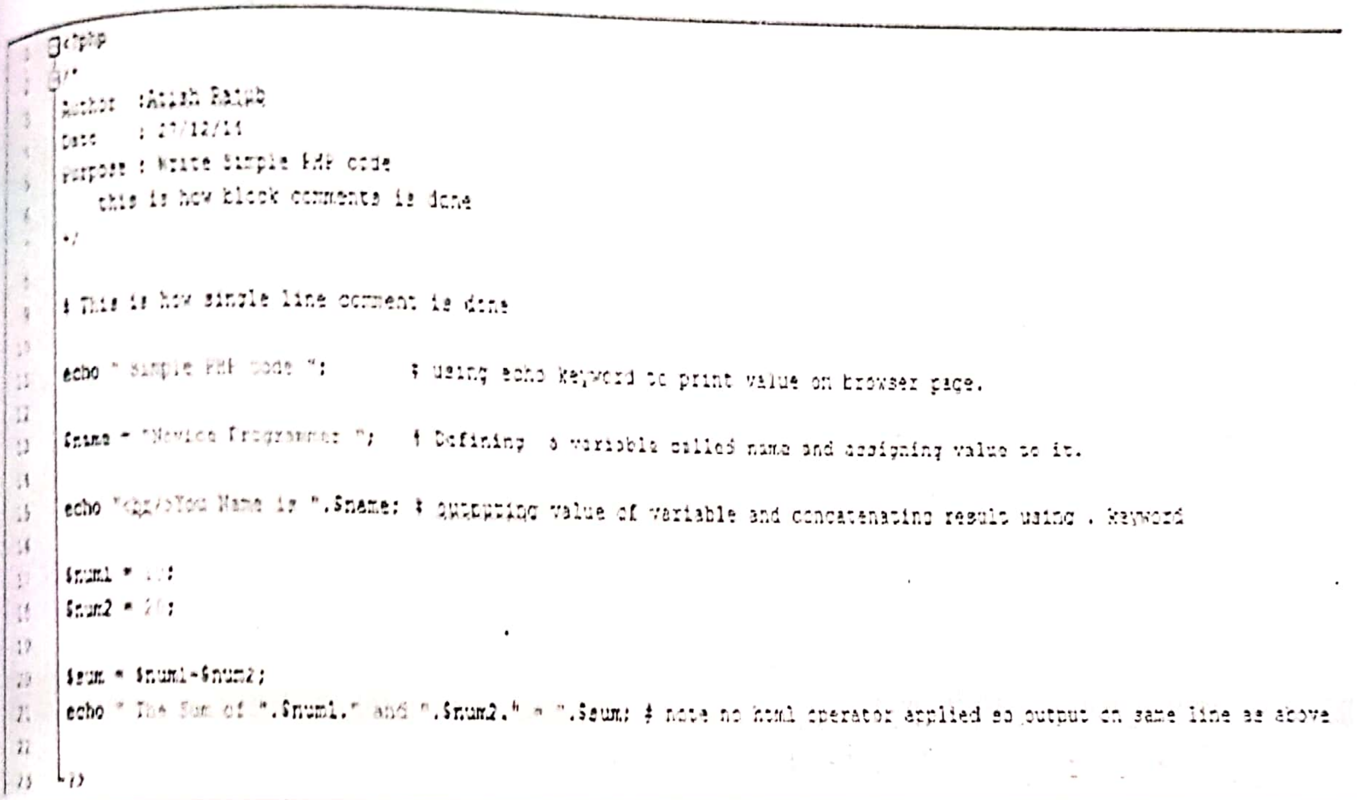
| | Name | Last modified | Size | Description |
|---|---|---|---|---|
| | Parent Directory | | - | |
| | sample.php | 2014-12-27 11:56 | 34 | |

