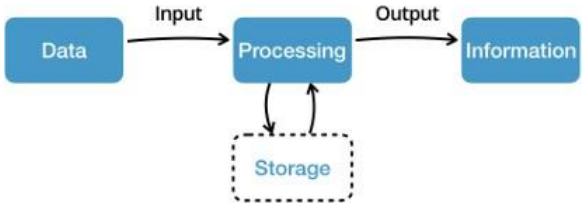
Unit-1(Concept of ICT)

Data vs Information

Data	Information
Raw, unprocessed facts and figures	Processed, organized, and meaningful data
Unstructured and meaningless on its own	Structured and meaningful
Collected for processing	Used for decision-making
Numbers, symbols, letters	Reports, summaries, insights
Input for systems	Output from systems
May contain errors or inconsistencies	Accurate and verified



Lifecycle of Data

Data: 2 types

i. **Qualitative** – non-numerical characters
Eg – Color of a ball, taste of a fruit etc..

ii. **Quantitative** – numerical
Eg – Marks obtained in a test

Characteristics of valuable information

- Relevance
- Timeliness
- Accuracy
- Completeness
- Understandability

Golden Rule of information - Value of information in its maximum level at the moment the information is created or the information is released. The value of the information is reduced gradually with the time and become less valuable.

Big Data – 3 main properties (Volume, velocity, variety)

Drawbacks of manual methods in manipulating data and information

- Inconsistency and duplication in data
- Room for errors
- Delay in processing
- Lack of sharing information
- Reduced customer services

The Internet

- The Internet is a network of networks where users can access information from other connected computers.
- Developed from ARPANET by the U.S. Department of Defense in 1969.
- Initially used by computer professionals, engineers, and scientists. Now a public, cooperative, and self-sustaining facility used by millions globally.
- Utilizes a portion of public telecommunication networks.

World Wide Web (WWW)

- Was created in 1991 by Tim Berners-Lee.
- HTTP (Hypertext Transfer Protocol): A set of rules for exchanging files like text, images, sound, video, and other multimedia. This Enables communication between web browsers and servers.
- Web Browser is a Software required to view web pages

Mobile communication refers to the wireless transmission of information without using cables or electrical conductors. Initially limited to one pair of users on a single channel, modern mobile communication uses techniques like TDMA, CDMA, and FDMA to accommodate multiple users. **Mobile computing** enables the transmission of data, voice, and video through wireless-enabled devices, allowing users to stay connected without relying on fixed physical links. (Mobile computing = Mobile Hardware + Mobile Software + Mobile Communication)

Cloud Computing

Refers to a Network or Internet. A Cloud is present at a remote location and provides services over a network.

3 Categories →

1. **Infrastructure as a Service (IaaS)** - provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc.
Ex - Customer Relationship Management, games, virtual desktop applications

---

2. **Platform as a Service (PaaS)** - provides the runtime environment for applications, development C deployment tools, etc. Ex - Database, web server, deployment tools

---

3. **Software as a Service (SaaS)** - allows to use software applications as a service to end users. Ex – Virtual machines, servers, storage, networks

---

A system can be defined as an interrelated components work together to achieve a common objective.

In an information system data input into the system is Processed using processing instructions given and information is produced as output. **This is called an abstract model of information.**

Computer System consists of four major components
• Hardware • Software • Firmware • Live ware

Output Devices
• **Monitor (Visual display Unit)** -> Forms image using pixels, Resolution of images depend on (ppi)
i. Cathode-Ray Tube (CRT)
ii. Flat-Panel Display.
a. Liquid Crystal Display (LCD)
b. Light Emitting Diode (LED)

• **Printer** -> Produces hard copies
i. Impact printers - Dot Matrix Printer, Line printer, Daisywheel
ii. Non-Impact printers - Laser printers and Inkjet printers, plotters

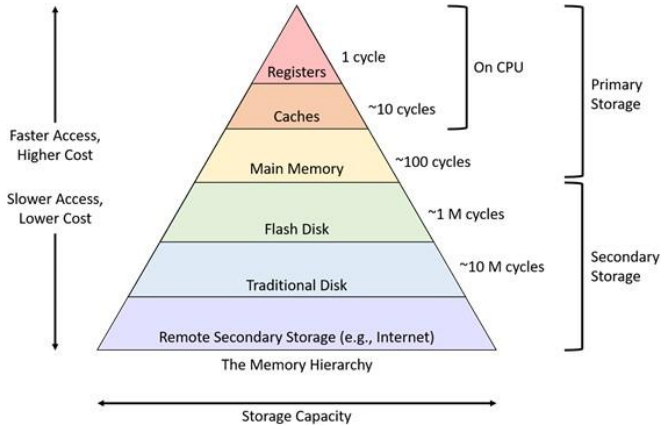
---

Motherboard (It is a circuit board located inside a computer system which every hardware components are connected to each other)

Memory Devices - used to store data and instructions,
• **Cache Memory** -> a very high-speed semiconductor memory which can speed up the CPU, acts as a buffer between the CPU and the main memory, It is known as CPU memory, volatile.

• **Primary Memory/Main Memory** -> Holds only those data and instructions on which the computer is currently working, has a limited capacity. Volatile. Generally made up of semiconductor device.

• **Secondary Memory/ External Memory/ Auxiliary Memory** -> non-volatile, slower than the main memory. Used for storing data and information permanently. CPU does not access these memories directly. Eg: Hard Disk, DVD, Flash drive, Memory Cards



Computer Software
1. **System Software** - Designed to operate and control the computer hardware and It provides a platform for running application software.
i. **Operating Systems** – Windows, Linux, Mac OS
ii. **Utility Software** – Virus Guard, Device drivers
iii. **Language Translators** – Compiler, Interpreter

---

2. **Application Software -** designed to satisfy a particular need of a particular environment.
E.g. - MS Office, Paint
i. **Open-Source Software** - Source code is available. Free to use, modify or redistribute.
ii. **Proprietary Software** - Source code is not available. Should have the permission from the owner of the software to use it.

Firmware - A computer program that is “Embedded" in s hardware device. It is called a Embedded software. In computers, firmware embedded in ROM and which handles booting up process of computers.

The steps in data processing
Data gathering → Data validation → Data input → Data Processing → Data output → Data storage

---

Models of data input
1. Direct and Remote. 2. Online and Offline.

Data Processing

1. Batch processing - easy to handle huge amount of data. E.g. - (Billing systems, pay roll systems etc..)

2. Real time processing - Input, Processing and output as well as data storing occur simultaneously.
E.g. - monitoring systems, booking systems

Applications of ICT

1) Education → CBE (Computer Based Education), Simulations, Distance Learning and E-Learning, Individualized Learning.

2) Healthcare → ECG, EEG, CT scans, Diagnostic Systems, Patient Monitoring System, Pharmacy Information System

3) Agriculture → used to control conditions inside the green houses, Robotic machines, Automatic weed removers, insect control devices, Monitoring devices

4) Business and finance → payroll calculations, budgeting, sales analysis, financial forecasting, online accounting facility, ATM machines.

5) Engineering → CAD, CAM, CIM, 3D virtualization.

6) Tourism → Online planning and booking systems.

7) Media and journalism

8) Law enforcement → CCTV systems, GPS.

The impact of ICT in the society

• **Social and economic benefits of ICT** → impact in entertainment and leisure industry(online games and chat with friends etc..), increases in productivity, GDP and trade. access to information.

• **Social, economic, environmental, ethical, legal and Privacy issues caused by ICT** → job losses, time wastage, health issues, Digital divide(refers to the gap between individuals, communities, or countries with access to modern digital technology and those without.), e-waste. Property and right violations.

Green computing is the practice of using computing resources efficiently and eco-friendlily, aiming to reduce e-waste and promote recycling of electronic components, with many developed countries implementing regulations to minimize environmental impact.

Key Concepts in Information Security s Ethics

a) Confidentiality of Data/Information - Ensured using password protection, encryption techniques, and digital signatures.

b) Stealing/Phishing - A method used to steal sensitive information by pretending to be a trusted entity. Protection includes firewalls, passwords, updates, and avoiding suspicious emails/links.

c) Piracy - Illegal copying, distribution, or use of software.

d) Copyright/Intellectual Property Laws - Protects original works such as inventions, books, and creative content under intellectual property rights.

e) Plagiarism - Plagiarism is stealing and presenting someone else's work as your own without giving credit.

A **software license** governs the use, modification, and distribution of software. Proprietary software keeps the source code hidden, while **FOSS** allows users to view, modify, and share the code.

Unlicensed software includes public domain or internal software used within organizations. Using unlicensed software outside legal terms can lead to copyright issues.

Unit-2(Introduction to Computer)

► ***Computer Evolution***

× **Pre-Mechanical Era (before 1450) – Abacus**; the first calculating device. used by Chinese around 5000 BC.

× **Mechanical era (1450-1840)**

- **Pascaline** - by Blaise Pascal in 1642. It could add and subtract two numbers.
- **Stepped Reckoner** - by Gottfried Wilhelm Leibniz in 1694. Carried out operations add, subtract, multiply and division.
- **Difference engine** - first mechanical computer – developed by Charles Babbage in 1822.
  He is considered as the "**Father of the computer**".
  Madam **Ada Augusta Lovelace** is considered as the first programmer since she wrote write programmes for the Analytical Engine(Also developed by Charles Babbage).

× **Electro mechanical era (1840 – 1G40)**

- **Tabulating machines –** by Herman Hollerith(1890)
- **Mark 1 -** the first automatic computer invented by Professor Howard Aiken in 1939.

× **Electronic Era**

1st Generation Computers (1G40-1G56) –

vacuum tubes as the basic component.

- **ENIAC** (Electronic Numerical Integrator and Calculator) the first electronic digital computer designed by John Mauchly and J. Presper Eckert in 1946.
- **EDVAC** (Electronic Discrete Variable Automatic Computer) is considered as the first digital computer that could store program was built in 1948.
- **EDSAC** (Electronic Delay Storage Automatic Calculator) was the first full size stored program computer developed by Maurice Wilkes in 1949.
- **UNIVAC** (Universal Automatic Computer) by Remington Rand.

Relied on machine language, Can solve only 1 problem At a time, Inputs were based on punched cards and paper Tapes, output on papers, expensive C often caused Malfunctioning, expensive, high electricity consumption And heat wastage.

2nd Generation Computers (1G56 – 1G63)

transistors were used as the fundamental building block. IBM 1620, IBM 7094, CDC 1604, CDC 3600, UNIVAC 1108 were some computers developed in this generation.

Relied on magnetic core memory, magnetic tapes.
Punched cards for input and print papers for output.
Assembly language was introduced

3rd Generation Computers (1G64 – 1G75)

Used Integrated Circuits (ICs)
IBM-360 series, PDP (Personal Data Processor), TDC-316 were the computers of 3rd generation.
Punched cards and printouts were replaced by keyboards and monitors, Ascii was introduced (in 1963)

4th Generation (1G75 – 1G8G)

VLSI (Very Large Scale Integrated) circuits were used.
Introduction of microprocessors (e.g., Intel 8086, 80286).
Development of graphical user interfaces (GUIs).

IBM introduced the first PC (1981).
Apple introduced Macintosh (1984) with a GUI and mouse.
MS-DOS was widely used on IBM PCs.
Microsoft launched Windows 1.0 (1985).
LAN (Local Area Network) emerged.

5th Generation Computers (1G8G – present)

The ULSI (Ultra Large Scale Integration) technology is used.
Focused of AI and Machine learning
Eg: Desktop, Laptop, Notebook, Ultrabook

► **Classification of Computers**

1. Based on the Technology - Analog Computer, Digital Computer
2. Based on the Purpose - Special purpose computer, General purpose computer
3. Based on Size - Super Computer (E.g. -TIANHE-1), Mainframe Computer, Mini Computer, Micro Computer(Desktop, Laptop, Notebook, Smart, Tablet)

► **Input Devices**

* Keyboard
* Pointing devices (Mouse, Touchpad, Remote control, touch screen)
* Direct entry (Magnetic strip reader, Bar code reader, smart card reader)
* Image and video(video camera, Digital camera, Webcam, CCTV)
* Scanning(Flatbed scanner, MICR, OMR, OCR, Graphic tablet, digitizer, Microphone)

Advantages of direct entry input device over key board entry input device - Automatically capture data, data entry cost is reduced, Accurate, more efficient.

► **Output devices**

Monitor

1. **CRT Monitor** (Cathode Ray Tube Monitor) - Images are produced when an electron beam strikes a phosphorescent surface
2. **LCD Monitor** (using Thin Film Transistor-TFT) Liquid Crystal Display (LCD) that has a transistor for each pixel. widely used in notebook and laptops.
3. **LED Monitor** (Light Emitting Diode Monitor) - Used as a computer monitor or television.

Advantages →Less expensive, More reliable, Generate low heat and consume less power, Longer lifespan and less environmental impact.

Printer - Used to produce a hard copy output.

1. **Dot matrix printer** - Uses a print head that moves on the page and prints by striking an ink ribbon against the paper.
2. **Inkjet printer** - The print head has several jets. As the paper moves past the print head, the nozzles spray ink onto it, forming the characters and images.
3. **Laser printer** - A laser beam draws the document on a selenium-coated drum using electrical charges. After the drum is charged, it is rolled in toner, a dry powder type of ink. The toner adheres to the charged image on the drum. The toner is then transferred onto a piece of paper and fused to the paper with heat and pressure.
4. **Graphic plotter** – interprets commands from a computer to make line drawings on paper with automated pens.

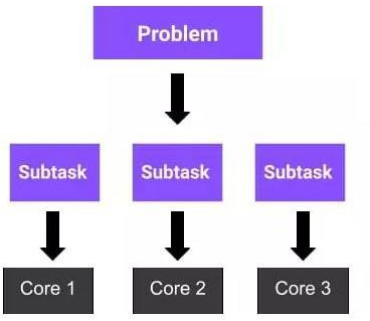
► **CPU and its compatibility with motherboard**

1. Socket support
2. Chipset support
3. Motherboard wattage support
4. BIOS support

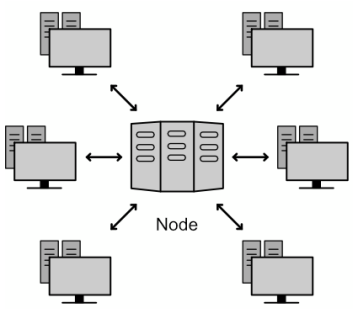
► **Storage Devices**

1. Fixed internal magnetic hard disk
2. External hard disk
3. Magnetic tape
4. Optical disc
5. Flash drive
6. Memory card

Parallel computing – A type of computing Architecture where multiple processors or cores work together simultaneously to solve a problem faster. It divides large tasks into smaller subtasks, which are processed in parallel.



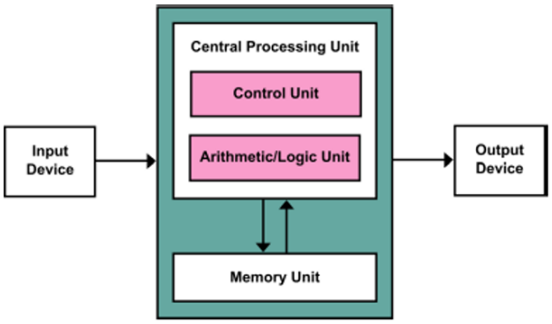
► **Grid computing** – Where Multiple computers across different locations work together to solve large-scale problems by sharing resources such as processing power, storage, and data over a network.



► **Schedulers** - Schedulers are types of applications responsible for the management of jobs

---

► **Von-Neumann Architecture**



Major components of this architecture:

1. Central processing unit

(i) Control unit (CU) : This unit controls signals of all devices of a computer system.

(ii) Arithmetic and logic unit (ALU) : It carries out mathematical and logical operations.

(iii) Memory register : A CPU register is one of a small set of data holding places which is part of the computer processor. A register may hold an instruction, a storage address, or any kind of data

2. Memory

(i) Primary memory

(ii) Secondary memory

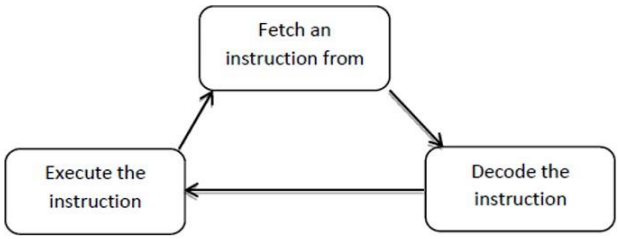
3. Input device

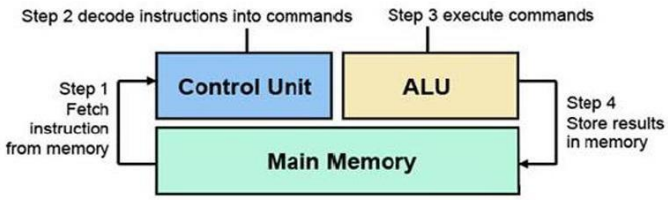
4. Output device

Data bus - A system within a computer or device, consisting of a connector or set of wires, that provides transportation for data.

Control bus - Used to transmit a variety of control signals to components and devices.

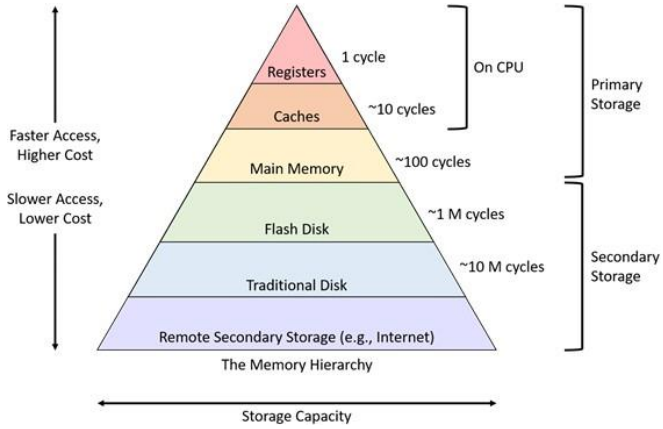
► **Fetch-execute-cycle**





► **Multi-core processor** is a single computing component with two or more independent actual processing units (cores), which are units that read and execute program instructions. Therefore, the single processor can run multiple instructions on separate cores at the same time.
Need of multi-core processor
1. Can be run a program by dividing some parts. So, it gets executed fast.
2. It enables parallel programming.
3. To get the high performance from a single machine.

Memory Hierarchy



► **Registers**
Allow the CPU to execute various tasks at very high speed.
2 types → General purpose C Special purpose registers
Memory Buffer register (MBR) stores the data being transferred from the memory for data processing and the data to be transferred to the memory.
Memory Address Register (MAR) contains the address of The memory of which we either want to read or write
Program Counter(PC) holds the address of the next instruction to be executed
The **Instruction Register** (IR) holds the fetched instruction, while the Control Unit (CU) decodes it. The Current **Instruction Register** (CIR) holds the instruction currently being executed.
Accumulator is a memory register to store intermediate values of arithmetic C logical processing

---

► **Volatile memory** - only holds the data while the device is powered.
1. Cache memory - Used to store program instructions that are frequently accessed by software during operation.
i. ***Level 1 (L1) cache*** - Extremely fast but relatively small, and is usually embedded in the processor chip (CPU).
ii. ***Level 2 (L2) cache*** - often more capacity than L1. It may be located on the CPU or on a separate.
iii. **Level 3 (L3) cache** - works to improve the performance of L1 and L2. It can be significantly slower than L1 or L2, but is usually double the speed of RAM.

---

2. RAM (Random Access Memory) - the main memory of The computer that holds data for running applications and required data for a computer.
i. DRAM (Dynamic RAM) - continuously refreshed or it will lose its contents.
ii. SRAM(Static RAM) – consists of a flip-flop which consists of 6 transistors to keep a bit. No need for constant refreshing. Lesser data density and relatively expensive than DRAM.
iii. SDRAM (Synchronous DRAM) It is a type of memory that synchronizes itself with the computer's system clock.
iv. DDR SDRAM(Double Data Rate SDRAM) doubles the number of read or write operations to two accesses per system clock cycle.

► **Non-Volatile memory -** This is a type of computer memory that has the capability to hold saved data even if the power is turned off.

1. ROM (Read Only Memory) - ROM stores essential programs such as the program that boots the computer.
Types of ROM
i. **PROM** (Programmable ROM) - data can be written only once. Once a program has been written onto a PROM, it remains there forever.
ii. **EPROM** (Erasable PROM) retains its contents until it is exposed to ultraviolet light. The ultraviolet light clears its contents, making it possible to reprogram the memory.
iii. **EEPROM** (Electrically Erasable PROM) It can be erased by exposing it to an electrical charge.

2. Secondary storage

i. **Magnetic storage device** - Manipulation of magnetic fields on a medium in order to record audio, video or other data. Mostly involves a tape medium that is moved by read and write heads.(Uses sequential access)

Eg: Hard disk, Floppy disk, Magnetic tape

ii. **Optical storage device** - Data is written and read with a laser for archival or backup purposes. Typically, data is written to optical media, such as CDs and DVDs and Blu-Ray discs. Data is stored as pits and bumps in the disc surface, representing binary values

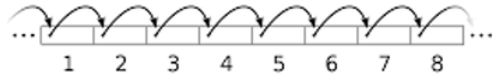
iii. **Solid state storage** - made from silicon microchips. It stores data electronically Solid - state storage can be found in three form factors: solid-state drives (SSD), solid-state cards (SSC) and solid-state modules (SSM). allows data transfer to and from storage media at a much higher speed and more durable.

Eg: Flash drive, Memory card

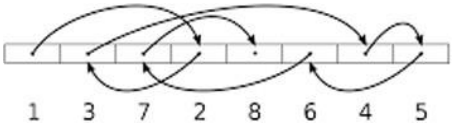
CMOS (Complementary Metal-Oxide-Semiconductor) memory is a small amount of volatile memory that stores BIOS (Basic Input/Output System) settings in a computer. It is backed by a battery, known as the CMOS battery, which ensures that the settings are retained even when the computer is powered off.

Memory Access methods

(i) Sequential access

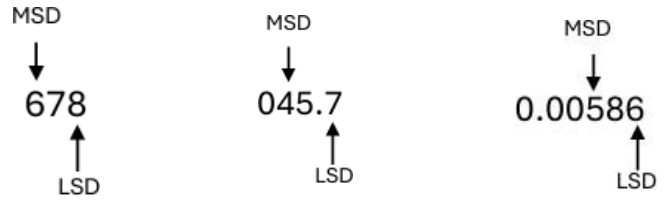


(ii) Random access



Unit-3(Number Systems)

✦ Most Significant Digit and Least Significant Digit



You can use this method to find the most and least significant digits of decimal, octal, and hexadecimal numbers.

✦ Most Significant Bit and Least Significant Bit

Only binary number system is used to identify the MSB and the LSB.

rightmost bit is the LSB, and the leftmost bit that is not zero is the MSB.

✦ Decimal Number System

- These are the numbers that we use in day-to-day life.
- The base is 10.
- Digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Positive values are represented by (+) and negative values are represented by (-)
- Contains integer values and fractional values. Fractional part is separated by a dot (.)

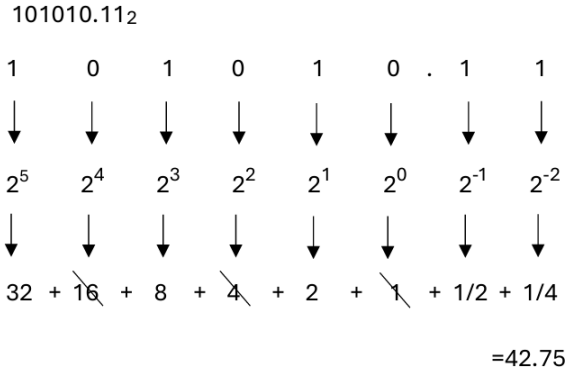
Eg - 345.632

✦ Binary Number System

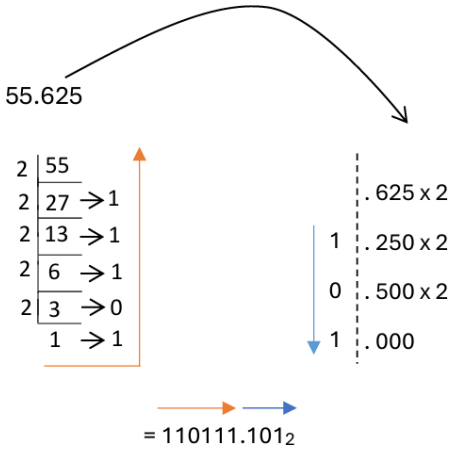
This system has two digits which can represent two states.
- These two states are represented by digits 1 and 0.
- In binary numbers, place values are multiples of 2.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

→ Binary to Decimal



→ Decimal to Binary



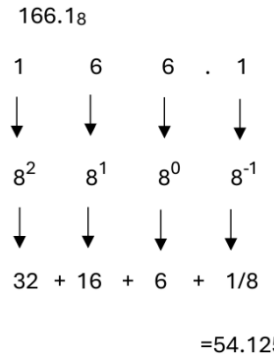
✦ Octal Number System

- Base value is 8.
- Digits used in this representation are 0,1,2,3,4,5,6 C 7.
- In octal numbers, place values are multiples of 8.

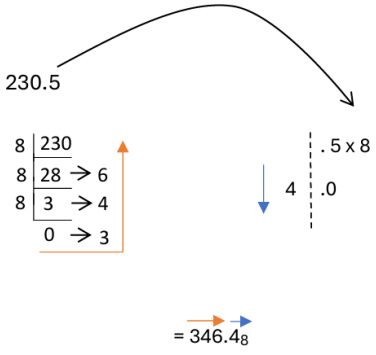
E.g :

8^2	8^1	8^0	8^{-1}	8^{-2}
64	8	1	1/8	1/64

→ Octal to Decimal



→ Decimal to Octal



✦ Hexadecimal Number System

- Base value is 16.
- Contains 16 digits starting from 0 to last digit F.
- The values of the digits are represented as follows. usernames, passwords, and different access levels.

Hexadecimal	0	1	2	3	4	5
Decimal	0	1	2	3	4	5

6	7	8	9	A	B	C	D	E	F
6	7	8	9	10	11	12	13	14	15

➔ Hexadecimal to Decimal

$2A1.2_{16}$

2 A 1 . 2

16^2 16^1 16^0 16^{-1}

512 + 160 + 1 + 0.125

=673.125

➔ Decimal to Hexadecimal

422.25

8 | 422
8 | 26 → 6
8 | 1 → 10
0 → 1

. 25 x 16
4 .0

$= 1A6.4_{16}$

✦ Conversions between Number systems

➔ Binary to Octal

* Group 3 binary bits each, starting from the right.
* Find the total of each group and use the corresponding octal digit.

Eg – 11001101_2

0 1 1 | 0 0 1 | 1 0 1_2

3 1 5 $_8$

→ Octal to Binary

* Represent each octal digit as 3 binary bits.

Eg - 607_8

6 0 7 $_8$

1 1 0 | 0 0 0 | 1 1 1_2

➔ Binary to Hexadecimal

* Group 4 binary bits each, starting from the right.
* Find the total of each group and use the corresponding octal digit.

Eg – 1010100111101_2

1 0 1 0 | 1 0 0 1 | 1 1 0 1_2

A 9 D $_{16}$

→ Hexadecimal to Binary

* Represent each octal digit as 4 binary bits.

Eg – $6B3_{16}$

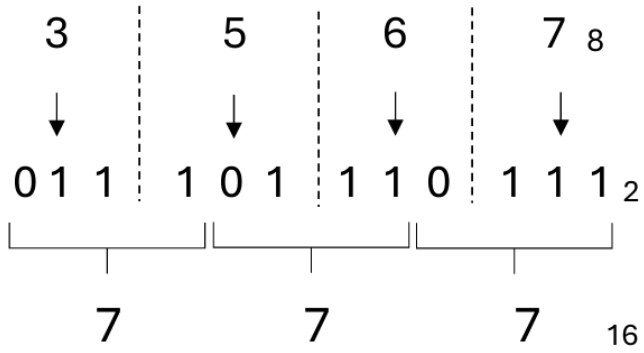
6 B 3 $_{16}$

0 1 1 0 | 1 0 1 0 | 0 0 1 1_2

➔ Octal to Hexadecimal Conversion

* First, convert to binary number and then Convert to Hexadecimal number.

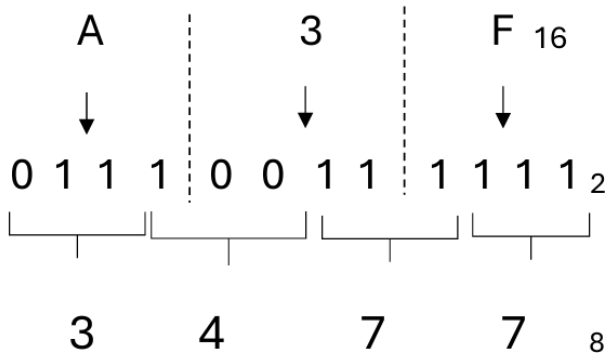
Eg – $6B3_{16}$



➔ Hexadecimal to Octal Conversion

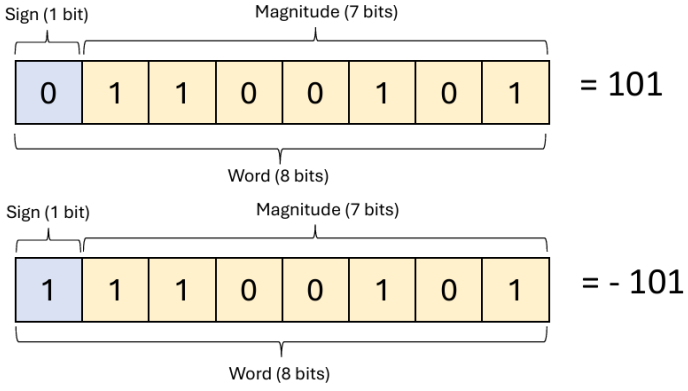
* First, convert to binary number and then Convert to Hexadecimal number.

Eg – $6B3_{16}$



✦ Number Representation

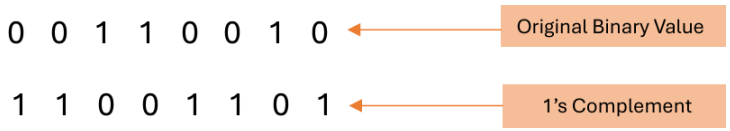
➔ Sign Magnitude



➔ One's Complement

* positive numbers are represented in standard binary form.
* negative numbers are formed by flipping all bits- changing zeros to ones and ones to zeros.

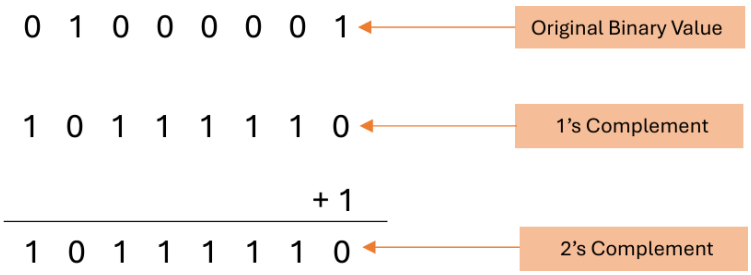
Eg : -50 in One's complement



➔ Two's Complement

* positive numbers are represented in standard binary form.
* to obtain the negative value of a number, invert all the bits of its positive value and then add one to the result.

Eg : - 65

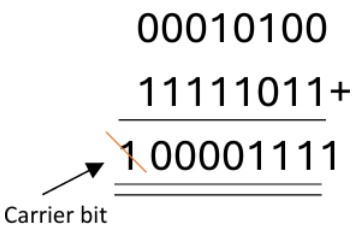


Addition using 1's complement and 2's complement

- Computation 20+(-5) is done in 8-bit two's complement form.

20 → 00010100 **5** → 00000101

- 5 → 11111010 + 00000001 = 11111011



* **In one's Complement additions**, carrier bit is added again to the answer, **But in Two's complement**, carrier bit is ignored

➔ **Differences between 1's Complement and 2's Complement**

1's Complement	2's Complement
1. Zero has two representations, +0 and -0	Zero has only one representation
2. Negative values obtained by flipping bits	To obtain a negative value, should flip bits and add 1
3. More complex for arithmetic operations due to considering carrier bit	Much easier in arithmetic operations
4. Less used in modern computing systems	Widely used in modern computer systems

✦ **Fixed point numbers**

* The decimal point is unchanged.

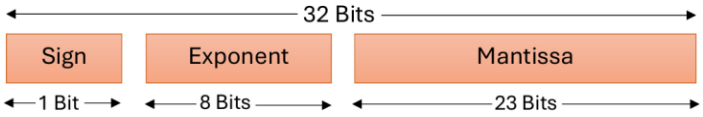
Eg :- 850.753
652.365 +
1503.118

✦ **Floating point numbers**

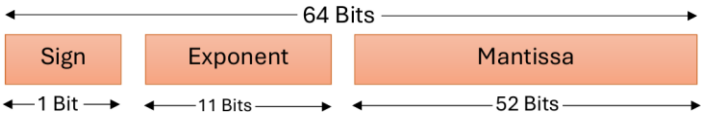
Used to represent:
- Numbers with fractions, such as 3.1416
- Very small numbers, like 0.000000001
- Very large numbers, for example, 3.15576 × 10^9

IEEE 754 standard has three main components:
1. Sign Bit: This indicates the sign of the number.
2. Exponent: A bias is added to the actual exponent
3. The Normalized Mantissa

1. Single Precision



2. Double Precision



Advantages and disadvantages of Fixed-point and Floating-point representations

	Advantage	Disadvantage
Fixed Point Representation	The performance is good, not requiring extra hardware or software logic	Limited range of values can represent
Floating point representation	A wider range of numbers can be represented, with different levels of precision.	Requires more storage space, slower processing times, and reduced precision.

✦ **Coding systems used in Computers**

➔**Binary Coded Decimal (BCD)**
* each decimal digit is represented by 4 bits.

Decimal value	BCD value
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

➔ **ASCII – American Standard Code for Information Interchange**
* 7-bit code used to represent text in computers and communication devices.
* primary encoding system used in IBM PCs.

Extended ASCII expands this by using 8 bits

Eg:

Character	ASCII code	Binary code
null	o	0000000
a	97	1100001
b	98	1100010
0	48	0110000

➔ **EBCDIC (Extended Binary Coded Decimal Interchange Code)**
* 8-bit character encoding is primarily used on IBM mainframe and midrange systems.

Differences between ASCII and EBCDIC

ASCII	EBCDIC
Used for electronic communication	Used on IBM mainframe C IBM midrange computers
Uses 7 bits	Uses 8 bits
Only 128 characters can be represented	256 characters can be represented
Compatible with Unicode	Not compatible with Unicode
More efficient	Less efficient

→UNICODE

* Widely adopted by major companies like Apple, HP, IBM, Microsoft, Oracle, SAP, and many others.
* It can store a vast number of characters, including those from almost every language in the world.

There are different types of Unicode:
- UTF-8: Uses 1 byte (8 bits).
- UTF-16: Uses 2 bytes (16 bits).
- UTF-32: Uses 4 bytes (32 bits).

Differences between ASCII and Unicode

Property	ASCII	Unicode
Space Requirement	Needs less space	Needs more space than ASCII
Bit Usage	Uses 7 bits	Uses 8/16 or 32 bits
Relationship	ASCII is a subset of Unicode	Unicode is the superset of ASCII
Conversion	ASCII →Unicode possible	Most Unicode → ASCII impossible
Language Support	Limited to English	Supports most written languages

✦ **Bitwise operations**

1. NOT operation

E.g:- NOT 01112 (7_{10})
= 10002 (8_{10})

A	NOT A
0	1
1	0

2. Bitwise AND operation

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

E.g. :- 01012 (5_{10}) AND 00112 (3_{10})

$0\ 1\ 0\ 1_2$
$0\ 0\ 1\ 1_2$
$0\ 0\ 0\ 1_2\ (1_{10})$

3. Bitwise OR operation

E.g. :- 01012 (5_{10}) OR 00112 (3_{10})

$0\ 1\ 0\ 1_2$
$0\ 0\ 1\ 1_2$
$0\ 1\ 1\ 1_2\ (7_{10})$

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

4. Bitwise XOR operation

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

E.g :- 00102 (210) XOR 10102 (1010)

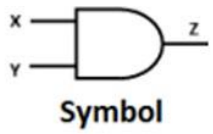
$1\ 0\ 1\ 0_2$
$0\ 0\ 1\ 0_2$
$1\ 0\ 0\ 0_2\ (8_{10})$

	Advantage	Disadvantage
BCD	• Easy to encode and decode decimals into BCD and vice versa. • Simple to implement a hardware algorithm for the BCD converter. • It is very useful in digital systems whenever decimal information is given either as inputs or displayed as outputs. • Digital voltmeters, frequency converters and digital clocks all use BCD as they display output information in decimal.	• Not space efficient. • Difficult to represent the BCD form in high speed digital computers in arithmetic operations, especially when the size and capacity of their internal registers are restricted or limited. •Require a complex design of Arithmetic and logic Unit (ALU) than the straight Binary number system. •The speed of the arithmetic operations slow due to the complete hardware circuitry involved.
ASCII	• Uses a linear ordering of letters. • Different versions are mostly compatible. • compatible with modern encodings	• Not Standardized. • Not represent world languages.
EBCDIC	• uses 8 bits while ASCII uses 7 before it was extended. • Contained more characters than ASCII.	• Does not use a linear ordering of letters. • Different versions are mostly not compatible. • Not compatible with modern encodings
UNICODE	• Standardized. • Represents most written languages in the world. • ASCII has its equivalent within Unicode.	• Need twice as memory to store ASCII characters.

Unit-4(Logic Gates)

✦ Basic Logic Gates

○ AND Gate

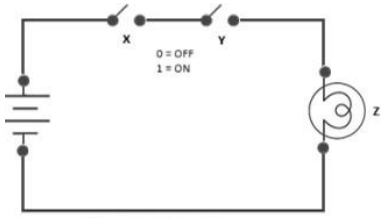


Symbol

Z = X.Y
Boolean Expression

X	Y	Z = X*Y
0	0	0
0	1	0
1	0	0
1	1	1

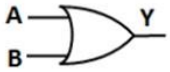
Truth Table



Electrical Circuit

○ OR Gate

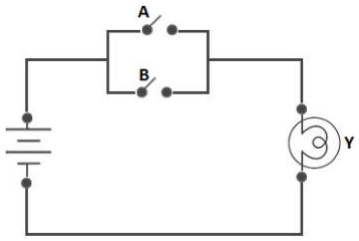
Y = A + B
Boolean Expression



Symbol

A	B	Y = A + B
0	0	0
0	1	1
1	0	1
1	1	1

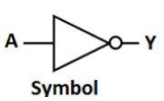
Truth Table



Electrical Circuit

○ NOT Gate

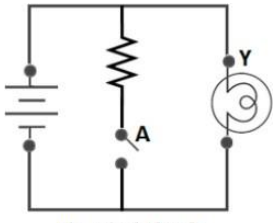
A = $\bar{A}$
Boolean Expression



Symbol

A	Y = $\bar{A}$
0	1
1	0

Truth Table



Electrical Circuit

✦ Combinational Gates

○ XOR Gate

(a) Circuit symbol

S1	S2	L
0	0	0
0	1	1
1	0	1
1	1	0

(b) Truth table

$L = S1 \oplus S2$

(c) Boolean expression

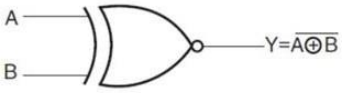
* Results a true if and only one of the inputs to the gate is true.

Truth table for three input XOR gate

INPUTS				Final Output
S1	S2	S3	S1 ⊕ S2	S1 ⊕ S2 ⊕ S3
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

* Returns True If odd number of '1' inputs to the gate.

○ XNOR Gate



$Y = (\overline{A \oplus B}) = (A.B + \overline{A}.\overline{B})$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

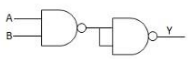
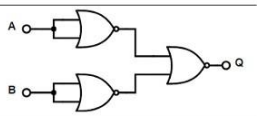
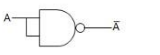
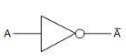
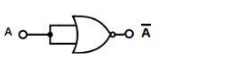
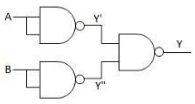
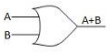
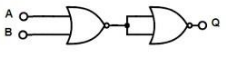
* The logical complement of the XOR gate.

✦ Universal Gates

* Can be used to construct all other logic gates.

* *The advantage of universal gates:*

NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

Using **NAND**	Basic operator	Using **NOR**
		
		
		

✦ Boolean laws

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0+x = x$
Null (or Dominance) Law	$0x = 0$	$1+x = 1$
Idempotent Law	$xx = x$	$x+x = x$
Inverse Law	$x\bar{x} = 0$	$x+\bar{x} = 1$
Commutative Law	$xy = yx$	$x+y = y+x$
Associative Law	$(xy)z = x(yz)$	$(x+y)+z = x+(y+z)$
Distributive Law	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$
Absorption Law	$x(x+y) = x$	$x+xy = x$
DeMorgan's Law	$\overline{(xy)} = \bar{x}+\bar{y}$	$\overline{(x+y)} = \bar{x}\bar{y}$
Double Complement Law	$\bar{\bar{x}} = x$	

✦ Minterms and Maxterms

A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$A+B+C$
$\bar{A}\bar{B}\bar{C}$
$A+\bar{B}+C$
$A+\bar{B}+\bar{C}$
$\bar{A}+B+C$
$A\bar{B}\bar{C}$
$A\bar{B}\bar{C}$
ABC

Maxterm

Minterm

✦ Standard forms in Boolean expressions

- **Sum of Product**

AB + ABC ABC +CDE + $\bar{B}C\bar{D}$

- × A single overbar cannot extend over more than one Variable.
- × A standard SOP expression is one in which all the variables in the domain appear in each product term in the expression. Eg: A $\bar{B}$CD + $\bar{A}\bar{B}$C$\bar{D}$ + AB$C\bar{D}$

- **Product of Sum**

(A+B)(A+B+C)($\bar{A}$+C) ($\bar{A}$+ $\bar{B}$+C)(C+$\bar{D}$+E)(+ $\bar{B}$+C+D)

- × A single overbar cannot extend over more than one Variable.
- × A standard POS expression is one in which all the variables in the domain appear in each product term in the expression. Eg: (A+ $\bar{B}$ +C)($\bar{A}$ + $\bar{B}$ +C)(A+B+$\bar{C}$)

✦ Transforming SOP into POS and vice versa

SOP TO POS

$F = A.\bar{B} + B.\bar{C} + \bar{A}.C$

$= \overline{A.\bar{B} + B.\bar{C} + \bar{A}C}$ — *Obtain the compliment of whole equation by adding an over bar to the whole*

$= \overline{A.\bar{B}} + \overline{B.\bar{C}} + \overline{\bar{A}C}$ — De Morgan's Law

$= (\bar{A}+\bar{\bar{B}}).(\bar{B}+\bar{\bar{C}})+(\bar{\bar{A}}+\bar{C})$ — De Morgan's Law

$= (\bar{A}+B).(\bar{B}+C).(A+\bar{C})$ — Double Inverse

POS TO SOP

$F = (\bar{A}+B).(\bar{B}+C).(A+\bar{C})$

$= \overline{(\bar{A}+B).(\bar{B}+C).(A+\bar{C})}$ — *Obtain the compliment of whole equation by adding an over bar to the whole*

$= (\overline{\bar{A}+B})+(\overline{\bar{B}+C})+(\overline{A+\bar{C}})$ — De Morgan's Law

$= (\bar{\bar{A}}.\bar{B})+(\bar{\bar{B}}.\bar{C})+(\bar{A}.\bar{\bar{C}})$ — De Morgan's Law

$= A.\bar{B} + B.\bar{C} + \bar{A}.C$ — Double Inverse

✦ Karnaugh Maps

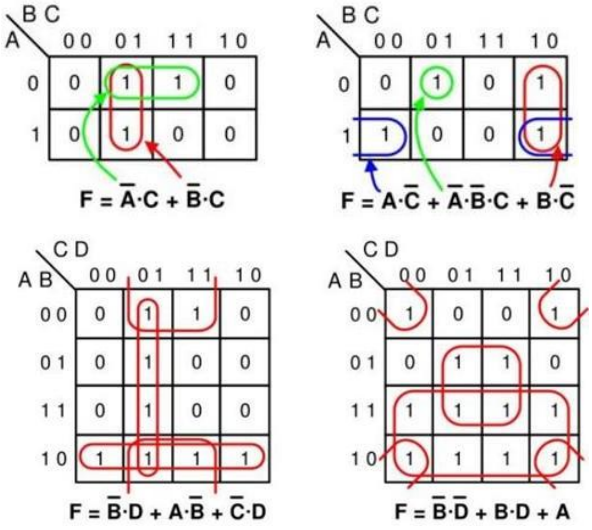
* An alternative way of simplifying logic circuits. Instead of using Boolean algebra simplification techniques

No. of cells = 2^n Where n is the number of input variables

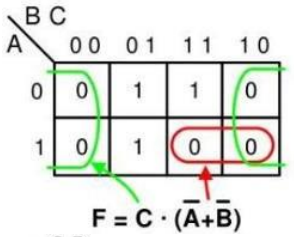
Rules of K- map simplification

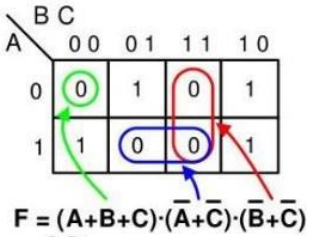
1. No zeros allowed in a group.
2. No diagonals.
3. Only power of 2 number of cells in each group.
4. Groups should be as large as possible.
5. Every "one" must be in at least one group.
6. Overlapping allowed.
7. Wrap around allowed.
8. Fewest number of groups possible.