*Apache/2.4.9 (Win64) PHP/5.5.12 Server at localhost Port 80*

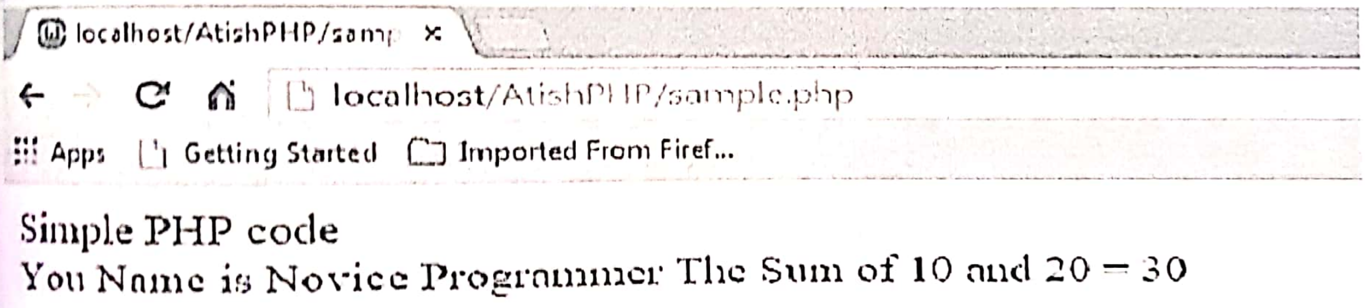


Hello World

## Example 2: Demonstration of Comments, Variables and Concatenation

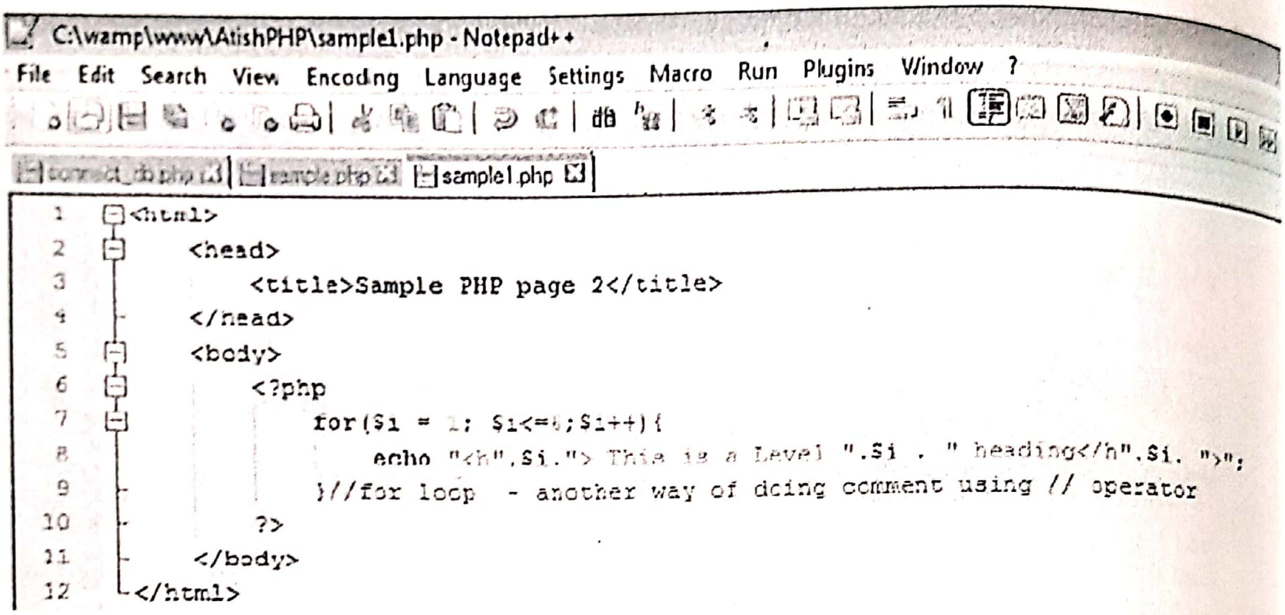


```
<?php
/*
Author  :Atish Raipb
Date    : 27/12/14
Purpose : Write Simple PHP code
   this is how block comments is done
*/

# This is how single line comment is done

echo " Simple PHP code ";        # using echo keyword to print value on browser page.

$name = "Novice Programmer ";    # Defining  a variable called name and assigning value to it.

echo "<br/>You Name is ".$name; # outputting value of variable and concatenating result using . keyword

$num1 = 10;
$num2 = 20;

$sum = $num1+$num2;
echo " The Sum of ".$num1." and ".$num2." = ".$sum; # note no html operator applied so output on same line as above

?>
```

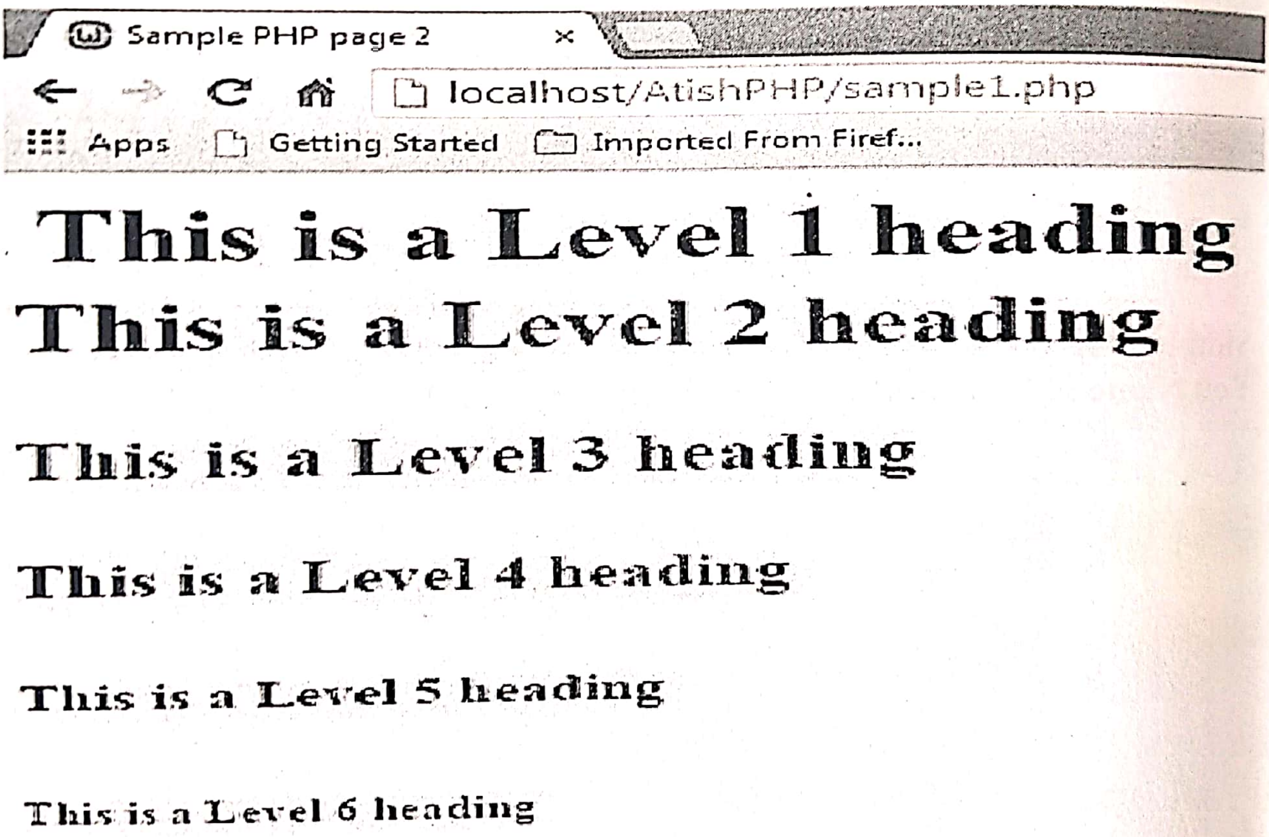


```
Simple PHP code
You Name is Novice Programmer The Sum of 10 and 20 = 30
```

Example 3 –Demonstration of Mixed HTML elements, Dynamic PHP and Loops


C:\wamp\www\AtishPHP\sample1.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
connect_db.php
sample.php
sample1.php

```
<html>
	<head>
		<title>Sample PHP page 2</title>
	</head>
	<body>
		<?php
			for($i = 1; $i<=6;$i++){
				echo "<h".$i."> This is a Level ".$i . " heading</h".$i. ">";
			}//for loop  - another way of doing comment using // operator
		?>
	</body>
</html>
```



This is a Level 1 heading

This is a Level 2 heading

This is a Level 3 heading

This is a Level 4 heading

This is a Level 5 heading

This is a Level 6 heading

Example 4

Connecting To database server, retrieving and displaying data.