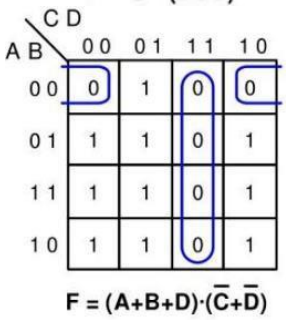
SOP Loops

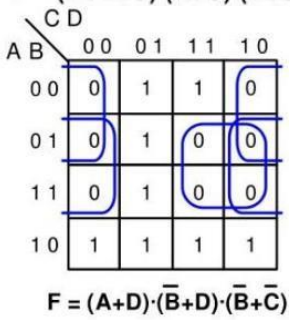


POS loops





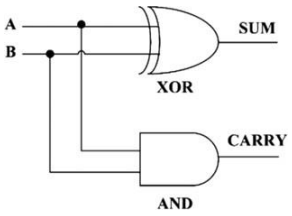




✦ **Half adder**

* used to add two single-bit binary numbers, producing two outputs: Sum (S) and Carry (C).
* sum is calculated using an XOR gate (S = A ⊕ B)
* carry is determined using an AND gate (C = A . B).

INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



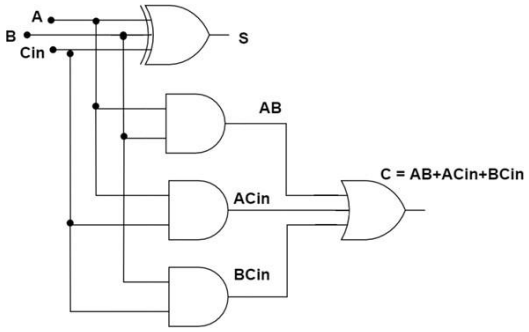
Combinational circuits Vs Sequential circuits

Feature	Combinational Circuits	Sequential Circuits
Dependence	Only on current inputs	On current inputs and previous states
Memory Elements	None	Present
Applications	Arithmetic operations, Data routing	Memory devices, Digital clocks, State machines
Examples	Adders, Multiplexers, Encoders, Decoders	Flip-flops, Counters, Registers

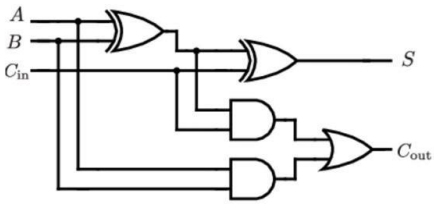
Full Adder

* full-adder has three inputs and two outputs.
* carry is designated as Carry Out (Cout = A . B . C)
* Sum (S = A⊕ B ⊕ C)

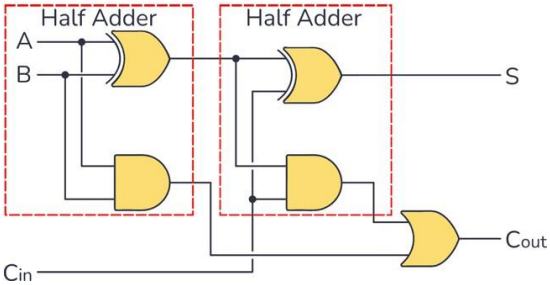
Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Method 2:



◆ **Full adder using two half adders**



✦ Sequential circuits

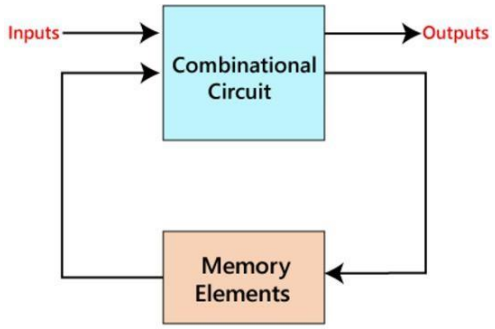


Figure – structure of a sequential circuit

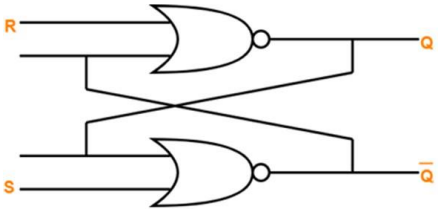
○ Flip Flops

* An application of logic gates
* Can create memory with them
* Can also be considered as the most basic idea of a Random-Access Memory.
* When a certain input value is given to them, they will be remembered.

○ Latch Flip-flop

* R-S (Reset Set) flip flop is the simplest flip flop of all and easiest to understand
* A device which has two outputs one output being the inverse or complement of the other, and two inputs.
* A pulse on one of the inputs to take on a particular logical state.
* The outputs will then remain in this state until a similar pulse is applied to the other input. The two inputs are called the Set and Reset input.

- **Latch constructed using NOR gates**

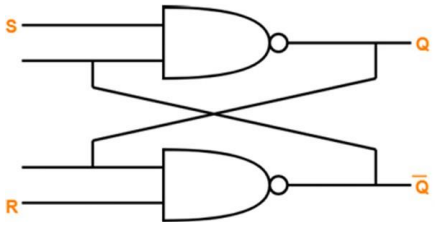


Truth Table

Set	Reset	Output
0	0	No change*
1	0	Q=1
0	1	Q=0
1	1	Invalid state

*can be used for data storage

- **Latch constricted using NAND gates**

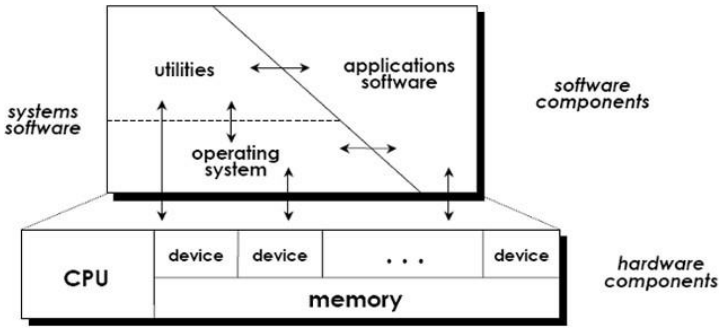


Truth Table

Set	Reset	Output
1	1	No change*
0	1	Q=1
1	0	Q=0
0	0	Invalid state

*can be used for data storage

Unit-5(Operating Systems)



Initial operations of a computer

- When powers on, CPU activates the BIOS.
- The BIOS runs POST (Power On Self-Test) using CMOS memory to check hardware functionality.
- BIOS reads the MBR (Master Boot Record) from the boot drive using the bootstrap loader.
- The operating system is loaded from the boot drive
- into RAM.
- The operating system takes control and displays the user interface.

This process is called **booting**, which means loading the operating system into RAM.

Categories of System Software

1. **Operating System (OS)**
   - Manages hardware and software, allowing user to access and use the computer's functions.
   - Acts as an interface between system software, application software, and hardware.
2. **Utility Software**
   - Manages and analyzes computer resources.
   - Performs specific tasks such as:
     - **Anti-Virus Software**
     - **Disk Formatting**
     - **Disk Defragmenters**
     - **Disk Cleaners**.
3. **Language Translators**
   - Convert high-level or assembly languages into machine language (binary code). Examples:
     - **Compiler**: Converts the entire program at once, reporting errors after.
     - **Interpreter**: Translates and executes code line by line, reporting errors immediately.
     - **Assembler**: Converts assembly language into machine language.

Application Software

- Runs on the Operating System, used for performing user-specific tasks such as creating documents, performing calculations, Playing games

Evolution of Operating System

1) No OS (late 1G40s – mid 1G50s) → Manual Program Scheduling, Uni-programming, processor sat idle when loading programs and doing I/O.

2) Simple Batch Systems → No direct access to hardware, Uni-programming, High response time, Processor sat idle during I/O

3) Multi-Programmed batch Systems → Introduced in the 3rd generation, focuses on reducing processor idle time during I/O operations. By partitioning memory to hold multiple programs, the OS switches the processor to another program when one is waiting for I/O.

4) Time Sharing Systems → Introduced to minimize the response time and maximize the user interaction during program execution, Uses context switching, Enables to share the processor time among multiple programs o Rapidly switching among programs.

Different types of Operating Systems

1. Based on the users: Single user, Multi User
2. Based on Number of tasks: Single Task, Multi Task

Different types of modern Operating Systems

- Single user-single task
- Single user-Multi task,
- Multi user-Multi task
- Multi-threading
- Real Time
- Time Sharing Systems

Important functions of an operating System

1. User Interfacing - provides an easy way for users to interact with the hardware, hiding its complexity.

- **Command Line Interface (CLI)** – Commands are entered using the keyboard, Requires correct syntax, Not visually appealing, Uses very little system resources.

- **Graphical User Interface (GUI).** – Uses WIMP: [Windows, Icons, Menus, and Pointer], Visually appealing and user-friendly, Requires more system resources, Includes a cursor for navigation, Slower compared to CLI.

2. Process Management – OS divides a program or task into multiple processes during execution, Only one process runs on the CPU at a time, enables multitasking,

3. Memory Management – OS handles memory allocation for tasks, stores data before processing, keeps results after processing, and frees memory when applications are closed. OS optimizes the use of primary and secondary memory to enhance computer performance.

4. **Device Management** → OS manages peripheral devices used for input and output, Plug and Play features because the OS includes built-in device drivers.

5. **Security Management** → OS manages user permissions, maintains log files, and allows the creation of profiles with usernames, passwords, and different access levels.

6. **Network Management** → Manages network connections, Handles IP address configurations, device settings of connecting in a network. Use of protocols, monitors network performance and usage, Provides tools for diagnosing and troubleshooting network issues.

7. **File Management**

Files : A file is a named collection of related information, usually a sequence of bytes.

A file can be viewed in two different ways.

1. **Logical** (programmer's) view: how the users see the file.
   liners collection of records.
   Image File – cells(pixels) of intensity values
   Linear sequence of bytes.
2. **Physical** (operating system) view: how the file is stored on secondary storage.

Directories are continues used to organize files logically.

File Attributes : file name , type, location(s), organization, access permissions, time and date of creation, File size…
File Types : One of implementation techniques of file type is to include the type as an extension to the file name.
File can be classified into various types based on the content - Executable(.exe), Text(.txt, .docx), Image(.png)…

FAT (File Allocation Table)	NTFS (New Technology File System)
Up to 4 GB per file	Supports files larger than 4 GB up to 16 TB
Uses a FAT to keep track of files in storage	Complex, with advanced features. Supports Unicode
Low security	High security
Up to 8 characters (8 for file name and 3 for extension)	Up to 255 characters
No/low compression and encryption	Supports compression and encryption
MS-DOS, Win95/98/ME/XP	Windows XP, Vista, Windows 7, 8, etc.

Fille Access Methods

1. Sequential Method → All the data are in sequential Manner. Basically, use to take backups, Not for running applications. E.g. – Magnetic Tapes

2. Direct Access Method → Data is stored in random locations on the disk. Accessing any data takes almost the same amount of time (Direct Access/Random Access). E.g. - All types of memory except magnetic tapes.

The Utility Programs in an OS

• Utility programs are used to improve the efficiency of a computer.

1. Backup Software → Backups are copies of important data, often stored on removable media.

2. Disk Scanner → Disk scanning detects and fixes physical errors on the disk surface. It repairs damaged data (1's and 0's) on the disk.

3. File Compression → Minimizes the size of a file without damaging its content. Saves hard disk space and increases file transfer speed.

4. Anti-virus Software → Protects the computer by detecting and removing malicious software (malware).

5. Disk Partitioning → Divides a single disk into multiple logical drives. Usually done before installing the operating system.

6. Disk Formatting → Formatting prepares a disk to store data. A disk cannot be used until it is formatted.

Fragmentation → Fragmentation occurs when data blocks are scattered across the disk. This reduces the efficiency of file access, making the system slower.

7. Defragmentation → Defragmentation reorganizes scattered file blocks to minimize fragmentation.

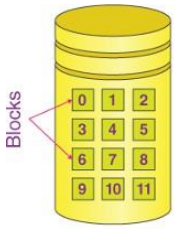
Compaction is the process of collecting all free memory space into a single large block. It helps allocate memory to processes by eliminating scattered fragments.

Internal Fragmentation → Wasted memory inside allocated blocks. Happens when fixed-sized memory blocks are given to processes, and the process doesn't use the entire block.

External Fragmentation → Wasted memory outside allocated blocks. Occurs when free memory is split into small, non-contiguous blocks, making it hard to allocate for new processes even if enough total space exists.

File Storage Management

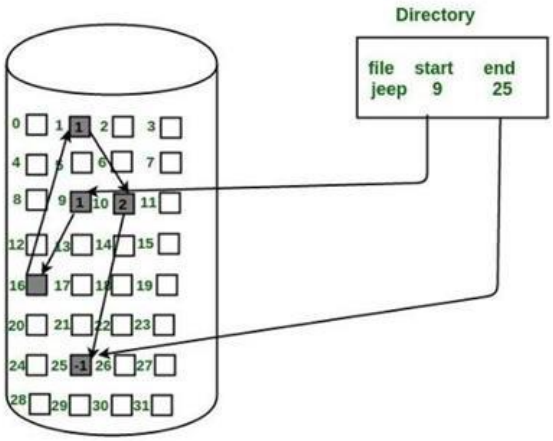
1. Contiguous Allocation → Allocate disk space as a collection of adjacent/contiguous blocks. This technique needs to keep track of unused disk space. Easy Access, Extending file size is difficult, External fragmentation (free unusable space between allocation)



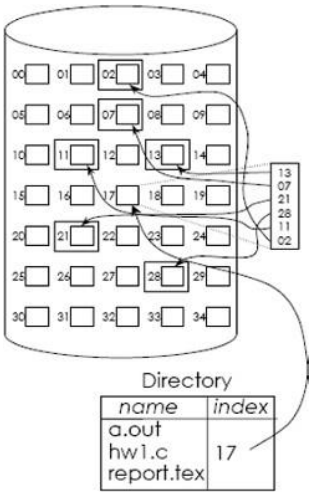
File Name	Start	Length	Allocated Blocks
abc.text	0	3	0, 1, 2
video.mp4	4	2	4, 5
jtp.docx	9	3	9, 10, 11

Directory

2. Linked Allocation → solves all problems of contiguous allocation. Inside each block a link is maintained to point to where the next block of the file is. It Avoids fragmentation but has slower access due to the need to follow links.



3. Indexed Allocation → Creates a table of pointers (index) at the time of the file creation. This table is modified as new blocks are allocated for the file or removed from the file. The index table is also saved in a block/s.
Example : UNIX file system



The two main parts of an Operating System (OS)

1. Kernel → The core component of the OS that directly interacts with hardware. Responsible for managing system resources. Provides essential services to the rest of the OS and applications, like process management, memory management, and device control.

2. Shell (or User Space) → The interface between the user and the OS. Includes utilities and programs that allow users to interact with the system (e.g., command line interface, graphical user interface). Provides application-level functionality, such as file management and task execution.

What is Process? → Process is a fundamental concept in modern operating systems. A process is basically a program in execution. Process is not a program. A program may have many processes.

Types of processes

• I/O bound processes • Processor bound processes

A process must have (at least) → ID, Executable code, Data needed for execution, Execution context (PC, priorities, waiting for I/O or not)

Reasons for process creation → New batch job, User starts a program, OS creates process to provide a service, Running program starts another process

Reasons for process termination → Normal termination, Execution time-limit exceeded, A resource requested is unavailable, An execution error, A memory access violation, An operating system or parent process request, Parent process has terminated.

❖ On process termination, OS reclaims all resources assigned to the process.

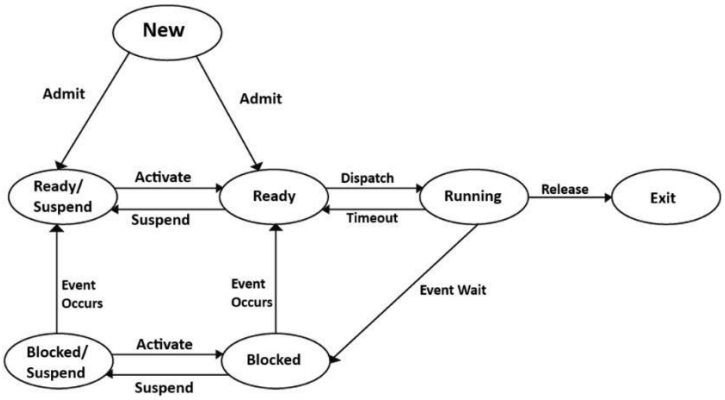
Interrupts → Interrupt is an event that alters the sequence of execution of process. can occur due to a time expiry an OS service request I/O completion. For example when a disk driver has finished transferring the requested data, it generates an interrupt to the OS to inform the OS that the task is over.

Interrupt Handling → Generally I/O models are slower than CPU. After each I/O call, CPU has to sit idle until I/O device complete the operation, and so processor saves the status of current process and executes some other process. When I/O operation is over, I/O devices issue an interrupt to CPU then stores the original process and reserves execution.

Process Management → In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called *process scheduling.* • Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.
• Allocates the processor (CPU) to a process.
• De-allocates processor when a process is no longer Required

Swapping is a mechanism in which a process can be swapped temporarily out of main memory (or move) to secondary storage (disk) and make that memory available to other processes. At some later time, the system swaps back the process from the secondary storage to main memory.

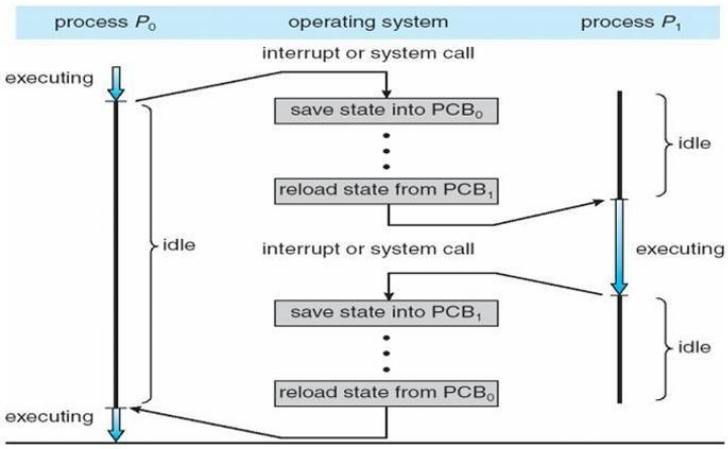
Seven State Process Transition diagram



Process Control Block (PCB)
A Process Control Block is a data structure maintained by the Operating System for every process. The PCB is identified by a process ID (PID). A PCB keeps all the information needed to keep track of a process.
A simplified diagram of a PCB:

Process ID
State
Pointer
Priority
Program Counter
CPU Register
I/O Information
Accounting Information
etc

Context switching is the process of storing a CPU's state in a process control block so that a process can resume execution later, enabling multiple processes to share a single CPU in a multitasking operating system. During a context switch, the scheduler saves the current process's CPU registers and loads the registers of the next process. While essential for multitasking, context switching adds overhead and can impact performance due to the time required to save and restore modern processors' numerous registers.



Types of Scheduling
1. Long-term scheduling (Job scheduling): It determines which programs are admitted to the system for processing. Job scheduler selects processes from the queue and loads them into memory for execution.
2. Medium-term scheduling: Medium term scheduling is in charge of swapping processes between the main memory and the secondary storage.
3. Short-term scheduling (low-level scheduling): Determines which ready process will be assigned the CPU when it next becomes available.

Long Term Scheduler	Short Term Scheduler	Medium Term Scheduler
Job Scheduler	CPU scheduler	Processes swapping scheduler
Selects processes from a pool and loads them into the memory for execution	Selects those processes which are ready to execute for dispatching	Swapped out/Re-introduces the processes into memory and execution can be continued.
Controls the degree of multiprogramming	Provides lesser control over the degree of multiprogramming	Controls the degree of multiprogramming
Speed is lesser than short term scheduler	Speed is fastest among other two	Speed is in between (short and long term schedulers)

Process Schedulers → Assigning the processor to the processes.