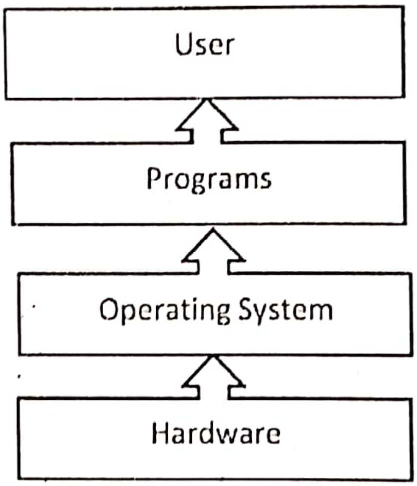
```
<?php
        $db_host = "localhost";                    # server IP or Name
        $db_user = "root";                         # username to access database server
        $db_pass = "";                             # password
        $db_name = "computerscience";              # database we want to use

        # trying to connect to database using the credential above
        $db_connect = mysql_connect("$db_host","$db_user","$db_pass") or die
(mysql_error());

        # selecting database we want to use
        $db_select = mysql_select_db ("$db_name") or die (mysql_error());

        # running a query against the database and storing the data in variable result
        $result = mysql_query("select * from customer") or die (mysql_error());
?>
<html>
        <head>
                <title>Sample Database Connection </title>
        </head>
        <body>
                <p>Using PHP to Connect to Server and Displaying the result from table
customer into a table </p>
                <table cellpadding="0" cellspacing="0" border="1">
                        <tr bgcolor ="red">
                                <th>ID</th><th>First Name</th><th>Last
Name</th><th>Address</th><th>Telephone</th><th>Date Of Birth</th>
                        </tr>
                        <?php
                                while($row = mysql_fetch_array($result) ){              #
converting the result into an array
                                        echo "<tr>";
                                                echo "<td>".$row["C_ID"]."</td>";
                                                echo
"<td>".$row["CFname"]."</td>";
                                                echo
"<td>".$row["CLname"]."</td>";
                                                echo
"<td>".$row["CAddress"]."</td>";
                                                echo "<td>".$row["CTel"]."</td>";
                                                echo "<td>".$row["CDob"]."</td>";
                                        echo "</tr>";
```

```
                                    }//while
                        ?>
                </table>
        </body>
</html>
```

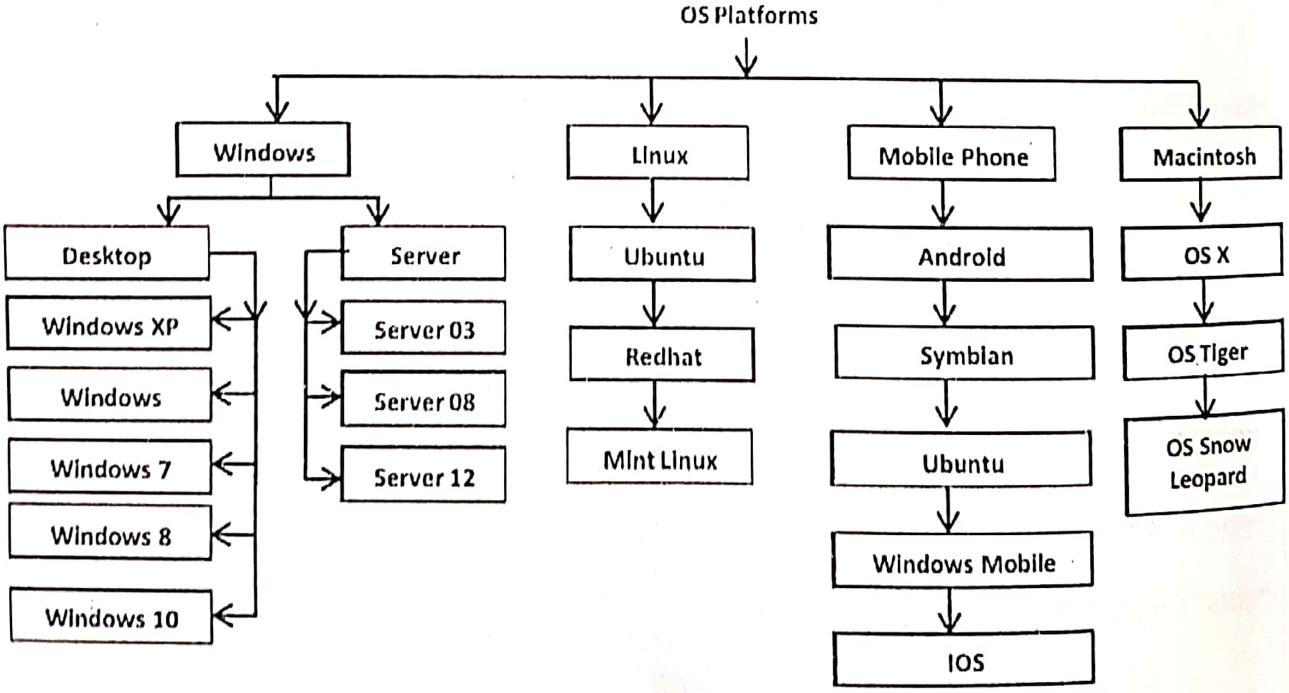
# 14.0 System Software – Operating System

## 14.1 Operating System

An operating system (OS) is a set of programs that manage computer hardware resources and provide common services for application software. The operating system is the most important type of system software in a computer system.



## 14.2 OS Platforms



## 14.3 Features of operating systems

The operating system (OS) must provide:

- ✓ Management of all hardware resources in terms of processor, secondary storage filing system and input/output devices.
- ✓ An interface between the user and the machine
- ✓ An interface between applications software and the machine
- ✓ Security for the data on the system

Utility software to allow maintenance to be done.

1. The main aims and features of an operating system are to manage the resources of the computer system:
   - ✓ **Processor management** – for multiprogramming, the low-level scheduler must decide which job will get the next use of the processor
   - ✓ **decide on appropriate scheduling algorithms**
   - ✓ **file management** – maintaining a list of files, directories and which file allocation units belong to which files
   - ✓ **input/output management** – control of all input and output devices attached to the computer
   - ✓ **Memory management** – using strategies such as segmentation and paging and the high-level scheduler to decide which jobs will be loaded next.
2. The OS provides a means of communication (the user interface) between the human user, or the outside world, and the computer.
3. The OS provides a platform on which the applications software can run.
4. The OS hides the complexity of the hardware from the user.
5. The OS controls access to the computer system so that you can put passwords on your files to stop other people seeing them. It controls the access rights of different users

## 14.4 Types of OS

### Single User Multi Tasking OS

In such system , only one user can use it at any one time.It recognises the user and user rights. However, it is able to give impression that more than one thing can be done at a time . For example , more than 1 application may be open at the same time.E.g typing a word document while checking email on the Internet.

### Batch Operating System

It is a system where the required data is collected together before processing all in one go.It is usually used for processing a set of similar items and this happens at a quiet time.Batch system does not require human interaction but requires a Job Control Language to control the process.E.g Payroll , Utility Billing , Cheque clearance at banks etc .Such applications need batch application since pay or bills cannot be calculated unless all the data about the work done is collected and where results are not time sensitive.

## Real Time Operating System

A real time operating system is defined as a system whereby the current output affects the next input. There is immediate response to the user and files which are updated immediately. E.g. Transaction Processing Systems (Airline Booking, any booking system). Other examples are CONTINUOUS MANUFACTURING PROCESS /MONITORING (any control/monitoring application). These applications are real time since there is need for a response in a reasonable time / immediate

## Real time Transaction Processing System

It is an online system that processes individual, whole transactions as they occur, e.g. an airline booking system or automated stock control system, where the maximum response time needs to be in the range of seconds – nowadays users tend not to tolerate delays of tens of seconds or even whole minutes

## Real Time Process Control

This use of the term real-time differs from its use in a real-time process control system, which continuously monitors and processes sensor data sufficiently rapidly to produce the output required to keep pace with the user's needs for information or control – although control of a greenhouse would probably tolerate a delay of a few seconds or even a few minutes, in most cases, this means a maximum response time between microseconds and seconds.

## Network Systems

A network system is one in which processing occurs independently in more than one location, but with shared and controlled access to some common resources.

## Automated Systems

Automated systems are broadly similar to control systems, but are dedicated to a particular task and lack the ability to collect and analyse data and the flexibility to allow for and act on user interaction beyond a very simple level.

**Single User OS** -A single-user operating system enables only one user at a time to access the system.

**Multi-User OS** - A multi-user operating system enables more than one user to access the system at the same time

## 14.5 System Software – Utility Programs

### Utility Programs

Utility software is a type of system software designed to help analyse, configure, optimize and maintain the computer. It performs a specific task in the OS. Examples of utility software include:

- Virus scanner - is used to protect your system from viruses by monitoring input and stored data to ensure that it does not contain malicious software
- Disk defragmenter - is used to speed up your hard disk by rearranging files on HDD which have been spread across the hard disk
- System monitor - is used to look at your current system resources and their usage with regards to performance
- File managers - are used to add, delete, rename and move files and folders across the OS.
- Disk Formatter - prepares the disk (HDD, Other Media) to receive data by dividing the later into smaller areas, each of which can be searched more easily for the OS use.
- Backup - makes copies of critical OS files and user data for use in case of data loss

### File Compression Software

- Reduce file size by compression space so as to transmit over internet or to save space on a secondary storage device by cutting out much of duplication of data in the file.
- These files include sound and video - > e.g. WinRAR, WinZip etc.

File Compression Software

- Reduce file size by compression space so as to transmit over internet or to save space on a secondary storage device by cutting out much of duplication of data in the file.
- These files include sound and video - > e.g. WinRAR, WinZip etc.

## 14.6 System Software – Library Programs

### Library programs

Library programs are collections of resources used to develop software. These include pre-written code and subroutines. Library programs contain code and data that provide services to other programs such as interface (look and feel), printing, network code and even the graphic engines of computer games. Most programming languages have a standard set of libraries that can be used, offering code to handle input or output, graphics and specialist maths functions.

You can also create your own custom libraries and when you start to write lots of programs with similar functionality you'll find them very useful. Below is an example of how you might import libraries into VB.NET:

```
'imports the libraries allowing a program to send emails

Imports System.Net.Mail

'imports the libraries allowing a program to draw in 2D

Imports System.Drawing.Drawing2D

'imports the Math Library

Imports System.Math
```

### DLL (Dynamic Link Libraries)

It is normally a library of codes that can shared among various applications under the windows platform. It usually contains a set routines and variables. The DLL in windows platform are of re-locatable and non-re-locatable type.

| Advantage | Disadvantage |
|---|---|
| ▪ DLL can be altered to improve them without the need for recompiling or relinking the applications depending on them. | ▪ It is usually slow to call DLL routines because a lot indirect call is required. |

# 15.0 System Software – Language Translators

## 15.1 High- and Low-level languages and their Translators

Computer Programming Languages can be grouped into two main types: Low Level Language (LLL) and High Level Language. Each one of the language category have their own advantages and drawbacks.