• **Turnaround time** : Time required for a particular process to complete, from submission time to completion.

• **Response time** : The time taken in an interactive program from the issuance of a command to the commence of a response to that command.

• **Throughput** : Number of processes completed per unit time. May range from 10 / second to 1 / hour depending on the specific processes.

• **Waiting time** : How much time a process spends in the ready queue waiting its turn to get on the CPU.

• **Burst Time**: Time required by a process for CPU execution.

Scheduling Policies

• **Non-preemptive** : Once a process is in the running state, it will continue until it terminates or blocks itself for I/O.

• **Preemptive** : Currently running process may be interrupted and moved to the Ready state by the OS. Allows for better service since any one process cannot monopolize the processor for very long.

Scheduling Algorithms

→ The Purpose of a Scheduling algorithm
Maximum CPU utilization, Fare allocation of CPU
Maximum throughput, Minimum turnaround time,
Minimum waiting time 6. Minimum response time

1. First Come First Serve algorithm: The FCFS scheduling algorithm processes tasks in order they arrive. The first task in the queue gets the CPU first. However, if the first task takes a long time, others may have to wait, leading to starvation.

Disadvantages → A process runs until it finishes, delaying others. Long tasks in the beginning can make shorter ones wait too long. High average waiting time compared to other methods, despite being simple to use.

2. Shortest Job First (SJF) Algorithm: This selects the process with the smallest burst time for execution. It minimizes the average waiting time by prioritizing shorter tasks. However, it may cause starvation if longer tasks keep being delayed.

3. Round Robin Scheduling Algorithm: widely used in operating systems. It is a preemptive version of the First Come First Serve (FCFS) algorithm, designed for time-sharing. In RR, each process is given CPU access for a fixed time slice, called the time quantum. If a process finishes during its time quantum, it terminates; otherwise, it returns to the ready queue to wait for its next turn.

4. Priority Scheduling Algorithm: each process is assigned a priority number. Depending on the system, a lower number may indicate higher priority, or vice versa. The process with the highest priority is given the CPU first.

A **Deadlock** happens when two or more processes are waiting for resources that other processes are holding, and none of them can continue. It's like a traffic jam where every car is waiting for another to move, but no one can proceed. Since both processes hold a resource the other process needs, the processes will have to wait indefinitely. OS should avoid deadlocks.

Spooling stands for Simultaneous Peripheral Operations On-Line. It is the process of storing data temporarily in a buffer (like print jobs in a queue) before sending it to a device. It allows slower devices, like printers, to handle tasks at their own speed without holding up the CPU or other processes.

Buffering Data → Device Synchronization → Parallel Operations

Memory management is an operating system function that efficiently handles primary memory by tracking all memory locations, allocating memory to processes, and moving them between main memory and disk during execution. It ensures efficient usage by deciding how much memory to allocate, determining which process gets memory and when, and updating the status when memory is freed or unallocated.

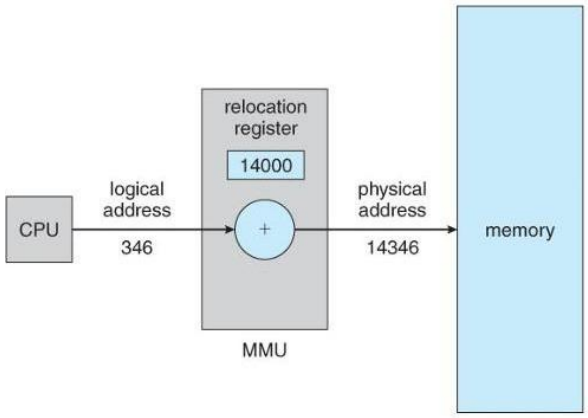
Memory Management Unit (MMU) - This is the hardware device that translates virtual addresses (used by programs) into physical addresses (used by memory). This ensures efficient memory management and security in a system.

How MMU Works:

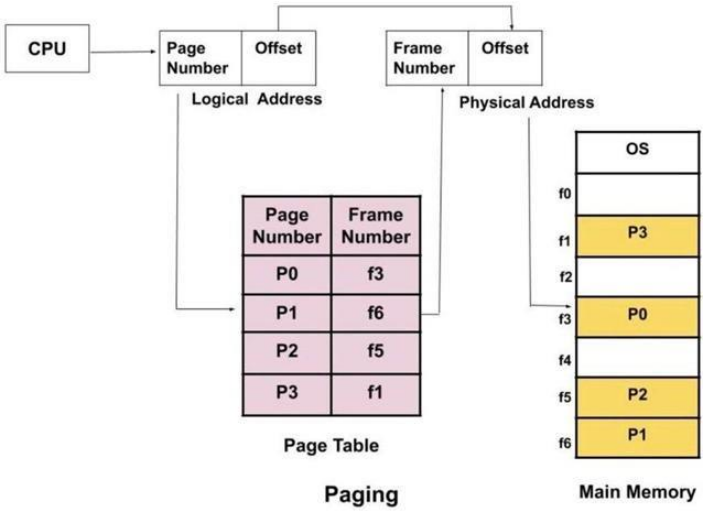
• **Address Translation:** The MMU adds the value in the relocation (base) register to each logical address generated by the program. This translated address is the physical address, where data is actually stored in memory.

• **Logical vs. Physical Addresses:** A program works only with logical addresses. The program never directly interacts with real physical memory addresses.

Physical Address=Base Register Value + Logical Address



Paging allows a process's logical address space to be non-contiguous by dividing physical memory into fixed-sized blocks called frames and logical memory into equally sized pages. The operating system tracks free frames and uses a page table to map logical addresses to physical addresses. To run a program, the OS loads its pages into available frames, though this can lead to internal fragmentation.

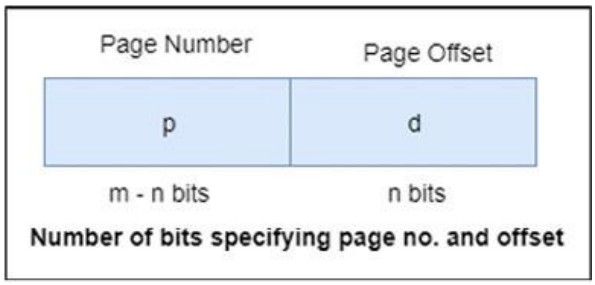


What is Mapping? → refers to the process of converting logical addresses (used by programs) into physical addresses (used by hardware). This is managed by the OS during memory allocation. At runtime, the Memory Management Unit (MMU) – a hardware component—handles this mapping dynamically.

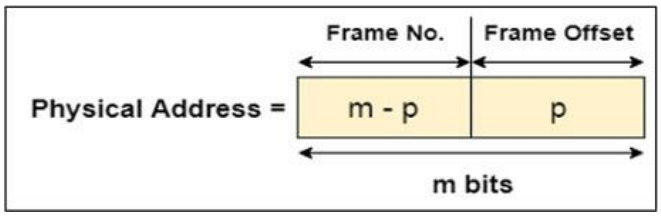
Page Table - A special data structure used in virtual memory systems to map virtual addresses to physical addresses. It plays a crucial role in memory management. Role of the Page Table: The page table contains mappings (or relationships) between pages and frames. Each logical address maps to a specific physical frame via this table.

Virtual memory allows a computer to use more memory than physically available by using disk space to emulate RAM. It enables larger programs and multitasking by loading only required parts of programs into memory, supporting applications larger than physical memory, and allowing multiple programs to run simultaneously. Developers benefit from simplified memory management, and shared memory regions enhance efficiency.

Logical Address



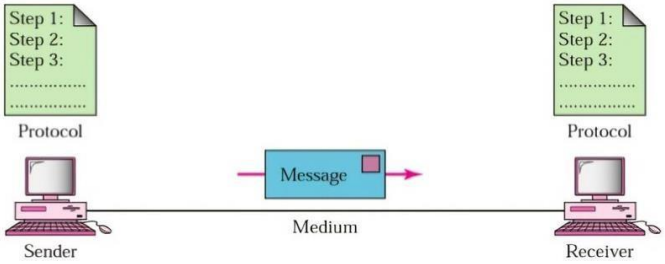
Physical Address



Unit-6(Networking)

Data Communication - the process of transmitting data between two or more communicating devices over some transmission media. Establishing such connections between computing devices is called **computer networking**.

Components of a Data Communication



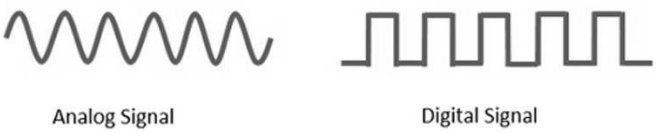
1. **Message** - The information to be communicated.
2. **Sender (Transmitter, Source)** - Any device of sending the Message
3. **Receiver (Destination)** - A device that the sender wants to communicate the message
4. **Communication Medium** - The path by which the message travels from sender to receiver.
5. **Protocol** - Defines the order and the format of data when the data is exchanged between two networking devices.

Signals - An electronic voltage or current, which varies with time. It is used to transfer data from one end to another.

- **Analog signal -** Are in continuous wave form in nature and represented by continuous electromagnetic waves.

Eg - sound, light and temperature

- **Digital signal** - Digital means discrete values C only two values are used to represent something, 1 and 0 (binary values).



Properties of signals

1. Amplitude : The height of the wave measured in meters.

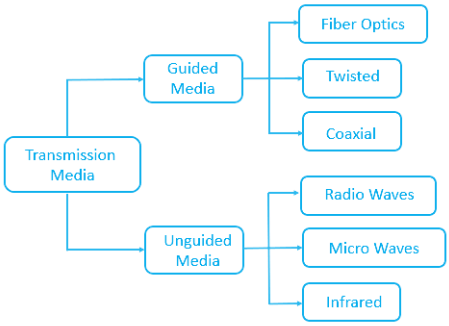
2. Frequency : The number of Complete waves that pass a point in one Second, Measured in Hertz (Hz)

3. Wavelength : The distance between adjacent crests, measured in meters.

4. Phase: A position of a point in time (instant) on a waveform cycle.

Propagation speed in a media : The speed at which a wave propagates through a given medium .The propagation speed also varies from medium to medium depending on the properties of the medium.

Transmission media - A communication channel that carries the information from the sender to the receiver.



Guided / Wired : If the medium used for data transmission is a physical medium, it is called guided or wired because they guide the data from one point to another without altering the frequencies. therefore, data impairments are reduced.

1. **Twisted Pair Cable** - Pairs of twisted copper wire are used for data transmission.

i. Unshielded Twisted Pair (UTP) : Used for telephone connections. These are very flexible and low-priced. However, it is difficult to transmit data for a long distance through UTP wires. suitable for <100M.

ii. Shielded Twisted Pair (STP) : STP is a better quality and secure data transmission medium. However, it is expensive.

Characteristics - Both digital and analog signals can be transmitted, Less cost, Easy to handle, Made of copper cable, Used in telephone lines and computer networks, Twisting reduces cross talk.

2. Coaxial Cable -consists of an electronic cable pair. The outer cable which is like a braided copper net produces electromagnetic field around the central cable. These two cables are separated by a plastic shield. These cables are expensive and used for TV antenna and CCTV.

3. Fiber Optics cable - consists of a pair of an cables. Theres a plastic jacket to separate the two cables. Core is a glass tube and there is glass cladding around it. The data transmission is carried out by while reflecting light. These are used in modern telephone networks. The cable is relatively more expensive.

Unguided/Wireless Media : Data is transmitted as signal through the air without using physical medium.

Radio waves - Data transmission is performed using radio waves. Eg- Mobile phone signals, AM/FM radio signals, Wifi and Bluetooth

Infrared – Used for short distance communications, Slow;less than 10mbps (Eg- Tv remotes, Wireless keyboards and mouse)

Micro Waves – Microwaves travel in a straight line between Transmission centers. These are positioned based on the area's geography. They're used in satellite communication, where satellites 36,000 km above capture and resend data . This method helps Transmit data over long distances and is also used for internet communication.

Properties of signal transmission media

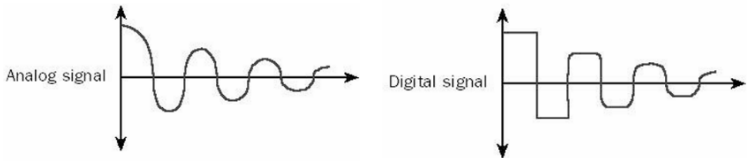
- **Latency :** An expression of how much time it takes for a unit of data To travel from one point to another. usually measured in milliseconds.
- **Bandwidth :** A range of frequencies and measured in Hertz.

Transmission Impairments in Data Communication

When analog signals travel through transmission media, their quality can deteriorate. This means the signal at the start isn't the same as at the end. This imperfection is known as signal impairment.

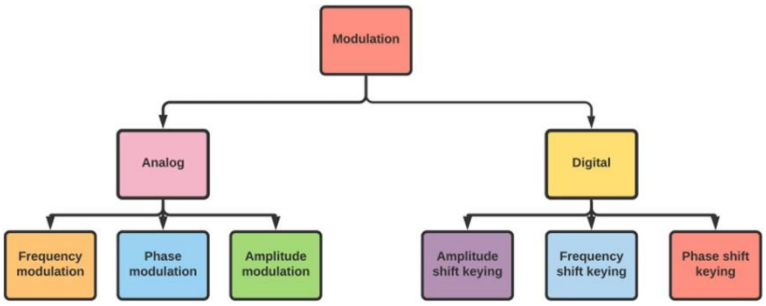
Causes of impairments:

- **Attenuation**: Reduction in signal strength due to increased distance. Amplifiers used to amplify the attenuated signal which gives the original signal back and compensate for this loss.

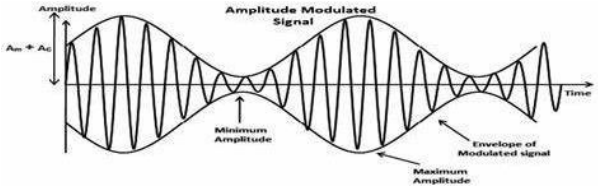


- **Distortion**: Changes in signal shape due to the medium's properties.
- **Noise**: Unwanted signals that interfere with the original signal.
  - **Induced noise**: Caused by motors and appliances acting as antennas.
  - **Thermal noise**: Created by electron movement in wires.
  - **Crosstalk noise**: One wire interfering with a nother.
  - **Impulse noise**: High-energy signals from lightning or power lines.

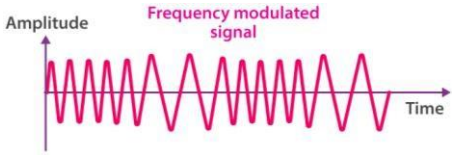
Modulation - Modulation is the technique used to send information by modifying the basic characteristics such as frequency, amplitude and phase, of an electromagnetic signal (modulating signal) by attaching it to a higher frequency signal (carrier signal), producing a modulated signal.



1. **Amplitude Modulation (AM)** : Amplitude of carrier signal varies according to the amplitude of modulating signal. The frequency or phase of the carrier signal remains unchanged.

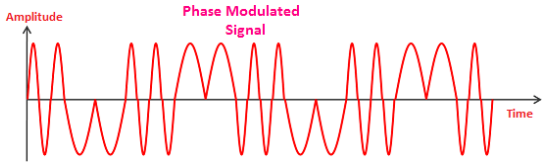


2. **Frequency Modulation (FM)** : The carrier signal frequency changes according to the frequency of the Modulating signal.



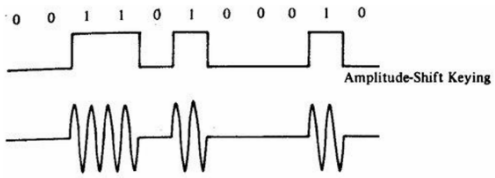
---

3. **Phase Modulation (PM)** : The phase of a carrier signal is modulated in order to reflect the changes in voltage (amplitude) of an analog data signal.



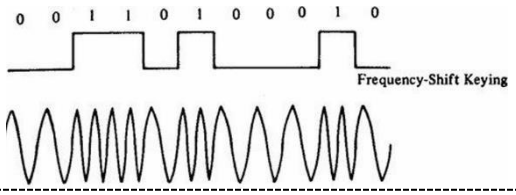
Digital-to-Analog Conversion

1. **Amplitude Shift Keying (ASK) :** The amplitude of an analog carrier signal is modified to reflect binary data. When binary data represents digit 1, the amplitude is held at 1, otherwise it is set to 0. Both frequency and phase remain same as in the original carrier signal .



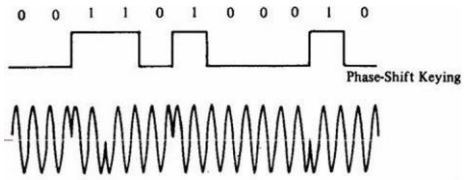
---

2. **Frequency Shift Keying (FSK) :** The frequency of the analog carrier signal is modified to reflect binary data.



---

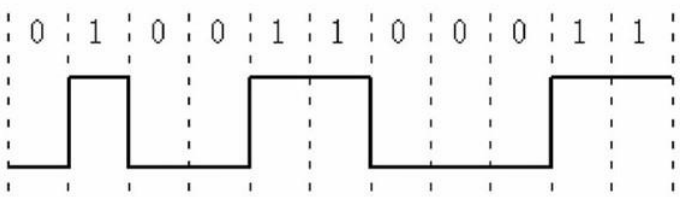
3. **Phase Shift Keying (PSK) :** In this conversion scheme, the phase of the original carrier signal is altered to reflect the binary data.



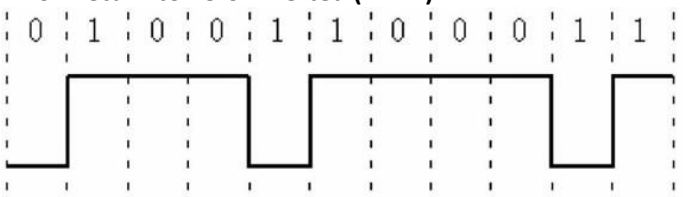
Synchronization ensures that data streams are correctly received and transmitted between devices by using a clock signal to maintain proper timing.

Signal Encoding - The conversion of data into digital signals.

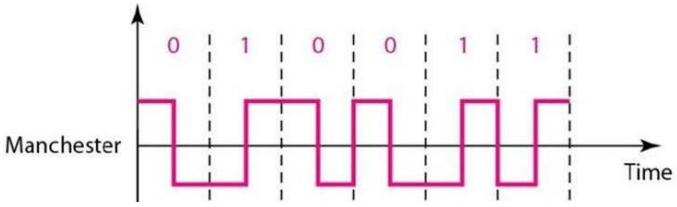
- **Non-return to Zero Level (NRZ-L):**



- **Non-return to Zero Inverted (NRZ-I):**



- **Manchester encoding:**

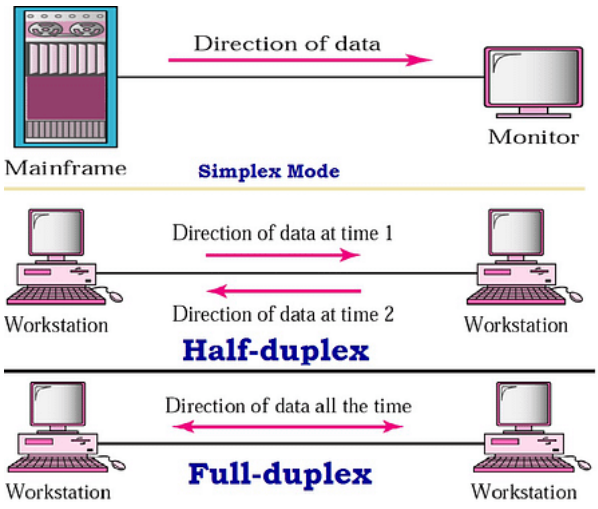


Error Control: Error detection identifies altered data bits during transmission. Error correction and recovery mechanisms fix these errors and restore the actual data bits.

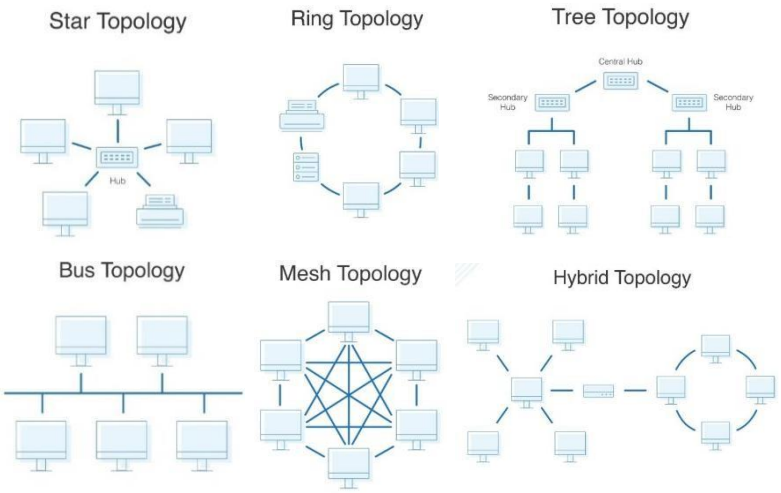
- **Parity Check-** An extra bit of data is added and sent along with the original data bits
  - **Odd Parity –** number of 1s in the data is odd
  - **Even Parity –** Number of 1s in the data is even

Public Switched Telephone network (PSTN) is the global network of circuit- switched telephone systems, operated by various telephony providers. It includes telephone lines, fiber optics, microwaves, cellular networks, satellites, and undersea cables, all linked by switching centers, enabling most telephones to connect.

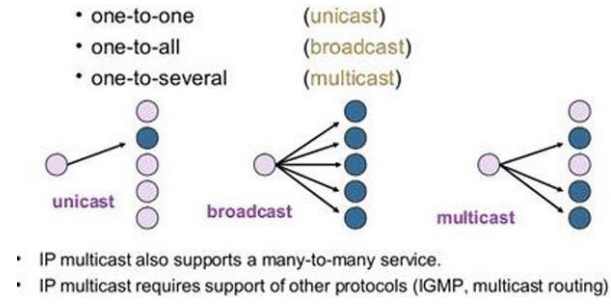
Data Transmission modes



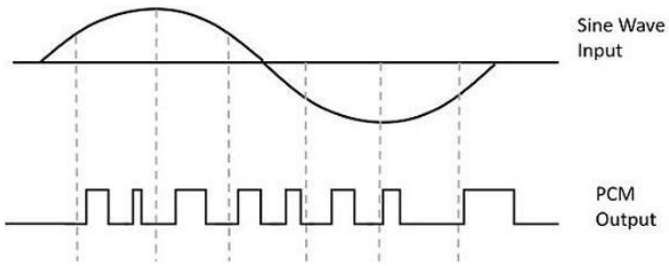
Network Topology - the pattern of connection in designing computer network



Topology	Advantages	Disadvantages
Bus	The easiest network topology for connecting peripherals or computers in a linear fashion. It is easy to connect or remove devices in this network without affecting any other device.	Bus topology is not great for large networks. If a main cable is damaged, whole network fails or splits into two. This network topology is very slow as compared to other topologies.
Ring	In this data flows in one direction which reduces the chance of packet collisions. Equal access to the resources. Speed to transfer the data is very high. Minimum collision. It is cheap to install and expand.	Due to the Uni-directional Ring, a data packet (token) must have to pass through all the nodes. If one workstation shuts down, it affects whole network or if a node goes down entire network goes down.
Star	It is very reliable – if one cable or device fails then all the others will still work.It is high-performing as no data collisions can occur. Easy fault detection because the link are often easily identified.	Requires more cable than a linear bus. If hub goes down everything goes down, none of the devices can work without hub. Extra hardware is required (hubs or switches) which adds to cost.
Mesh	Failure during a single device won't break the network. Adding new devices won't disrupt data transmissions. This topology provides multiple paths to succeed in the destination and tons of redundancy.	It's costly as compared to the opposite network topologies i.e. star, bus, point to point topology. Installation is extremely difficult in the mesh. Maintenance needs are challenging with a mesh.



PCM (pulse code modulation) - It converts an analog signal into digital form.



Network Devices

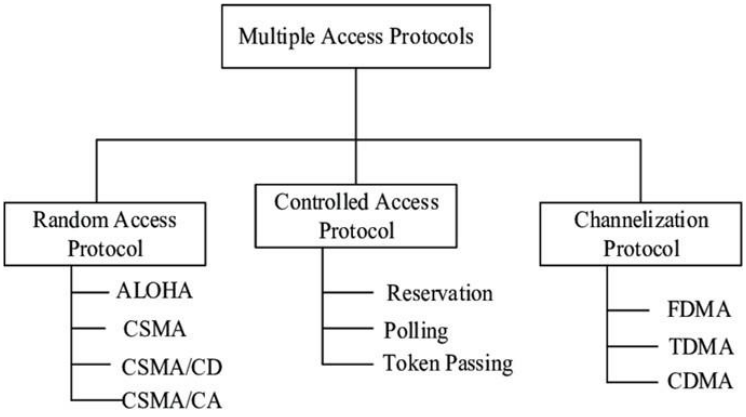
- **Modem** - A modem connects computers to the internet by Converting digital signals to analog and vice versa (Modulation and Demodulation) . Types include internal, external, and wireless modems, with routers often having built-in modems.
- **Repeater** – Receives weak signals C retransmits signals at a higher level or power.
- **Network Interface Card(NIC)** – Hardware component that connects a computer to a network.
- A **hub** receives data in one of its incoming connections and then shall forward the data to all of its outgoing connection
- A **switch** receives data in one of its incoming connections and forwards the data only on the outgoing connection which connects to the destination device.
- **Bridge** – Has 2 ports each attached to different segments of the network.
- **Router** – Connect multiple LANs with WANs , Makes Intelligent decisions on routing a data packet based on IP network address
- A **gateway** is a device that connects different networks and manages traffic flow. It uses multiple protocols, making it more complex than a switch or router. In workplaces, it routes traffic from the main workstation to external networks. At home, it provides internet access.

Types of computer networks

- **Personal Area Network (PAN**) – Within about 10m(e.g.- Bluetooth)
- **Local Area Network(LAN)** – within a limited area (Room or Floor)
- **Metropolitan Area Network(MAN)** – Several LANs, range from 5km to 50km
- **Wide Area Network(WAN)** – expands over large area such as cities or countries
- **Wireless Local Area Network (WLAN)** – WiFi can be main Technology

Protocol – defines how data is formatted and exchanged between devices.

Multiple Access Control Protocols - required to decrease Collision and avoid crosstalk.



In **FDMA**, the bandwidth is divided into frequency bands, each assigned to a specific station permanently.

In **TDMA**, stations share the channel's bandwidth over time. Each station gets a specific time slot to send data, ensuring organized transmission.

CDMA uses different codes for communication, allowing multiple stations to share the entire bandwidth simultaneously, unlike FDMA (which uses separate frequency bands) and TDMA (which assigns time slots).

ALOHA is designed for wireless LAN and shared mediums, allowing multiple stations to transmit data simultaneously. This can lead to collisions and garbled data.

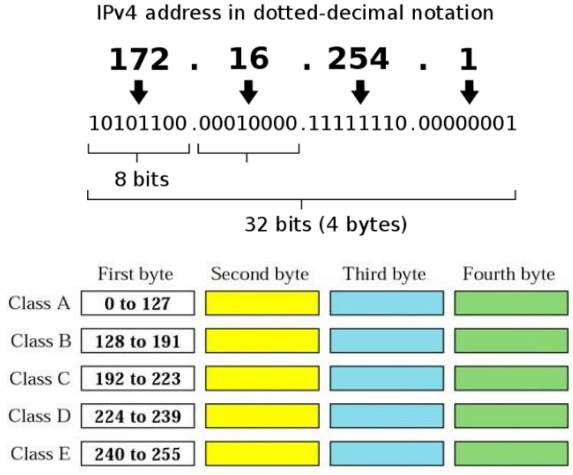
- **Pure Aloha**: Sends data, waits for acknowledgment. If none, waits random time and resends. Reduces collisions.
- **Slotted Aloha**: Sends data at start of time slots. Missed slot? Wait for next. Fewer collisions.

CSMA – Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data.

- **CSMA/CD**: Stations detect collisions and stop transmission if one occurs.
- **CSMA/CA**: Stations avoid collisions by waiting for the channel to be idle before starting transmission.

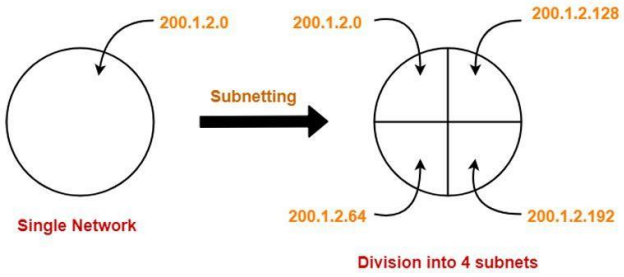
Transmission Control Protocol/Internet Protocol (TCP/IP)

Used to interconnect devices in internet or in ethernet. In TCP/IP, a unique identifier is called an IP address; every machine in a network has a unique IP address.



First IP Address – Network Address
Last IP Address – Broadcast Address

127.0.0.1 – for loopback testing



Classless Inter Domain Routing (CIDR): instead of full class A, B or C networks, organizations can be allocated any number of addresses using this scheme.

Private IPs: Three sets of address ranges are used for private use.

- 10.0.0.0 – 10.255.255.255 (10.0.0.0/8) – 16M addresses
- 172.16.0.0 – 172.31.255.255 (172.16.0.0/12) - 1M addresses
- 192.168.0.0 – 192.168.255.255 (192.168.0.0/16) – 64k addresses

---

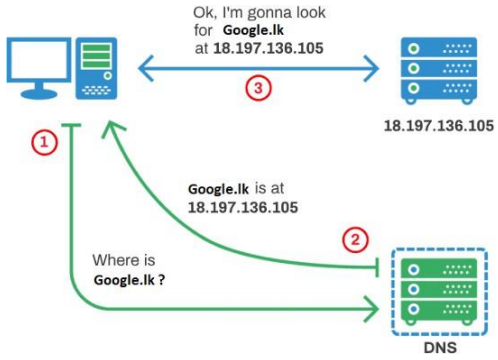
IPv6 - Uses 128-bit addresses (compared to 32-bit addresses of IPv4), offering more IP addresses. Written as eight groups of four hexadecimal digits, separated by colons.

- **Eg**: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

Media Access Control Address (MAC address)

MAC addresses uniquely identify each network interface of a device. They are 48-bit long, divided into 6 blocks, shown as hex numbers separated by colons
(e.g., 4A:8F:3C:4F:9E:3D).

Domain Name System (DNS): DNS provides directory lookup service for given URLs and the web addresses. (translate domain names into IP addresses)



Dynamic Host Configuration Protocol (DHCP) - Assigning IP addresses dynamically on a network. Provides a central database for keeping track of computers that have connected to the network. DHCP simplifies network administration it keeps track of IP address than requiring an administrator to manage the task.

Multiplexing: Combining multiple signals into one signal for transmission over a single medium.

Demultiplexing: Separating the combined signal back into individual signals at the receiving end.

Servers - A server runs specific programs to offer services that other machines (clients) request to perform tasks. This forms a client/server network, providing centralized access to information, resources, and data.

---

i. FILE SERVER - File Transfer Protocol (FTP) is a common server type. It is responsible for transferring files from server to a computer and vice versa. The default port of FTP is 20/21.

---

ii. Proxy servers improve network speeds and save bandwidth by compressing traffic and caching files. They also hide your real IP address from websites, logging the proxy server's IP instead.

---

iii. web server: hosts website files and serves them to web browsers. It loads each file of a web page and displays it as a complete page in the browser.

- **HTTP (Hypertext Transfer Protocol)**: The main protocol for data on the web, linking text with hyperlinks. Default port is 80, and secured version (HTTPS) uses port 443.

- **SSH (Secure Shell)**: The main method for securely managing network devices at the command level, replacing Telnet. Default port is 22.

- **Telnet**: A protocol used to remotely access and manage network devices over the internet, but it's not secure.

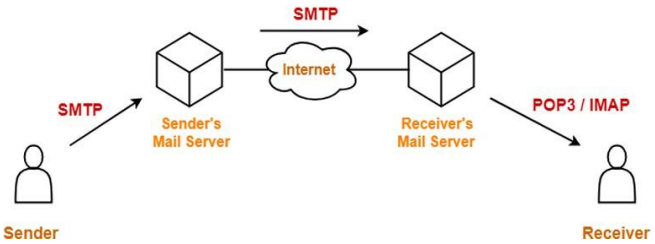
---

iv. MAIL SERVER - crucial for sending, receiving, and storing emails on networks and the internet, using standard email protocols.

- **SMTP(Simple Mail Transfer Protocol)** is a protocol used to send emails between servers and from users to mail systems. Default port is 25

- **pop3 (Post Office Protocol version 3)** A protocol to retrieve emails from the internet. It lets you download emails from the server and then deletes them from the server. Default port is 110

- **IMAP**: A protocol to retrieve emails from a server, but unlike POP3, it keeps the emails on the server. Default port is 143.



---

v. Printer server: Accepts print jobs connected in a network and send to printers(print spooling)

Network Models

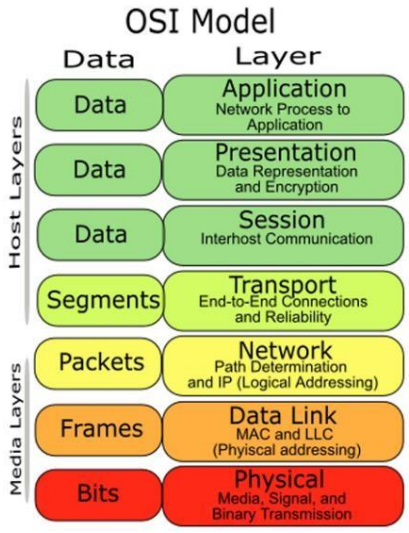
TCP/IP Model	OSI Model	Protocols
Application layer	Application layer	FTP, HTTP, Telnet
	Presentation layer	JPEG, MPEG
	Session layer	NFS, SQL,PAP
Transport layer	Transport layer	TCP, UDP
Internet layer	Network layer	IPv4, IPv6
Network Access Layer	Data Link Layer	ARP, CDP, STP
	Physical layer	Ethernet, Wi-Fi

TCP/IP Model (Transmission Control Protocol/Internet Protocol)

- **Network Access Layer:** The lowest layer; delivers data to devices, transmits IP datagrams, encapsulates into frames, and maps IP to physical addresses.
- **Internet Layer:** Manages network connections, addresses, and data delivery. Handled by Internet Protocol (IP).
- **Transport Layer:** Deals with data transmission using TCP and UDP.
- **Application Layer:** Provides network services to applications, using protocols like HTTP, SMTP, and FTP.

OSI model (Open Systems Interconnection Model)

The OSI model is a framework that defines the functions of a networking system into universal rules, ensuring different products and software can work together seamlessly. It helps achieve interoperability.



7. **Application Layer**: Interfaces with end-user and applications, using protocols like HTTP, SMTP, FTP, DNS, DHCP, Telnet, POP3
6. **Presentation Layer**: Handles data presentation, encoding, and encryption, with formats like JPEG, MP3.
5. **Session Layer**: Manages communication sessions, establishing, maintaining, and terminating connections.
4. **Transport Layer**: Controls data flow, error detection, and correction. Key protocols are TCP, UDP.
3. **Network Layer**: Routes data packets between devices in Different networks, using IP addresses. Manages packet forwarding.
2. **Data Link Layer**: Connects devices within the same physical network, breaking data into frames. Handles flow control and error detection.
   2 sub layers:
   i. MAC – Media Access Layer →MAC addresses
   ii. LLC – Logical Link layer → frame synchronization, frame control C error checking
1. **Physical Layer**: Deals with physical components like cables and radios, delivering raw data between devices.

ICMP -Internet Control Message Protocol – Used by devices such as routers to send error messages (For reliable C ordered delivery of data)

TCP vs UDP	
TCP is a connection-oriented protocol	UDP is a connectionless protocol
TCP is comparatively slower than UDP	UDP is much faster
Can guarantee delivery of data	Cannot guarantee delivery of data
Does not support Broadcasting	Does support Broadcasting
Packets arrive in order at the receiver.	There is no sequencing of data in UDP.
Used by HTTPS, HTTP, SMTP, POP, FTP, etc	Video conferencing, streaming, DNS, VoIP, etc

Network Commands

- **ipconfig**: Contains IP address, network mask, default gateway for all physical C virtual network adapters
- **ipconfig /all**: Contains IP address, Ethernet MAC address, DNS settings, DHCP server info, host name, default gateway for all physical C virtual network adapters
- **nslookup**: Finds the IP address associated with a domain name.
- **tracert**: (Trace route) Display all the routers between source to destination
- **netstat**:(Network Statistics) Lists active network connections and ports.
- **ping**: Used to test reachability of a host on an IP network, measure round trip time from sending messages to destination and back again

Threats

- **Malware:** Software designed with harmful intentions.(virus, trojans, warms etc..)
- **Viruses:** Programs that enter your system and perform harmful activities without your knowledge.(Boot sector viruses, macro viruses, logic bomb, time bomb)
- **Trojans:** Appears as a useful code, but does something harmful when executed
- **Phishing:** Attempts to steal sensitive information by pretending to be a trustworthy entity.
- **Spyware**: spying and stealing personal information
- **Adware**: Ads displayed without user permission
- **Spoofing**: A technique where an attacker impersonates another device or user by falsifying data to gain unauthorized access to a system.

- **Information Disclosure**: The unauthorized release of confidential information, often due to poor security controls or vulnerabilities.
- **Denial of Service (DoS)**: An attack that overwhelms a system, network, or service, rendering it unavailable to legitimate users.
- **Port Scan**: A method of probing networked devices to Identify open ports, which can reveal potential vulnerabilities to exploit.
- **Espionage**: The act of spying or gathering confidential information, typically by governments or organizations, often for political or competitive advantage.
- **Eavesdropping**: Intercepting private communications or data transfers, often covertly, to gain sensitive information.
- **Man-in-the-Middle (MitM)**: An attack where an adversary secretly intercepts and possibly alters the communication between two parties to exploit or manipulate the data being exchanged.

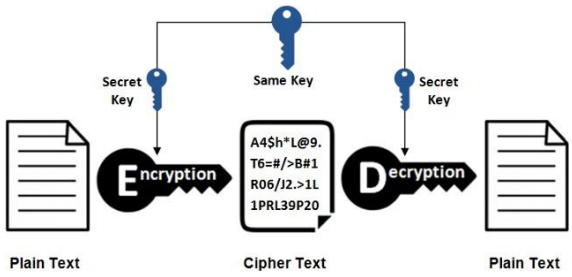
Protection against unauthorized malicious accesses

- **Firewalls:** Devices that monitor and filter network traffic, acting as barriers between private networks and the internet.
- **Antivirus Software:** Programs that detect and isolate malicious software trying to harm a computer.
- **User Education:** Essential for protecting devices from attacks. Choose strong passwords and regularly update antivirus software to safeguard systems.

Network Security

Encryption: A method in cryptography ensuring data confidentiality during transmission.

- **Symmetric Key Encryption**:



- **Asymmetric Key Encryption**

A **digital signature** is like a signature on a paper document — it shows the receiver that the content is trustworthy. For digital documents, a digital signature verifies the source, author, date, and time, confirming that the message content is authentic and hasn't been changed.

ISPs: An Internet service provider (ISP) is an organization that provides services to accessing and using the Internet services.

Dialup connection – Uses traditional telephone lines, and an analog modem. Can't make calls when connected to internet and vice versa.

Advantages of DSL

- **Independent Services**: Losing internet doesn't mean losing phone service. With cable, a failure can knock out all services.
- **Security**: DSL keeps each user separate, unlike some cable networks where users share the network, risking security.
- **Compatibility**: DSL connects easily with other network tech, making remote work setups easier.

Advantages of ADSL

- **Lower Cost**: ISPs often offer high-speed ADSL at affordable rates, usually with a static IP.
- **Flexible Setup**: Engineers can adjust VPN connections quickly to solve network issues.
- **Fast Internet**: ADSL provides high-speed browsing, streaming, and large file downloads, much faster than dial-up.

Network Address Translation

typically found in routers and firewalls. These devices use NAT to convert private IP addresses to a public IP address

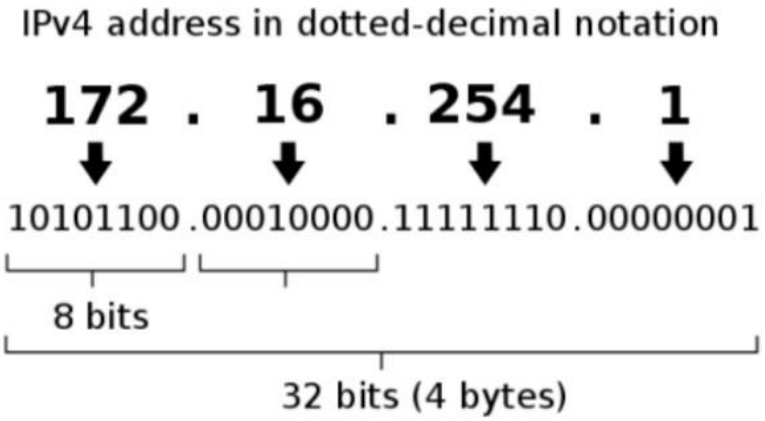
IP ADDRESS & SUBNETTING

What is an IP Address?

An Internet Protocol (IP) address is the unique identifying number assigned to every device connected to the internet.

How IP Address is written?

- For human convenience the IP address is written in dotted decimal notation.
- The 32-bit address is divided into 4 groups of 8 bits (an octet).
- Each octet is written as a decimal number ranging from 0 to 255.
- The decimal numbers are separated by dots.



What are Classes of IP address?

Class	Range			No. of Addresses
A	**0.0.0.0**	**-**	**126.255.255.255**	**16,777,214 (2^{24})**
B	**128.0.0.0**	**-**	**191.255.255.255**	**65,536 (2^{16})**
C	**192.0.0.0**	**-**	**223.255.255.255**	**256 (2^{8})**
D	**224.0.0.0**	**-**	**239.255.255.255**	**-**
E	**240.0.0.0**	**-**	**255.255.255.255**	**-**

What is the problem with Classful IP addresses?

Imagine you run a Company and you have 500 devices in your Company. And you need each of your devices to be connected to Internet. To do so, you need IP addresses for each device. So, if you buy a Class C IP address it won't be sufficient to connect 500 devices. Therefore, you have to go for a Class B address.

In Class B, there are 65,536 addresses in which you will be using only 500 addresses. Which result in waste of 65,000 addresses. As a Solution to this **Classless IP** address is Used.

Classless Inter Domain Routing (CIDR)

Instead of full class A, B or C networks, organizations can be allocated any number of addresses using this scheme. CIDR introduced a new method of representation for IP addresses, now commonly known as CIDR notation, in which an address or routing prefix is written with a suffix indicating the number of bits of the prefix, such as 192.0. 2.0/24 for IPv4.

How to find the Number of Addresses in a CIDR notation?

HOST BITS = 32 – CIDR NOTATION

$2^{\text{HOST BITS}}$ = No. of Addresses

What is the No. of Addresses that can be allocated with the IP Address 192.168.10.0/25?

Host Bits = 32 – 25 = 7

2^7 = 128 addresses

Therefore, With the IP address 192.168.10.0/25…128 IP addresses can be Received.

SUBNETTING

Sub-netting is a technique used to overcome the problem of depletion of network address of a 32-bit addressing scheme. In sub-netting each physical network is assigned 32-bit address mask, which is used to identify networks among other networks. All machines in the subnet should have the same subnet mask.