### 15.1.1 Low Level Languages (LLL)

Low Level Languages are programming languages which are closer to machines. These include Machine Code and Assembly Language. Machine code uses series of binary codes i.e. 0 and 1 while assembly language uses a set of mnemonics.

### 15.1.2 High Level Languages (HLL)

High Level Languages are close to human language and have been developed for specific applications. Usually it uses English like keywords. Example of HLL includes JAVA, VISUAL BASIC, FORTRAN, and COBOL and so on. HLL is much easier for humans to program in HLL but as computers only understand machine code programs written in HLLs need to be translated into machine code before they can be executed.

### 15.1.3 Contrast between HLL and LLL

| High Level Language | Low Level Language |
|---|---|
| ▪ Programs are easy to read and write<br>▪ Programs are portable to distinct types of Machines<br>▪ Programs are easy to test, debug or maintain<br>▪ Programs codes are problem oriented | ▪ Programs are relatively difficult to read and write<br>▪ Programs are machine are not portable<br>▪ Programs are difficult to debug, maintain and test<br>▪ Programs are machine oriented |

## 15.2 Types of Translators

Computers only understand binary. HLL is close to human language. There are 3 main types of translators which are Compiler, Interpreter and Assembler.

### Machine Code

Computers work and understand only machine code .Simple instructions are represented by a binary pattern in the computer. Programming in machine code takes a long time and is prone to errors. However, machine code run faster since there is no need for translation. E.g. 0101 1100

### Assembly Languages

It was developed to improve program writing and is designed specifically for a family of processor. It makes use of mnemonics and names for locations in memory. E.g. LDA B means

Load Content of the Memory location whose address is represented by the variable B into the accumulator.

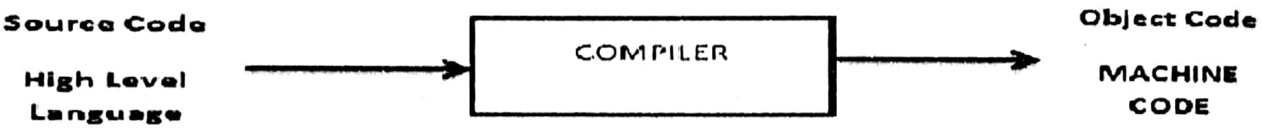
Each assembly instruction represents a single machine instruction which means that it is fairly easy to translate a program written in assembly language to machine code. Assemblers which are loaded into the computer translate the assembly language to machine code. Writing programs in assembly language, although easier than using machine code, is still tedious and takes a long time.

## 15.3 Interpreters & Compilers

Many high-level languages use both. Programmers use the interpreter during program development and, when the program is fully working, use a compiler to translate it into machine code. This machine code version can then be distributed to users who do not have access to the original code.

### Compiler

A compiler translate high-level languages into machine code at one go. The machine code version can be loaded into the machine and run without any further help as it is complete in itself. The high-level language version of the program is called the source code and the resulting machine code program is called the object code.



**Disadvantages of Compilers**

- ✓ Use a lot of computer resources.
- ✓ Has to be loaded in the computer's memory at the same time as the source code and there has to be sufficient memory to hold the object code.
- ✓ Has to be sufficient memory for working storage while the translation is taking place.
- ✓ Errors in the original program are difficult to pin-point.
- ✓ Used to compile: Final release of software such as Games etc.

### Interpreters

Interpreters take each instruction in turn and translates it into machine code. It executes the translated instruction before the next instruction is translated. (Line by Line). It is usually used for line by line debugging

**Advantages**

- ✓ Need less memory than compilers (useful in early computers which had limited power and memory).
- ✓ Continual compilation of whole code is wasteful / time consuming during testing particularly if very minor changes have been made.

- ✓ During testing translator diagnostics will be more complete as error messages will be produced in relation to the HLL being used and not the machine code.
- ✓ As the error messages when the error is produced is on the line it is encountered it is easier to identify / isolate the instruction causing the problem.
- ✓ Individual segments can be run without needing compile the whole program.

**Disadvantages:**

- ✓ Slow execution compared to that of a compiled program because:
  - ○ The original program has to be translated every time it is executed.
  - ○ Instructions inside a loop have to be translated each time the loop is entered.