Simply,
Imagine your company has 200 Computers and it has four departments such as Accounting, IT, Marketing and Production. And you need to give 50 computers for each department. So that it is required to give IP addresses for each computer in each Department.
Now what you do is You get one IP address which can be allocated for 200 addresses (ex – 192.168.16.0 / 24) And then you Divide this address into 4 groups (256/4 = 64) and allocate proportionally for each department.

What is FLSM and VLSM in Subnetting?
In the Process of Subnetting, If the Division of IP addresses is done equally for all departments (SUBNET) then it is **FLSM (Fixed Length Subnet Mask).**

Example – Each department is allocated with 64 addresses.

In the Process of Subnetting, If the Division of IP addresses is Unequally according to requirements of each Departments (SUBNET) then it is **VLSM (Variable Length Subnet Mask).**

Example –

Accounting:	128 addresses
IT:	64 addresses
Marketing:	32 addresses
Production:	32 addresses

What is a Subnet Mask?
A subnet mask is a number that distinguishes the network address and the host address within an IP address.

A Subnet Mask looks like following:

255.255.255.0

There are default Subnet Mask for each Classes of IP addresses

Class A – 255.0.0.0

Class B – 255.255.0.0

Class C – 255.255.255.0

As Each Class has its default Subnet Mask, there is no need to Find it.

But what if the IP address given is Classless. You can find the Subnet Mask using CIDR notation or using No. of addresses.

> **Last Octet of Subnet Mask = 256 – No. of Addresses**
>
> **(Note: This method works only for addresses with 256 or less and CIDR /24 +)**

What is the Subnet Mask of an IP address with 128 addresses?

Last octet of Subnet Mask = 256 – 128 = **128**
Therefore, SNM of an IP address with 128 addresses is **255.255.255.128**

What is the Subnet Mask of an IP address with CIDR notation /25?

In this case, First Find the No. of Addresses first and then find the Subnet Mask

Host bits = 32 – 25 = 7
No. of addresses = 2^7 = 128
Last octet of Subnet mask = 256 -128 = 128
Therefore, SNM of an IP address with CIDR /25 is **255.255.255.128**

How to Find the No. of addresses of an IP address if the Subnet Mask of the IP address is Given?

256 - last octet of SNM = no. of addresses

What is the No. of addresses of An IP address with a Subnet Mask of 255.255.255.192?

No. of addresses = 256 – 192 = 64
Therefore, This IP has 64 addresses.

How to Find the Subnet mask of an IP address with a CIDR notation range from /16 to /23?

1. Find the Host Bits using the CIDR Notation as Usual:
   Ex - CIDR = /23
   Host Bits = 32 – 23 = 9

2. Write the Host Bit as a Power of 2.
   Host Bits = 9
   Represent as = 2^9

3. Deduct 2^8 from the above step (2)
   = 2^9 - 2^8= 2^1

4. Get the Answer of the Power in above step (3).
   2^1 = 2

5. Deduct the answer in the above step (4) from 256, And insert the Value into the Third Octet of Subnet Mask.
   Third Octet of Subnet Mask = 256 - 2
   = 254

   Therefore, Your Subnet Mask for an IP address with a CIDR Notation /23 is 255.255.254.0

How to Find the Subnet mask of an IP address with more than 256 addresses?

1. Represent the No. of Addresses as a power of 2.
   No. of Addresses = 1024
   Represent as = 2^{10}

2. Deduct 2^8 from the above step (1)
   = 2^{10} - 2^8
   = 2^2

3. Get the Answer of the Power in above step (2).
   2^2 = 4

4. Deduct the answer in the above step (3) from 256, And insert the Value into the Third Octet of Subnet Mask.
   Third Octet of Subnet Mask = 256 - 4
   = 252 \

Therefore, Your Subnet Mask for an IP address with a CIDR Notation /23 is 255.255.252.0

How to Find the No. of Addresses of a Subnet Mask, where the Change Happens in Third Octet

1. Deduct the Change in Third octet of Subnet Mask from 256.

   Subnet Mask = 255.255.248.0
   Change in Third Octet = 248
   = 256 - 248
   = 8

2. Represent the above answer in step (1) as a Power of 2.

   $8 = 2^3$

3. Add 2^8 to the answer in the above step (2).

   $= 2^3 + 2^8$
   $= 2^{11}$

4. Get the Answer of the above step (3) and that will be your No. of Addresses.

   $2^{11} = 2048$

Therefore, there are 2048 Addresses for an IP address with a Subnet Mask 255.255.248.0

Unit-7(Information System)

System
orderly grouping of interrelated and interdependent components linked together according to a plan to achieve a specific objective.

Types of Systems

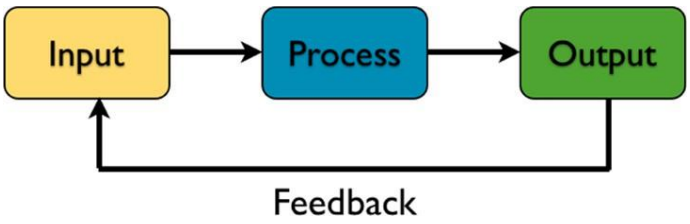
1. Open System
- Take inputs from the environment.
- Gives output to the environment.

2. Closed system
- It doesn't interact with its environment.
- It doesn't take input or give output to the environment.

Transportation System of a Country ← **Manmade System**
Human Blood Circulation System ← **Natural System**

Solar System ← **Nonliving System**
Human Respiratory System ← **Living System**

What is an Information system?
an integrated set of components for Collecting, Storing, Processing and Communicating Information.



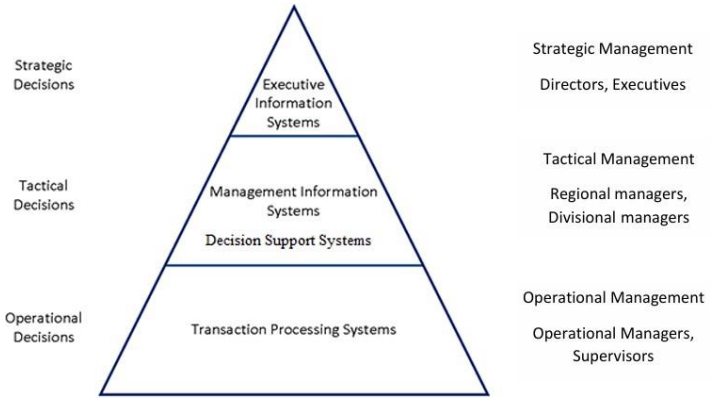
Types of Information System

Transaction Processing Systems (TPS)
used to perform and record the daily routine transactions necessary to conduct a business and serve the operational level users of an organization. Used to make structured decisions
Eg: payroll, Retail point of sale system

Management Information Systems (MIS)
serve the functions of planning, controlling, and decision making by providing routine summary and exception reports for the management level users of an organization.
Eg: Inventory control System

Decision Support Systems (DSS)
combine data and sophisticated analytical models or data analysis tools to support semi-structured and unstructured decision making for the management level users
Eg: Market analysis systems

Executive Support Systems (ESS)
Address unstructured decision making through advanced graphics and communications for the strategic level users of an organization. **Eg**: Financial Planning Systems.



Structure of Decisions	Examples
Structured decisions • Repetitive and routine decisions with predefined procedures to follow	• Restock Inventory • Determine special offers to customers
Semi-structured decisions • Have some agreement on the data, process, and/or evaluation to be used with some level of human judgment	• Designing a marketing plan • Allocate resources to managers
Unstructured decisions • Non-routine. Require insight based on many sources of information and personal experiences.	• Decide entrance or exit from a market • Decide long term objectives

Office Automation Systems (OAS)
designed to increase the productivity of employees through automating information gathering, communication, and presentation processes. **Eg**: Work scheduling systems

---

Geographic Information Systems (GIS)
connect data with geography. GIS allow to map, model and analyze large quantities of data within a single database according to their location. **Eg**: Web based map services, Urban Planning and Transportation Planning applications

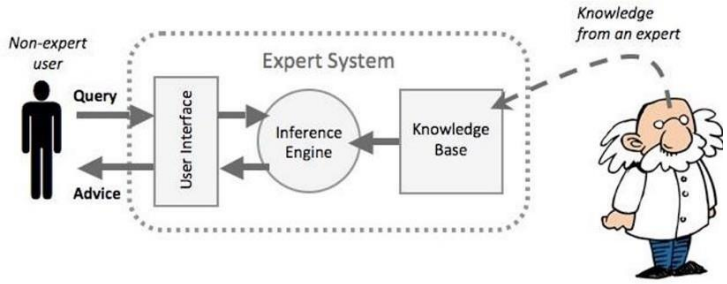
---

Knowledge Management Systems (KMS)
comprise a range of practices used in an organization for acquiring, creating, storing, distributing, applying, integrating knowledge.
Eg: Knowledge work systems (KWS), Customer Feedback systems

Content Management Systems (CMS)
support the creation and modification of digital content. It supports to multiple users working in a collaborative environment.
Eg: web-based publishing(WordPress), format management, history editing and version control.

Enterprise Resource Planning (ERP)
allow organizations to use integrated applications to manage the businesses. ERP software integrates product planning, development, manufacturing, sales and marketing. **Eg**: SAP ERP

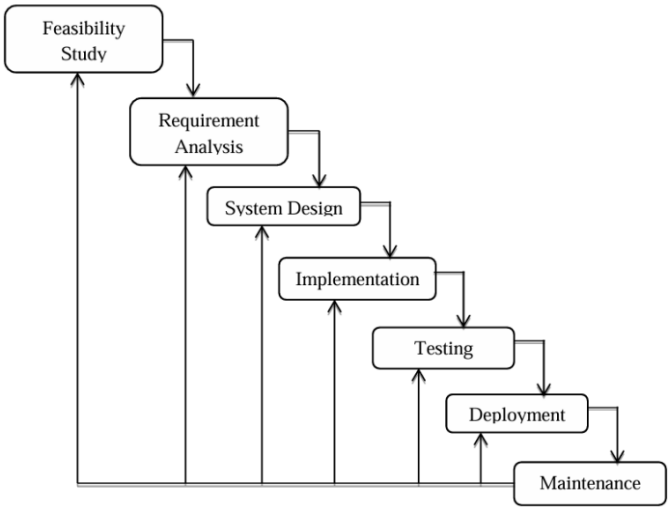
Expert Systems(ES)
emulate the decision-making ability of a human expert. To design an expert system, one needs a knowledge engineer, an individual who studies how human experts make decisions and translates the rules into terms that a computer can understand. **Eg:** MYCIN



Smart Systems
make decisions based on the available data in a predictive or adaptive manner, by means of sensing, actuating and controlling. **Eg**: Anti-lock Braking System (ABS)

System Development Lifecycle models

Waterfall Model

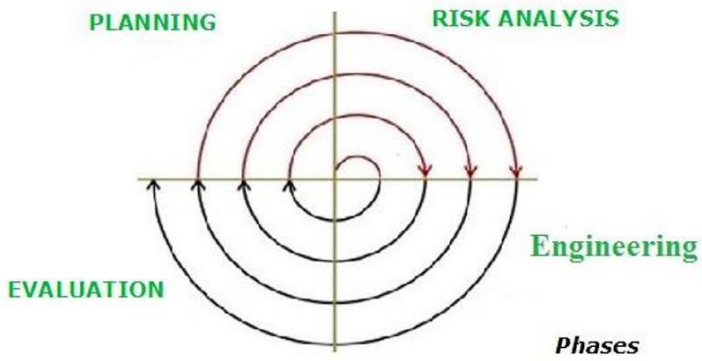


More suitable – when func. Requirements clear and stable, To develop short term simple systems,

Advantages - Easy to manage and control, Every phase has a defined start and end point, Phases are processed and completed one at a time, Works well for smaller projects where requirements are very well understood and stable.

Disadvantages - difficult to respond to changing customer Requirements, No working software is produced until late during the life cycle, High amounts of risk and uncertainty, Poor model for complex and ongoing projects.

Spiral Model



A combination of iterative development and sequential linear development processes.

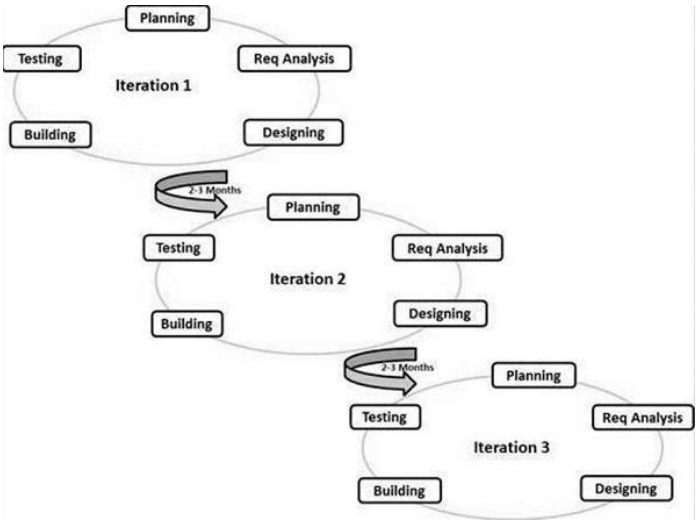
Applicability – For systems need continuous risk evaluation, for expensive complicated projects, projects with continuous changes

Advantages - Fast development, more features can be added during the process, proper risk evaluation is involved

Disadvantages - Not suitable for small projects, spiral may go infinitely, cant understand the final product in the beginning, Management being complex, Costly

---

Agile Model



Applicability – When frequent changes are required, if continuous customer attraction is required, uses the existing methods and modifies them to suit the project requirements.

Advantages – Improved quality, Better control, high consumer satisfaction, reduced risk, Increased flexibility

Disadvantages – Can't predict effort in the beginning, difficult to measure the progress, difficult to implement, may easily fall out of track

Prototype Model

• A model is created before developed for demonstration.
Models – Throw-away Model, Evolutionary Model, Incremental Model

Advantages- active user involvement, better understanding About the system in the beginning, Quicker user feedbacks

Disadvantages- risks due to lesser req analysis, confusions may occur between prototypes and final products.

Rapid Application Development (RAD)

• Minimal planning C more focused on development
Phases

1. Business Modelling
2. Data Modelling
3. Process Modelling
4. Application Generation
5. Testing C Turnover

Advantages - Reduced development time, Encourages customer feedback, suitable for Changing requirements. Integration from very beginning solves a lot of integration issues.

Disadvantages -Only modularized systems can be built using RAD, Requires highly skilled developers/designers for modeling, High dependency on modeling skills.

System development methodologies

i. Structured methodology - Provides a framework (structure) with a set of well-defined guidelines through steps of tasks

ii. Object Oriented methodology - Models system as a collection of objects that work interactively.

Structured System Analysis s Design Methodology (SSADM)

involves study the present system and sketches a blueprint to develop a new system or to modify the existing system.

Parts of a SSADM - Modules, Stages, Steps, Tasks

Modules of SSADM

• **Feasibility Study** - The business area is analyzed.

• **Requirements Analysis -** The requirements of the system to be developed are identified.

• **Requirements Specification** - Detailed func C non-func requirements are defined and new techniques are introduced to define the required processing and data storage.

• **Logical System Specification** - System to be developed is specified logically

• **Physical Design** - Logical design is transformed into a physical design taking technical constraints into consideration.

SSADM

Module 0 - Feasibility Study
Module 1 - Requirements Analysis
Module 2 - Requirement Specification
Module 3 - Logical System Specification
Module 4 - Physical Design

Advantages - More focus on analysis and design, Better quality system specification and documentation, Effective communication and user involvement, Flexible continuity when staff changes, Improved management control

---

Preliminary Investigation - focuses on recognizing the need for a new system and reaching a clear initial picture of what the physical system actually is.

Preliminary investigation is done in two phases.
1. Problem definition 2. Feasibility study

Problem definition - A preliminary survey of the system is carried out to identify the scope of the system.

Feasibility study - means whether the development of a new or improved system is practical and beneficial.

i. Technical feasibility - evaluates whether the developers have ability to construct the proposed system. whether the technology needed exists? how difficult system will be to develop? whether the developers have enough experience using that technology?

---

ii. Economic feasibility - studies cost and benefits to evaluate whether the benefits justify the investments in the system development.

---

iii. Operational feasibility - assesses the willingness and ability of the users to support and use the proposed system. Will the system be useful?

---

iv. Organization feasibility - determines the extent to which the proposed system supports the objectives of the organization's strategy

• End of the feasibility study → feasibility report is produced

---

Requirement analysis - process of studying and analyzing the user needs to arrive at a definition of the problem domain and system requirements. The main objective is to discover the boundaries of the new system C how system must interact within the new problem domain.

• Methods to identify requirements – Interviews, surveys, Questioners, observations, prototyping, sample document collection

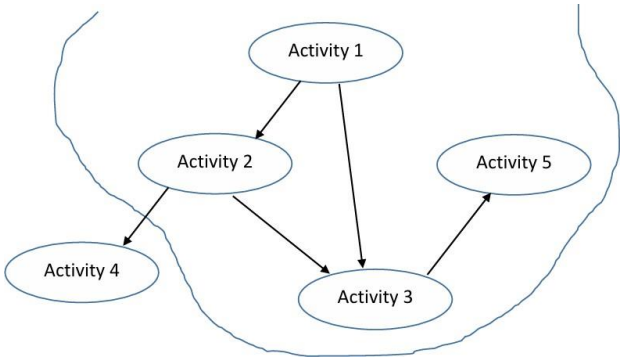
Functional requirements: What activities that the system should carry out.(functionality/behavior of the system)
Non-Functional requirements: How well or with in what limits requirement should be satisfied

Func. Requirements	Non-Func. Requirements
Mandatory	Not mandatory
On user requirements	On user expectations
What product does	How product works
Product features	Product properties

Essential requirements are defined with **"Shall"** whereas nice to have requirements are defined with **"Should"**

Analytical Tools

• **Business Activity Modeling -** A start-off technique for understanding what's going on in a system. It is used to show the business activities that the acts in the system's environment.
Steps – represent activities, link them in flow, mention the system boundaries



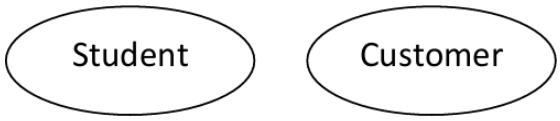
• **Data Flow Modeling (DFM)** - used to model data processing In the system. It is used to define partitions into sub systems. DFM consists of a set of Data Flow Diagrams (DFD).

3 Levels of DFD

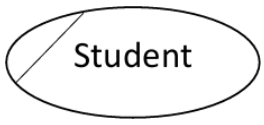
- DFD0 – Context Diagrams
- DFD1 – Level 1 DFD
- DFD2 – Level 2 DFD

Components in DFD

- External Entities



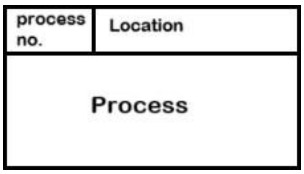
Representing duplicates of external entities



- Data Flows

* data flows between two external entities are shown by **dashed arrows** (---------->)

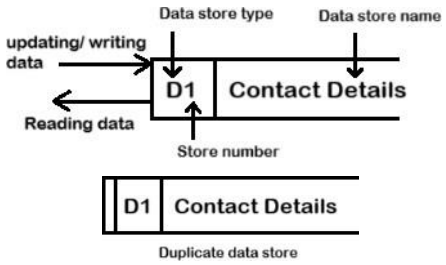
- Processes



- Data Stores

Four types:

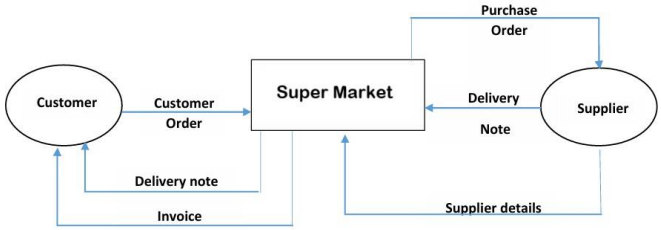
- Computerized (D)
- Temporary data (T) – Ex. temporary program file
- Manual (M) – Ex. filing cabinet
- Manual Temporary T(M) – Ex.in-tray, mail box



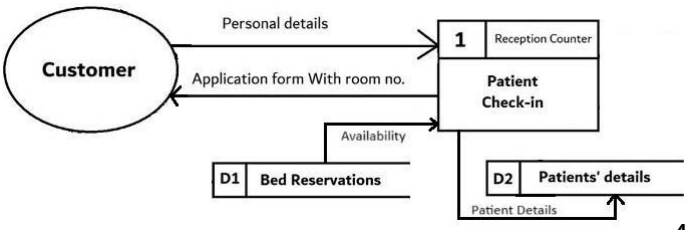
Data flow modelling Rules

- Data can flow directly between:
  - ✓ Two external entities
  - ✓ An external entity and a process
  - ✓ Two processes
  - ✓ A Process and a data store
- A direct data flow cannot exist between:
  - × An external entity and a data store
  - × Two data stores
- Inputs to Data Stores come only from Processes
- Outputs from Data Stores go only to Processes

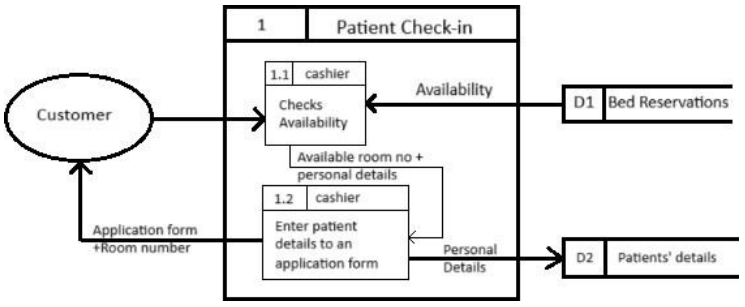
Context Diagram -The highest level of abstraction of the system.



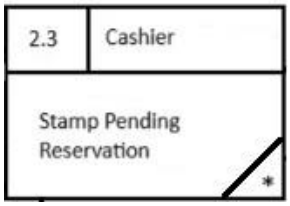
Level 1 DFD – provides a higher-level overview of the system's data processing, shows data movements among the major components of the system.