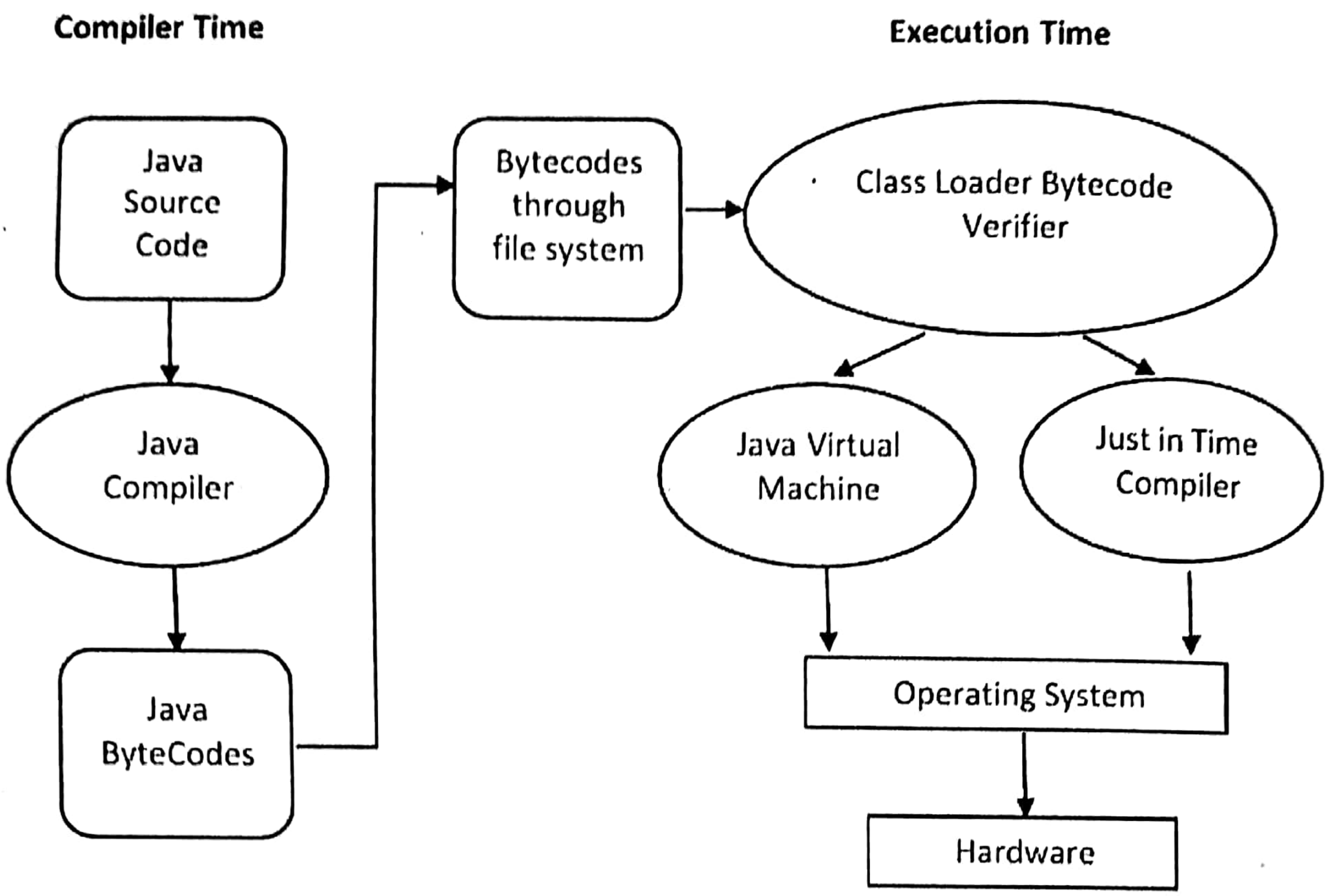
```
For count = 1 to 99999 do
    Print count
End for
```

Difference between Compiler V/S Interpreters

| Interpreter | Compiler |
|---|---|
| ▪ Interpreter translates line of code and then runs it.<br>▪ Interpreter retains source code.<br>▪ Interpreter must be present for run. | ▪ Compiler translates entire program before run.<br>▪ Compiler creates an object code.<br>▪ Compiler must be present for translation. |

## 15.4 Hybrid Language

There are some language such as Java which is partly compiled and partly interpreted. The source code is compiled into a program that runs inside a JVM (Java Virtual Machine) which allows the java code to run on any hardware running the JVM which overcomes one of the drawbacks of the compiler. The JVM is an abstract computing machine that allows hardware and Operating system independence


Compiler Time
Execution Time
Java Source Code
Java Compiler
Java ByteCodes
Bytecodes through file system
Class Loader Bytecode Verifier
Java Virtual Machine
Just in Time Compiler
Operating System
Hardware

# 16.0 Security, Privacy and Data Integrity – Data Security

## 16.1 Data Security

Data Security is defined as the safeguard of data to maintain its privacy and integrity along with protection from accidental damage, corruption or human errors. Often online systems need to be secured from attacks including denial of service attacks, phishing, pharming and others. Such online system includes online shopping, online banking, online trading applications etc.

## 16.2 Data Privacy (Confidentiality)

It refers to the process of preventing unauthorised access to data, deliberate disclosure through malicious acts or even accidental disclosure due to flaws in procedures in place.

## 16.3 Data Safeguard methods for standalone and online systems

- DRM – Digital Rights Management
- Firewalls both software and hardware along with proxy servers
- Updated Antivirus with regular scan
- Regular backups
- Use of password using keyboard or biometric devices (Finger Print, Retina Scan)
- Use of Secure Socket Layer (SSL) for data transmission
- Use of symmetric encryption
- Use of Digital Certificates
- Screening of cables to avoid wire tapping

### Antivirus Software

***Software used to detect and remove viruses from a computer system.*** Once the software is installed on the computer, it monitors all traffic on the system, then alerting users on potential infection. Regular updates of the software are required to ensure that new viruses are detected. E.g. Kaspersky, Norton Internet Security

### File Backup

Files backup is described as a security measure where a copy of files is made and kept in a different location. In case the original files get lost or damaged, then the backup copies are used to recover/restore computer operations. Mediums that can be used for backup includes tapes and disks, carefully explaining that these devices must be removable and not part of the actual system from which backup is being made.

## Firewall

***A combination of hardware, software and security mechanisms to restrict the flow of information in and out of a computer system. System designed to prevent unauthorized access to or from a private network.*** Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. E.g. Cisco PIX, Barracuda.

## Cryptography

It is the conversion of data into a scrambled code that is decrypted and sent across a private or public network. Cryptography is used usually to protect email messages, credit card information and corporate data during transmission. Ciphers are algorithms used to encrypt or decrypt data. Used often in online shopping sites, online banking sites

## 16.4 Mirroring and Backup Disaster Recovery Strategies

### 16.4.1 Disk Mirroring Strategy

This is a technique used to protect a computer system from data loss or even potential losses due to hard disk failures. Disk mirroring technique can be implemented using either a hardware based or software based system.

### 16.4.2 Hardware based disk mirroring

This type of system is implemented using RAID controllers installed in the system using several separate hard disk drives. Each hard disks appears as distinct volumes and the same data sector is identically written to all volumes. There is only a mild system degradation at the expense of fault tolerance. There are different types of RAID (Redundant Array of Independent/ inexpensive disks) levels used.

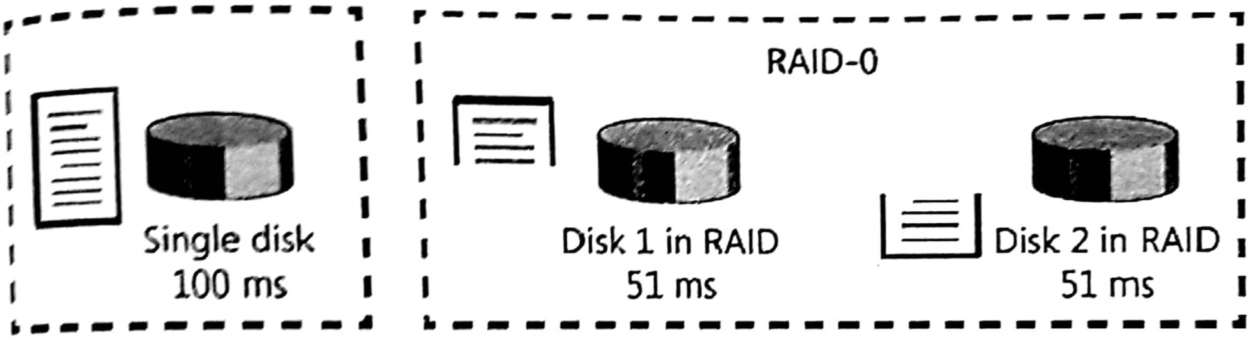
RAID uses multiple disks to provide increased performance and fault tolerance. Fault tolerance refers to any system that can suffer a fault but can still tolerate the error and continue to operate. With RAID in place, a disk can fail but the system will continue to operate.

The different types of RAID are:

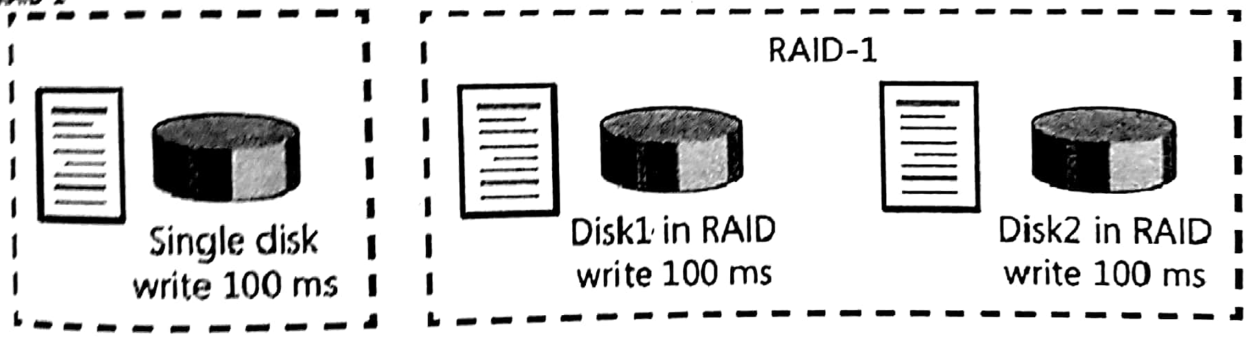
- ✓ RAID-0
- ✓ RAID-1
- ✓ RAID-5
- ✓ RAID-10

| RAID | Fault Tolerance | Minimum number of Disks | Comments |
|---|---|---|---|
| RAID-0 | NO | 2 | Use for performance only |
| RAID-1 | YES | 2 | Duplexing adds a second controller |
| RAID-5 | YES | 3 | Writes are slower than RAID-0 due to parity calculations |
| RAID-10 | YES | 4 | Best and most expensive |

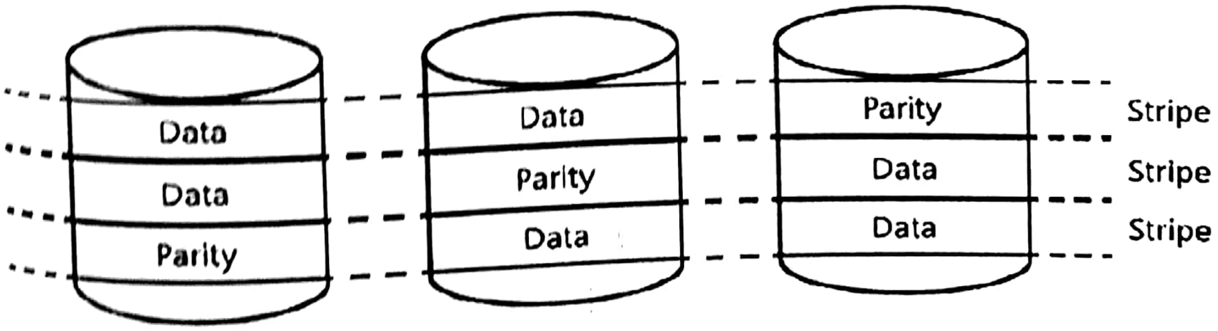
**RAID-0**



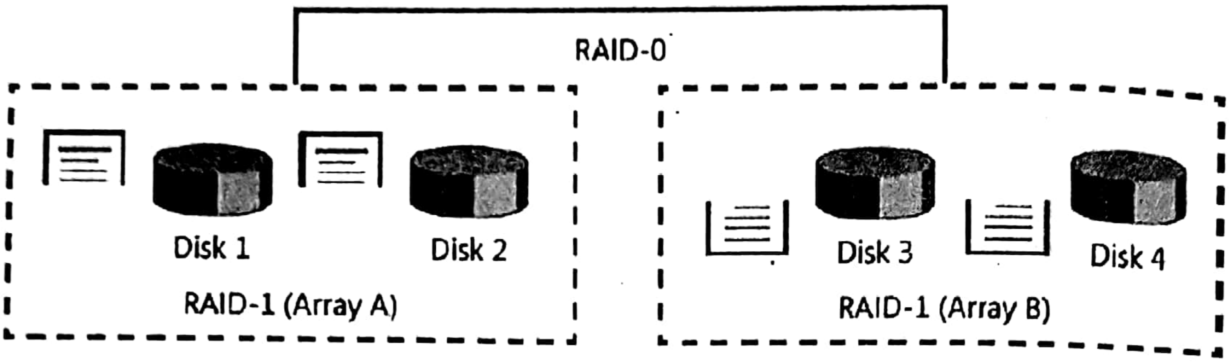
**RAID-1**



**RAID-5**



**RAID-10**



## 16.5 Software based disk mirroring

This type of system requires special type of software to be installed in the system and is usually the less expensive and more flexible. However, this type of mirroring results in system degradation and is more prone to incompatibilities like boot time. For e.g., in Windows-based systems, you can use dynamic disks and create software based RAID arrays with the biggest benefits that it doesn't cost anything.