Level 2 DFD - Further decompose Processes into each and every task.



Elementary Process - A process that doesn't need any further Decomposition, Associates a textual description known as Elementary Process Description (EPD) Mentioned an asterisk (*) At the right below end of a process.



Elementary Process Description - Contains enough details for program specification, Written in plain English or pseudo code

Elementary Process Description
Process Id: 2.2
Process Name: Prepare daily collection summary
Description: Triggered by end of the day routine. First, get relevant payment details from M3 data store. Then prepare daily collection summary and send it to Management

---

Logical Data Modeling (LDM) models the systems data processed by the processes identified in DFM. It consists of a diagram called Logical Data Structure (LDS) and associated textual descriptions.

- M1: Package
  - package name, site, capacity, price

Entity: Package

Attributes: package name, site, capacity, price

Entity Matrix

	Customer	Package	Reservation	Payment
Customer			×	×
Package			×	
Reservation				×
Payment				

- Helps to identify the relationships among entities.
- Provides all possible pairings of entities to identify relationships among them in the interested system.
- Associations between each pair of entities are checked for once.

Business System Options (BSO) - help to select the most appropriate option that satisfies the business requirements.

A BSO consists of,

- A functional description
- A high-level technical description
- Major benefits to the business
- Approximate cost estimate
- Development time scale
- Impact on organization and other existing systems

Ex. BSOs for Camp Reservation system

- BSO 1: Multi-user MIS
  Satisfies all the essential requirements of the business
- BSO 2: Web-based MIS
  Satisfies all the essential requirements plus online advertising, reservations & payments
- BSO 3: Multi-user DSS
  Satisfies all the essential requirements plus decision making support

---

Logical schema Vs Relational schema

Logical Schema	Relational Schema
Entity	Table
Attribute	Field
Instance of an entity	Record of a table
Unique attribute	Primary key

Data Dictionary - an integral part of database, Holds information about the database and the data that it stores (data about data - metadata)

Testing

Test Cases - set of actions executed to verify a particular feature or functionality of a software application.

A test case consists of – test ID, tester name, steps, expected results, actual results etc..

Software testing techniques

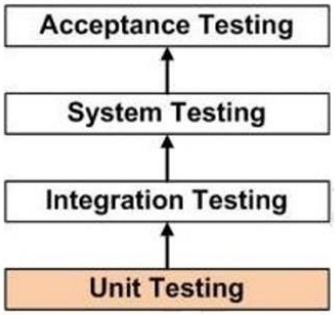
1. Black Box testing - behavior is examined by studying the inputs and outputs. No need technical knowledge. Used in system testing and user acceptance testing

---

2. White Box testing - detailed investigation of internal logic and structure of the code. Requires technical knowledge and used in unit testing and integration testing.

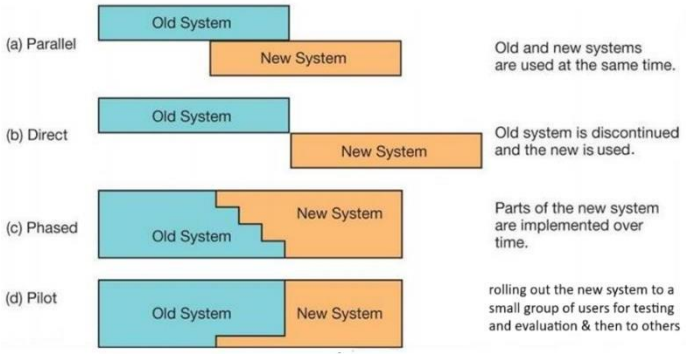
Black Box testing	White Box testing
The internal workings of an application need not be known.	Tester has full knowledge of the internal workings of the application.
Performed by end-users and also by testers and developers.	Performed by testers and developers.
Limited coverage	Maximum coverage is attained
Does not help in optimizing the code	Helps in optimizing the code
Moderately skilled testers can test the application	A skilled tester is needed

Software testing types



- Testing of individual program units **(Unit Testing)**
- Continues with the testing of the integration of these units **(Integration Testing)**
- testing of the system's functionality as a whole **(System Testing)**
- testing to see whether the system is acceptable to the users **(Acceptance Testing)**

Deployment Methods



Maintenance

- changes to hardware, software, and documentation to Support its operational effectiveness.
- Making changes to improve a system's performance, correct errors, deal with security issues, or address new user requirements.

Commercial-Off-The-Shelf (COTS) System

ready-made softwares and are readily available for purchase.

Advantages – Can be implemented in significantly less time. Less resources are required (labor, space C money). more configurable than custom developed software. Availability of wide range of alternatives in the market. More tested by users.

Disadvantages - can be highly complex and usually include many features that will be never used.
- Difficult to gain any competitive advantage from using COTS system as the competitors can also buy and use the same COTS.
- May not meet the expected organizational requirements
- Less contact and support by the developers

Custom-developed Softwares

designed specifically to the requirements and built to operate exactly as needed.

Advantages - Satisfy unique requirements, Can gain real competitive advantage, Can be modified, Can be integrated with existing systems of future systems, Provides all needed functions and none of not needed.

Disadvantages – Large initial Investment, Takes longer time to implement

Unit-8(Database Management)

Data vs Information

Data	Information
Raw, unprocessed facts	Processed and meaningful data
No context or interpretation	Provides context and value
Requires processing	Already processed and useful
Example: "45, 78, 32"	Example: "Average score: 51%"

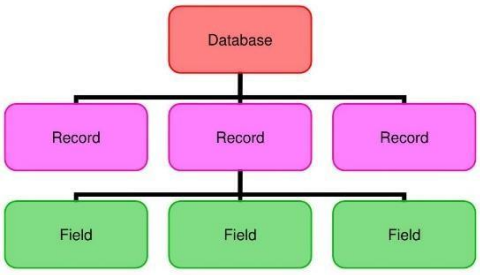
Structured data vs Unstructured data

Structured Data	Unstructured Data
Highly organized and formatted	No predefined format or structure
Easily searchable in databases	Difficult to search and analyze
Stored in rows and columns	Stored in varied formats (e.g., text, images)
Example: Spreadsheets, SQL databases	Example: Emails, social media posts

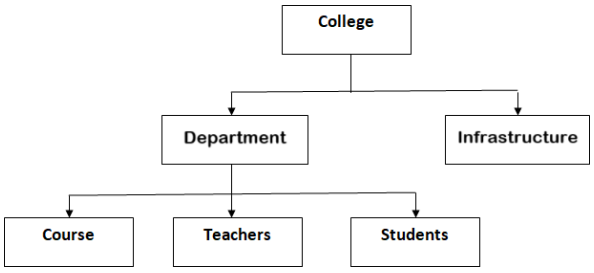
Database: an organized collection of structured data, usually stored on a computer. It is managed by software called a database management system (DBMS).

DBMS Models

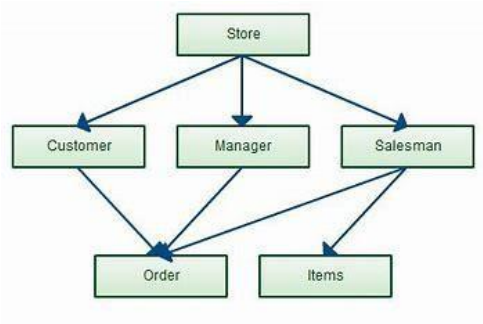
Flat file database -A large collection of data where tables C records are not connected. It have just one table.



Hierarchical model - Data are organized into a tree-like structure. Presents parent: child relationship (1:M)

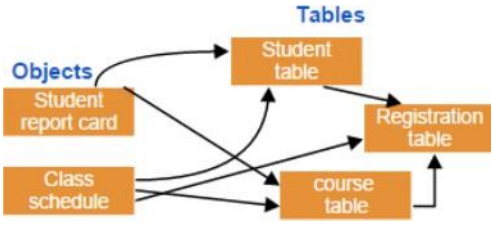


Network Model – Extended version of hierarchical model. Supports more than one parent relationship (M:M)



Relational Model - stores data in the form of relations (tables).

Object relational model - combination of a Object oriented database model and a Relational database model. it supports objects, classes, inheritance etc.



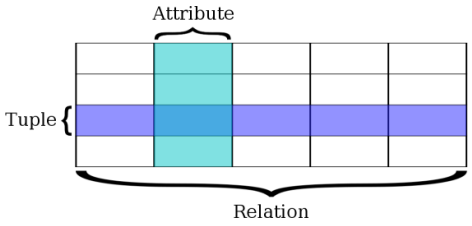
Main components of the relational model

Record/ Tuple: Each row of a table is known as record. It is also known as tuple.

Attributes/Columns : Each column in a Table. Attributes are the properties which define a relation. e.g., Student_Rollno, Name, Address.

Degree :The total number of attributes which in the relation is called the degree of the relation.(Number of columns)

Cardinality :Total number of rows present in the Table



Constraints - rules that ensure data consistency when changes are made to the database:

- **NOT NULL Constraint**: Ensures a column cannot have an empty (null) value.
- **Unique Constraint**: Ensures all values in a column are different.
- **Primary Key Constraint**: Uniquely identifies each record in a table. Must be unique and not null.
- **Foreign Key Constraint**: Links a column in one table to the primary key in another, maintaining relationships.
- **Check Constraint**: Limits the range of values that can be entered in a column.

Keys

Candidate key ← Combination of attributes used to identify a record. One of them becomes the primary key

Primary key ← Uniquely identifies each row in a relation

Alternate key ← Candidate key which is not a primary key.

Composite key ← Primary Key with a combination of 2 or more columns

Foreign key ← provides a link between data in two tables

Dependencies

Functional Dependency ← one or more attributes uniquely identify other attributes in a table.

Full Dependency ← All attributes depend on a single attribute.

Partial Dependency ← A non-prime key functionally depend on a part of candidate key.

Transitive Dependency ← Non-prime attribute fully depend On another non-prime attribute.

Data Consistency – Ensures any data is written to a database Must be valid according to defined rules.

Integrity – Maintains the data accuracy and consistency.
i. Integrity constraint – Primary key, Not null, Unique, Foreign key, check, default

ii. Domian Integrity – Ensures all data in a field are within a valid range. Prevents invalid data being stored.

iii. Entity Integrity – Ensures each row in a table uniquely identified by using the primary key.

iv. Referential Integrity – Ensures value in one table references an existing value in another table.

Data atomicity – ensures that all operations within a data entry are either fully completed or none if one part fails.

Data Anomalies

- **Insertion anomaly** –Inability to add data without having some required existing data
- **Update anomaly** - Issues when changing data in one place but not in others, causing inconsistencies.
- **Deletion anomaly** – When deleting a data, losing some other important data

---

Common errors in a database

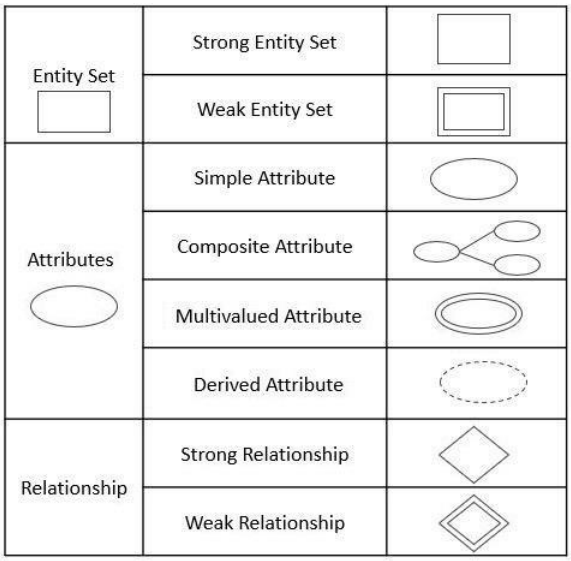
- Data redundancy
- Having null data
- Having data beyond the range
- Data type mismatching

Cardinality/Relationships

- One to one(1:1)
- One to many(1:M)
- Many to many(M:N)

Entity Relationship (ER) Diagrams

ER diagram is a blueprint(plan) we should prepare before creating a physical data storage

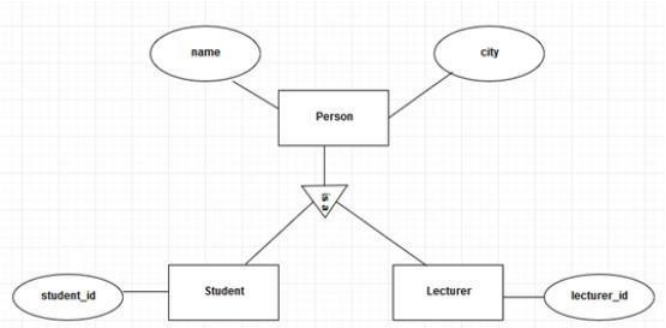


Extended Entity Relationship (EER) diagram is an extention of ER diagrams.

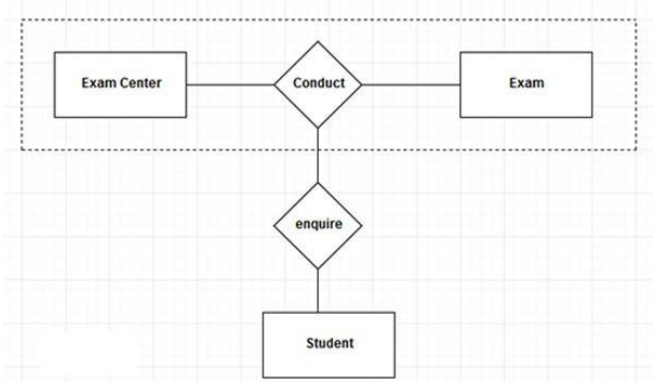
- Generalization
- Specialization
- aggregation.

Generalization- Lower level entities can be combined to produce a higher level entity.(bottom to top approach)

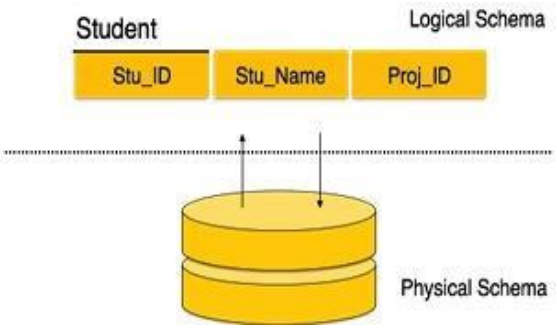
Specialization - high-level entities can be divided into lower level entities. (top to bottom approach)



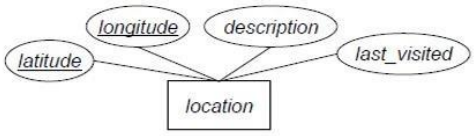
Aggregation - relation between two entities is a treated as a single entity.



Physical schema s Logical schema

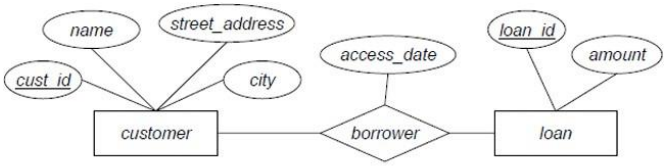


Converting ERD to Logical Schema



location(latitude, longitude, description, last_visited)

M:N ERD to logical schema



customer(cust_id, name, street_address, city)
loan(loan_id, amount)
borrower(cust_id, loan_id, access_date)

Normalization

2 main Purposes:

- Eliminate redundant (useless) data.
- Ensure data dependencies make sense. (data is logically stored)

0NF – Zero Normalization Form

- Extra memory usage
- Anomalies occur

1NF – 1ST Normalization Form

- Removes duplicate records.
- Separate tables for multi-valued attributes.
- Establishes connections using PK C FK

* All the non-key attributes functionally depend on primary key

---

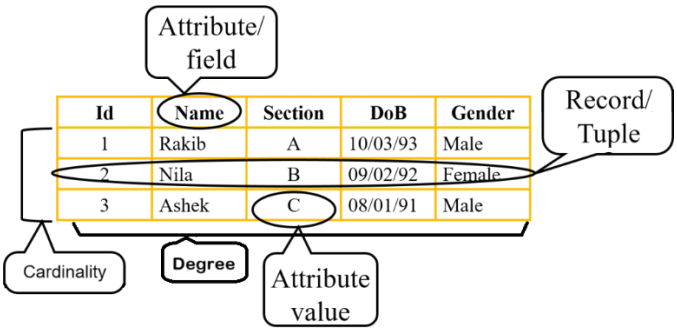
2NF – 2nd Normalization Form

- Removes partial dependency

3NF – 3rd Normalization Form

- Removes Transitive dependency

SQL



Constraints

- NOT NULL Constraint → (st_id int(5) **NOT NULL**)
- Unique Constraint → (st_id int(5) **UNIQUE**)
- Primary key constraint → (st_id int(5) **PRIMARY KEY**)
- foreign key constraint → (**FOREIGN KEY**(st_id) **REFERENCES** Students(st_id))
- check constraint → (Age int(2) **check (Age>=18)**)
- default constraint → (Gender varchar(10) **default** 'Male')

SQL → DDL- **Data Definition Language**
DML- **Data Manipulation Language**
DQL- **Data Query Language**
DCL- **Data Control Language**
TCL- **Transaction Control Language**

Data Types → VARCHAR, CHAR, INT, DATE

DDL Commands

- CREATE
- ALTER
- DROP
- RENAME
- TRUNCATE
- COMMENT

CREATE

- CREATE DATABASE School;
- CREATE TABLE student(
  St_no int(5) PRIMARY KEY,
  Name varchar (20)NOT NULL,
  Sub_id int(3),
  FOREIGN KEY(Sub_id) REFERENCES SUBJECT(Sub_id),
)

ALTER

- ALTER TABLE STUDENT DROP PRIMARY KEY;
- ALTER TABLE ADD Address varchar(30) NOT NULL;
- ALTER TABLE DROP COLUMN Address;
- ALTER TABLE DROP FOREIGN KEY(Sub_id);

DROP

- DROP DATABASE School;
- DROP TABLE Teacher;

DML Commands

- SELECT
- DELETE
- UPDATE
- INSERT

QUERY Commands – Used to carry out tasks
i. SELECT ii. FROM iii. WHERE

- **INSERT INTO Employee**(Empid, Empname, Salary) **Values**(007, "Kasun", "45000") OR

- **INSERT INTO Employee VALUES** (007, "Kasun", "45000");

UPDATE Employee SET Empname= "Kamal" **where** Empid= "007"

DELETE FROM Employee **WHERE** Empid= '007'

Select Statement

- **SELECT** Empid, Salary From Employee
- **(*) -All data → SELECT** * FROM Student (Display all data)

(WHERE) – To Filter Records
SELECT * FROM Employee **WHERE** Department='IT';

Operator	Description	Example
=	Equal	SELECT * FROM Products WHERE Price = 18;
>	Greater than	SELECT * FROM Products WHERE Price > 30;
<	Less than	SELECT * FROM Products WHERE Price < 30;
>=	Greater than or equal	SELECT * FROM Products WHERE Price >= 30;
<=	Less than or equal	SELECT * FROM Products WHERE Price <= 30;
<>	Not equal	SELECT * FROM Products WHERE Price <> 18;
BETWEEN	Between a certain range	
LIKE	Search for a pattern	
IN	To specify multiple possible values for a column	

SELECT * FROM Employee WHERE
Salary **BETWEEN** 50000 **AND** 100000;

SELECT * FROM Employee WHERE
Empname **LIKE** 'a%'**; ← starting with "a"**

SELECT * FROM Employee WHERE
Empname **LIKE** '%an%'**; ← have "an" in the middle**

SELECT * FROM Employee WHERE
Empname **LIKE** '%s'**; ← ends with "s"**

SELECT * FROM Employee WHERE
Empname **LIKE** 'Jo_ _'**; ← starts with 'Jo' and exactly has 2 more characters**

SELECT * FROM Employee WHERE Department
IN ('IT, 'Accounts', 'HR') **← Selects customers from mentioned departments**

SELECT * FROM Employee WHERE
NOT Department='IT'; ← All records except ones from IT

SELECT * FROM Employee WHERE
Department='IT' **AND** Branch='Colombo'; **← select staments where both statements are true**

SELECT * FROM Employee WHERE
Department='IT' **OR** Department= 'HR'; **← select staments where one of the statements are true**

SELECT * FROM employee **ORDER BY** ASC/DESC

Functions

AVG(), SUM(), MAX(), MIN(), COUNT()

SELECT MIN(Salary) FROM Employee;

SELECT SUM(Salary) **as Total** from Employee;

Total

Select Department, sum(Salary) From Employee
GROUP BY Department; ← Group elements

Select Distinct(Department) From Employee;
← Limits Repeated data only to be displayed once

Grouping 2 Tables

SELECT * FROM EMPLOYEE,PROJECT WHERE
EMPLOYEE.EMPNO=PROJECT.EMPNO; ← Display everything in both tables where Empno in both are same

---

SELECT **employee.Empno,** name, SUM(Hours) FROM
Employee,Project WHERE EMPLOYEE.EMPNO=PROJECT.EMPNO;
← If same attributes exists in both tables, should mention from which table we get to query.

Inner Join - selects records that have matching values in both tables
Select Employee.Empno, Salary, Hours From Employee **INNER JOIN Project on** Employee.Empno=Project.Empno;

Multiple Statements

SELECT name FROM city WHERE
country_id IN **(**
SELECT country_id FROM country
WHERE population > 20000000);
←This query finds cities in countries that have a population above 20M:

SQL (Structured Query Language)

What is SQL ?