**1. On-site backups**

Normally such system use dedicated backup servers to store periodic backups. Even a shared infrastructure can be used for agility and flexibility. E.g. of such system: Parallels Plesk, cPanel, Idera/R1Soft, WHM

**2. Off-site backups**

These system use servers stored online off site which often use cloud technologies. These are particularly useful in case of catastrophic failure.

## 16.6 Authorisation and Authentication

Authorisation and authentication are processes designed to protect the computer system and data.

### Authentication

Authentication is defined as the process of verifying that someone is who they claim they are. Technique used for authentication can be implemented by:

- ✓ The use of passwords
- ✓ The use of smartcards / Biometric Devices (Finger Print, Retina Scan)
- ✓ Implemented by the use of digital signatures in online communications

### Authorisation

Authorisation is defined as the process of establishing if the user is permitted to have access to a resource. Hence, ensure that confidential information only reaches the intended recipient. E.g. granting permission to a user for access to a data file or a directory. Authorisation technique include:

- ✓ Responses to special questions
- ✓ Provision of memorable data

### Access Control

Access control is defined as the process of enforcing the required security for a particular resource. It can be seen as the combination of both authentication and authorisation with additional measures. E.g. in web or other software , access control can be applied using file permissions, URL access lists, bespoke logic and other security features of available framework.

### Data Integrity

It refers to the process of maintaining data reliability against corruption from software, hardware, errors, viruses, hacking and data transmission

**Factors that causes data to be unsafe**

- Malicious actions from viruses
- Unauthorised viewing, alteration or corruption of data
- Hardware failures or software failures
- Hacking

**Error Detection**

Error detection is defined as the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver.

**Error Correction**

Error correction is defined as the detection of errors and reconstruction of the original, error-free data.

## 16.7 Data Validation

Validation is a process of checking if data satisfies certain criteria when input i.e. falls within accepted boundaries. It is carried out by the computer. Examples of validation techniques include:

| Technique | Description |
|---|---|
| Length check | This checks if the input data contains the required number of characters e.g. if a field needs 6 digits, then inputting 5 digits or 7 digits, for example, should be rejected. |
| Range check | This checks whether data is within given/acceptable values e.g. checks if a person's age is > 0 but is also < 140. |
| Character check | This checks that the input data doesn't contain invalid (type check) characters; e.g. a person's name shouldn't contain numbers. |
| Format check | This checks that data is in a specified format (template) e.g. (picture check) date should be in the form dd/mm/yyyy.<br>Note : UK – we use dd/mm/yyyy but US- we use mm/dd/yyyy |
| Presence check | Check that data is actually present and not missed out e.g. in an electronic form, somebody's telephone number may be a required field. |
| Limit check | This is similar to a range check except only ONE of the limits (boundaries) is checked e.g. input data must be > 10. |
| Check digit | This is an extra digit added to a number which is calculated from the digits; the computer re-calculates and validates the check digit following input of the number |
| Consistency check | This checks if fields correspond (tie up) with each other e.g. if Cross field check ***Mr.*** has been typed into a field called "Title" then the "Gender" field must contain ***M*** or ***Male***. |

## 16.8 Data Verification

It is defined as the process of checking the input data with the original data to make sure that there have been no transcription errors when data is copied from one medium to another. It can be done both manually and electronically.

There are 3 main method which include:

1. **Double entry** – in this method, there is input of the data twice to the computer system. The computer then checks the two data values (which should be the same) and, if they are different, the computer knows that one of the inputs is wrong. It won't know which one is wrong but it can ask the operator to check that particular input. E.g. confirmation of password when registering on a website

2. **Visual check** – This technique checks for errors by comparing entered data with original document.
3. **Parity check**

Parity Detection

The simplest and oldest error detection method. In this scheme, a binary digit called parity is used to indicate whether the number of bits with value of one in a given set of bits is even or odd and is appended to original data. It is usually used to detect transmission error. The *sender* adds the parity bit to existing data bits before transmission

The *receiver* checks for the expected parity, if wrong parity found, the received data is discarded and retransmission is requested. Parity is used not only during data transfer between devices but also when data are transferred between different components of the CPU. Parity technique can be either even or odd.

- ✓ Odd- > no of 1's is odd and Even -> No of 1's is even.
- ✓ Note: Parity can used only for single bit (1 bit only) error only.

E.g. If we use Even parity to transmit the following 7 bits of data

i. _ 110 0111 -> parity bit is 1
ii. _110 0011 ->parity bit is 0

If we use Odd Parity:

i. _111 0001 -> parity bit is 1
ii. _ 011 0001 -> parity bit is 0

## Two Way Parity

### Traverse Parity

Transverse parity is the term normally used when the chunk of data to which the parity bit applies in indeed a single character

### Longitudinal Parity

Longitudinal parity is the term used when the chunk of data to which a parity bit applies is a single bit position of a sequence of characters that make up a block. Thus, effectively a parity byte is added to a block of characters.

**Note:**

The technique can be used as an error correcting technique if one assumes that only a single bit can become corrupt in a block in which case the single bit in the block that matches both a transverse parity check failure and a longitudinal parity bit failure would be the corrupt bit.

**Consider the following block (Assume Even Parity)**

| Parity Bit | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Character 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Character 2 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | Character 3 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | Character 4 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Longitudinal Parity |

## Checksums

***It is defined as a value calculated by adding all blocks in a byte. Checksum is used to check on whether errors have occurred during data transmission.*** If the recipient checksum does not match sender, then an error has occurred and usually retransmission of data is involved.