- SQL stands for Structured Query Language.
- SQL is a standard language for storing, manipulating and retrieving data in databases.
- SQL lets you access and manipulate databases.
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

---

1. CREATE DATABASE

Creates a new database.
Example:
Create Database SchoolDB;

---

2. SHOW DATABASES

Lists all available databases.
Example:
Show Databases;

---

3. USE DATABASE

Selects a database to operate on.
Example:
Use SchoolDB;

---

4. DROP DATABASE

Deletes an existing database and all its contents.
Example:
Drop Database SchoolDB;

---

5. BACKUP DATABASE

Creates a backup of the database.
Example (MySQL):

BACKUP DATABASE customers
To DISK= 'D:\backups\customers.bak';

---

6. CREATE TABLE

Creates a new table with specified columns and data types.
Example:
Create Table Students (StudentID Int Primary Key, FirstName Varchar(50), LastName Varchar(50), BirthDate Date, Email Varchar(100) Unique);

Data types
1. CHAR(Size) - Fixed length of string (letter, number, special characters) Default- 1 (0-255)
2. VARCHAR(Size) - Variable length string (0-65535)
3. Boolean - True/False
4. Int (Size) - Integer
5. Float/double
6. Date
7. Time

---

7. CREATE TABLE Using Another Table

Creates a new table based on the structure or data from an existing table.
Examples:

Create Table customer2 AS
Select customername, Tel.No
From customers;

---

8. SHOW TABLES

Lists all tables within the selected database.
Example:
Show Tables;

---

9. DROP TABLE

Deletes an existing table and all its data.
Example:
Drop Table Students;

---

10. ALTER TABLE

Modifies the structure of an existing table.
Examples:

- **Add a Column:**
  Alter Table Students
  Add Column PhoneNumber Varchar(15);

- **Drop a Column:**
  Alter Table Students
  Drop Column PhoneNumber;

- **Modify a Column:**
  Alter Table Students
  Modify Column Email Varchar(150);

  Alter Table Students
  Change Column Email Varchar(150);

---

11. Commonly Used Constraints

Rules applied to table columns to enforce data integrity.
Examples:

- **PRIMARY KEY:** Uniquely identifies each record.
  StudentID Int Primary Key
- **FOREIGN KEY:** Ensures referential integrity between tables.
  Foreign Key (ClassID) References Classes(ClassID)
- **UNIQUE:** Ensures all values in a column are unique.
  Email Varchar(100) Unique
- **NOT NULL:** Ensures a column cannot have NULL values.
  FirstName Varchar(50) Not Null
- **DEFAULT:** Sets a default value for a column.
  EnrollmentDate Date Default Current_Date

---

12. SQL CHECK Constraint

Ensures that all values in a column satisfy a specific condition.
Example:
Create Table Employees (EmployeeID Int Primary Key, Name Varchar(100), Age Int, Check (Age >= 18));

---

13. INSERT INTO (Two Methods)

Adds new records to a table.
Methods:

- **Method 1: Specifying Columns:**
  Insert Into Students (FirstName, LastName, BirthDate, Email)
  Values ('John', 'Doe', '2000-05-15', 'john.doe@example.com');
- **Method 2: Without Specifying Columns:**
  (Ensure values are in the same order as table columns)
  Insert Into Students
  Values (Null, 'Jane', 'Smith', '1999-08-22', 'jane.smith@example.com');

---

14. UPDATE

Modifies existing records in a table.
Example:
Update Students
Set Email = 'john.newemail@example.com'
Where StudentID = 1;

---

15. DELETE

Removes records from a table.
Example:
Delete From Students
Where StudentID = 2;

---

16. SELECT Statement

Retrieves data from one or more tables.
Examples:

- **Select All Columns:**
  Select * From Students;
- **Select Specific Columns:**
  Select FirstName, LastName, Email From Students;

---

17. SELECT TOP

Retrieves the top N records from a table.
Examples:

Select Top 5 * From Students;

---

18. Operators

Used to specify conditions in SQL statements.
Common Operators:

- **Comparison Operators:** =, !=, <, >, <=, >=
- **Logical Operators:** AND, OR, NOT
- **Range Operators:** BETWEEN, IN, LIKE

Example:
Select * From Students
Where Age >= 18 And (City = 'New York' Or City = 'Los Angeles');

---

19. ORDER BY

Sorts the result set based on one or more columns.
Example:
Select FirstName, LastName, Age
From Students
Order By Age Desc;

---

20. MAX() Function

Returns the maximum value in a specified column.
Example:
Select Max(Age) As OldestAge From Students;

---

21. MIN() Function

Returns the minimum value in a specified column.
Example:
Select Min(Age) As YoungestAge From Students;

---

22. COUNT() Function

Returns the number of rows that match a specified condition.
Example:
Select Count(*) From Students;

---

23. AVG() Function

Returns the average value of a numeric column.
Example:
Select Avg(Age) From Students;

---

24. SUM() Function

Returns the total sum of a numeric column.
Example:
Select Sum(TuitionFee) As TotalRevenue From Students;

---

25. SQL JOIN

Combines rows from two or more tables based on a related column.
Types of Joins:

- **INNER JOIN:** Returns records with matching values in both tables.
- **LEFT (OUTER) JOIN:** Returns all records from the left table and matched records from the right table.
- **RIGHT (OUTER) JOIN:** Returns all records from the right table and matched records from the left table.
- **FULL (OUTER) JOIN:** Returns all records when there is a match in either left or right table.

---

26. SQL UNION Operator

Combines the result sets of two or more SELECT statements, removing duplicates.
Example:
Select Email From Students
Union
Select Email From Alumni;

(Note: To include duplicates, use UNION ALL.)

---

27. SQL Comments

Adds comments to SQL code for documentation purposes. These comments are ignored during execution.
Types:

- **Single-line Comments:**
  -- This is a single-line comment
- **Multi-line Comments:**
  /* This is a
  multi-line comment */

---

28. AUTO INCREMENT Field

Automatically generates a unique number for a column, typically used for primary keys.
Example:
Create Table Courses (CourseID Int Auto_Increment Primary Key, CourseName Varchar(100));

---

29. LIKE Statement

Searches for a specified pattern in a column using wildcard characters.
Wildcards:

- **%** - Represents zero or more characters.
- **_** - Represents a single character.
- [] - Matches any single character within the specified range or set.
- [^] - Matches any single character not within the specified range or set.

Examples:

- **Find names starting with 'Jo':**
  Select * From Students
  Where FirstName Like 'Jo%';
- **Find emails ending with 'example.com':**
  Select * From Students
  Where Email Like '%@example.com';
- **Find names with 'a' as the second character:**
  Select * From Students
  Where FirstName Like '_a%';
- **Find names where the third character is 'n':**
  Select * From Students
  Where LastName Like '__n%';
- **Find names where the second character is either 'a' or 'e':**
  Select * From Students
  Where FirstName Like '_[ae]%';

- **Find names where the first character is a vowel:**
  Select * From Students
  Where FirstName Like '[AEIOU]%';

---

30. DISTINCT

Retrieves unique values from a specified column.
Example:
Select Distinct City From Students;

---

31. GROUP BY

Groups rows that have the same values in specified columns, often used with aggregate functions.
Example:
Select ClassID, Count(*) As NumberOfStudents
From Students
Group By ClassID;

---

32. HAVING

Filters groups based on a specified condition, used in conjunction with GROUP BY.
Example:
Select ClassID, Count(*) As NumberOfStudents*
From Students
Group By ClassID
*Having Count(*) > 30;

---

33. INDEX

Improves the speed of data retrieval operations on a table.
Example:
Create Index idx_lastname On Students (LastName);

---

34. VIEW

Creates a virtual table based on the result-set of an SQL statement.
Example:
Create View StudentEmails As
Select FirstName, LastName, Email
From Students;

---

35. TRANSACTION

Ensures a sequence of SQL operations are executed as a single unit, maintaining data integrity.
Example:
Start Transaction;

Update Accounts Set Balance = Balance - 100 Where AccountID = 1;
Update Accounts Set Balance = Balance + 100 Where AccountID = 2;

Commit;

INNER JOIN

The INNER JOIN keyword returns rows when there is a match in both tables involved in the join. It combines rows from two or more tables based on a related column between them.

How It Works:

- Only returns rows that have matching values in both tables.
- If there are rows in one table that do not have corresponding rows in the other table, those rows are not included in the result.

Example:
Suppose we have two tables: Students and Classes.

- Students Table

StudentID	FirstName	ClassID
1	John	101
2	Jane	102
3	Mike	103

- Classes Table

ClassID	ClassName
101	Math
102	Science
104	History

Query:

```
SELECT Students.FirstName, Classes.ClassName
FROM Students
INNER JOIN Classes ON Students.ClassID = Classes.ClassID;
```

Results:

FirstName	ClassName
John	Math
Jane	Science

LEFT JOIN (or LEFT OUTER JOIN)

The LEFT JOIN keyword returns all rows from the left table (table1), and the matched rows from the right table (table2). The result is NULL from the right side if there is no match.

How It Works:

- Returns all rows from the left table.
- If there is no match in the right table, the result is NULL for columns from the right table.

Example:
Using the same tables:

Query:

```
SELECT Students.FirstName, Classes.ClassName
FROM Students
LEFT JOIN Classes ON Students.ClassID = Classes.ClassID;
```

Result:

FirstName	ClassName
John	Math
Jane	Science
Mike	NULL

RIGHT JOIN (or RIGHT OUTER JOIN)

The RIGHT JOIN keyword returns all rows from the right table (table2), and the matched rows from the left table (table1). The result is NULL from the left side if there is no match.

How It Works:

- Returns all rows from the right table.
- If there is no match in the left table, the result is NULL for columns from the left table.

Example:
Using the same tables.

Query:

```
SELECT Students.FirstName, Classes.ClassName
FROM Students
RIGHT JOIN Classes ON Students.ClassID = Classes.ClassID;
```

Results:

FirstName	ClassName
John	Math
Jane	Science
NULL	History

FULL JOIN (or FULL OUTER JOIN)

Description:
The FULL JOIN keyword returns all rows when there is a match in either table. It combines the results of both LEFT JOIN and RIGHT JOIN.

How It Works:

- Returns all rows from both tables.
- Rows with no match in the other table will have NULL values in the columns from the table with no match.

Example:
Using the same tables.

Query:

```
SELECT Students.FirstName, Classes.ClassName
FROM Students
FULL JOIN Classes ON Students.ClassID = Classes.ClassID;
```

Result:

FirstName	ClassName
John	Math
Jane	Science
Mike	NULL
NULL	History

CROSS JOIN

The CROSS JOIN keyword returns the Cartesian product of the two tables. This means it returns all possible combinations of rows from both tables.

How It Works:

- Combines each row of the first table with each row of the second table.
- The number of rows in the result set is equal to the number of rows in the first table multiplied by the number of rows in the second table.

Example:
Using the same tables.

Result:

FirstName	ClassName
John	Math
John	Science
John	History
Jane	Math
Jane	Science
Jane	History
Mike	Math
Mike	Science
Mike	History

Unit-9(Programming)

Arithmetic Operators

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

Assignment Operators

Operator	Example	Same as
=	c= 5	c = 5
+=	m += 3	m = m + 3
-=	m -= 3	m = m - 3
*=	m *= 3	m = m * 3
/=	m /= 3	m = m / 3
%=	m %= 3	m = m % 3
**=	m **= 3	m = m ** 3
//=	m //= 3	m = m // 3

Comparison Operators

Operator	Name	Example
==	Equal	a == b
!=	Not equal	a != b
>	Greater than	a > b
<	Less than	a < b
>=	Greater than or equal to	a >= b
<=	Less than or equal to	a <= b

Logical Operators

Operator	Description	Example
and	Returns True if both statements are true	a < 5 and b< 10
Or	Returns True if one of the statements is true	a < 5 or b < 4
not	Reverse the result, returns False if the result is true	not(a < 5 and b< 10)

Bitwise Operators

Operator	Name	Description	Example
&	AND	Sets each bit to 1 if both bits are 1	(a & b) =12 (means 0000 1100)
|	OR	Sets each bit to 1 if one of two bits is 1	(a | b) = 61 (means 0011 1101)
^	XOR	Sets each bit to 1 if odd number of bits are 1	(a^b) = 49 (means 110001)
~	NOT	Inverts all the bits	(~a) = -61 (means 1100 0011 in 2's complement)

Operators Precedence

Operator	Description
()	Parentheses
**	Exponent (raise to the power)
~	Bitwise NOT
*, /, %, //	Multiplication, Division, Modulus and Floor Division
+, –	Addition and Subtraction
>>, <<	Bitwise Right Shift and Bitwise Left Shift
&	Bitwise AND
^, |	Bitwise XOR and OR
<=, <, >, >=	Comparison Operators
==, !=	Equality Operators
not, or, and	Logical Operators

Input() function

- name = input("Enter your name: ") ← String by default
- age = int(input("Enter your age: ")) ← data type is saved as an integer value

---

Control structures

- Sequence
- Selection – if, if-else, if-elif -else
- Iteration – while, for

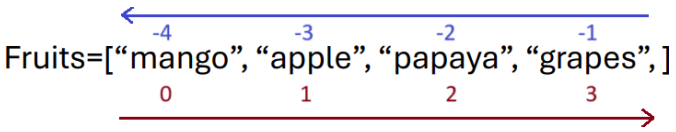
Break statement

```
for a in range(10):
    If a==6:
        break
    print(a)
```

Continue statement

```
for w in range(10,0,-1):
    if w==8:
        continue
    print(w)
```

Lists



Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

Get a list as input from user

```
# For list of integers
list1 = []
list1 = [int(item) for item in input("Enter the list items : ").split()]
or
inlist1=input("Enter the list items: ")
list1 = [int(item for item inlist1.split())]

# For list of strings/chars
list2 = []
list2 = [item for item in input("Enter the list items : ").split()]
print(list2)
```

Tuples newtuple=(11,22,33,44,55)

methods	example	description

a.index(tuple)	>>> a=(1,2,3,4,5)
	>>> a.index(5)
	4
		Returns the index of the first matched item.

a.count(tuple)	>>>a=(1,2,3,4,5) >>> a.count(3) 1	Returns the count of the given element.
len(tuple)	>>> len(a) 5	return the length of the tuple
min(tuple)	>>> min(a) 1	return the minimum element in a tuple
max(tuple)	>>> max(a) 5	return the maximum element in a tuple
del(tuple)	>>> del(a)	Delete the entire tuple.

Sets newset = {1, 2, 3, 4}

add()	insert an element into the set
remove()	remove an element
len()	Get the number of elements in a set
clear()	Remove all elements from a set.
pop()	Remove and return a random element from the set.
copy()	Return a copy of the set.

Dictionaries Marks = {"Ann": 45, "Bob": 60, "Sam": 83}

Method	Description
clear()	Removes all the elements from the dictionary
fromkeys()	Returns a dictionary with the specified keys and values
get()	Returns the value of the specified key
items()	Returns a list containing the tuple for each key-value pair
keys()	Returns a list containing the dictionary's keys
pop()	Removes the element with the specified key
popitem()	Removes the last inserted key-value pair
update()	Updates the dictionary with the specified key-value pairs
values()	Returns a list of all the values in the dictionary

Strings

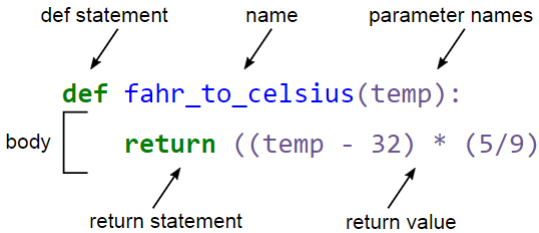
Method	Example	Output
The upper() method returns the string in upper case:	a = "Hello, World!" print(a.upper())	HELLO, WORLD!
The lower() method returns the string in lower case:	a = "Hello, World!" print(a.lower())	hello, world!
The strip() method removes any whitespace from the beginning or the end:	a = " Hello, World! " print(a.strip()) # returns "Hello, World!"	Hello, World!
The replace() method replaces a string with another string:	a = "Hello, World!" print(a.replace("H", "J"))	Jello, World!
The split() method splits the string into substrings if it finds instances of the separator:	a = "Hello, World!" print(a.split(",")) # returns ['Hello', ' World!']	['Hello', ' World!']

Slicing –

>>> my_list[1:3] – extracts a sub list from index 1 up to 3 but not 3.
>>> my_list[1:] - extracts a sub list starting from index 1 to the end.
>>> my_list[:3] – from the beginning up to 3 but not 3.
>>> my_list[:] - extracts the entire list

Subset Lists of sub lists
>>> my_list2[1][0] - Accesses the index 0 of the sub list extracted
>>> my_list2[1][:2] - Accesses up to 2 but not 2 of the sub list extracted

Functions



Scope of variables

```
x = 300 #Global

def myfunc():
  x = 200 #Local
  print(x)
```

File Handling

There are four different methods (modes) for opening a file:

"r"	- Read	- Default value. Opens a file for reading, error if the file does not exist
"a"	- Append	- Opens a file for appending, creates the file if it does not exist
"w"	- Write	- Opens a file for writing, creates the file if it does not exist
"x"	- Create	- Creates the specified file, returns an error if the file exists

Basic structure

```
f=open("demo1.txt", "r")
....
....
f.close()
```

File read

f = open("demo1.txt", "r") → opens the file in read mode
print(f.read()) → reads and prints the entire content of the file
print(f.read(5)) → prints the first 5 characters in a file
print(f.readlines()) → reads all lines in a file, to a list
print(f.readline()) → prints the next line(one line per command)

File Write

- "a" - Append - will append to the end of the file.
- "w" - Write - will overwrite any existing content.

Create a New File

✓ "x" - Create - will create a file, returns an error if the file exist
✓ "a" - Append - will create a file if the specified file doesn't exist
✓ "w" - Write - will create a file if the specified file doesn't exist

---

Delete a file

Remove the file "n1.txt":

```
import os
os.remove("n1.txt")
```

Additional notes:

Comments - Comments starts with a #, and Python will ignore them.

Variables - Variables are containers for storing data values.

Rules for Python variables:

- ✓ A variable name must start with a letter or the underscore character
- ✓ A variable name cannot start with a number
- ✓ A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- ✓ Variable names are case-sensitive (age, Age and AGE are three different variables)
- ✓ Python keywords can't be used

Python with MySQL

Create connection

```
import mysql.connector

# Establishing the connection
mydb = mysql.connector.connect(
  host='localhost',     # Your host, usually localhost
  user='yourusername',  # Your MySQL username
  password='yourpassword', # Your MySQL password
  database='yourdatabase'  # Name of the database to connect to
)

# Creating a cursor object using the cursor() method
mycursor = mydb.cursor()

# Closing the connection
mycursor.close()
mydb.close()
```

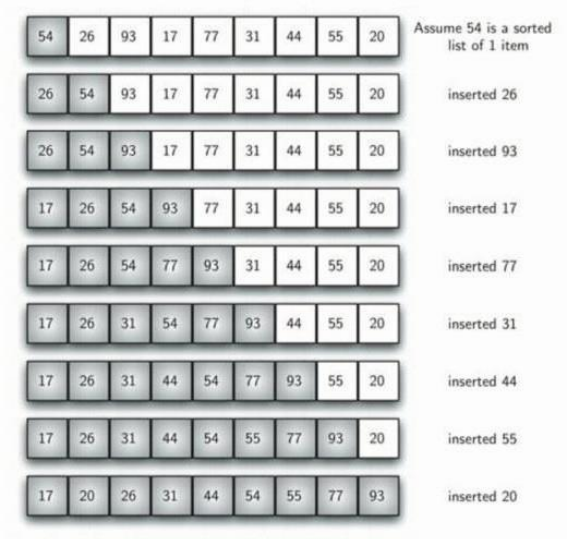
Create a database named 'school'

```
mycursor = mydb.cursor()

mycursor.execute("CREATE DATABASE
school ")
```

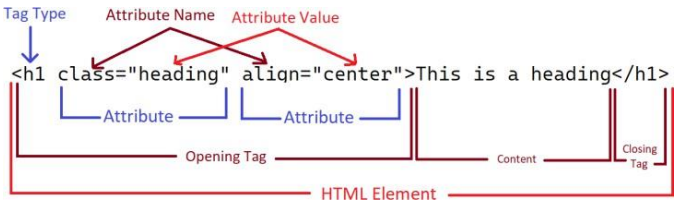
Creating a table

```
mycursor.execute(
  "CREATE TABLE student ( \
  regNo INT PRIMARY KEY, \
  name VARCHAR(100), \
  address VARCHAR(255), \
  contactNo VARCHAR(15) )"
)
```

Bubble sort method (for Sorting list in ascending order)



Unit-10(Web Development)



Basic Tags

<html> </html>
Creates an HTML document

<head> </head>
Sets off the title C other info that isn't displayed

<body> </body>
Sets off the visible portion of the document

<title> </title>
Puts name of the document in the title bar; when bookmarking pages, this is what is bookmarked

Body attributes

<body bgcolor=?>
Sets background color, using name or hex value

Text Tags

<pre> </pre>
Creates preformatted text

<h1> </h1> --> <h6> </h6>
Creates headlines -- H1=largest, H6=smallest

<cite> </cite>
Creates a citation, usually processed in italics

<address> </address>
Creates address section, usually processed in italics

Creates subscript text

Creates subscript text

Links

clickable text
Creates a hyperlink to a URL

<a name="NAME">
Creates a target location within a document

clickable text
Creates a link to that target location

Formatting

Creates a new paragraph

Inserts a line break (carriage return)

<q></q>
Adds short quotations

<blockquote> </blockquote>
Puts content in a quote - indents text from both sides

Creates bold text

Creates italicized text

<u></u>
Creates underline text

<strike><strike>
Creates strikethrough text

Emphasizes a word

Emphasizes a word(Bold)

<small></small>
Much lighter text

<mark></mark>
Highlights text

Sets size of font - 1 to 7

Sets font color

Defines the font used

Aligns the position of a content

Used to format block content with CSS

Used to format inline content with CSS

Lists

<ul type= “?”>
Creates an unordered list

<ol type= “?” start=”?”>
Creates an ordered list

Expresses each list item

<dl> </dl>
Creates a definition list

<dt>
Precedes each definition term

<dd>
Precedes each definition data

Graphical elements

<hr size=? width=?>>
Inserts a horizontal rule. Sets size (height)and width of horizontal rule

<img src="URL" alt= “?”>
Adds image; it is a separate file located at the URL

<audio controls>
<source src= “?” type= “?”>
</audio> - Adds an audio file to the HTML

<video controls>
<source src= “?” type= “?”>
</video> - Adds a video file to the HTML

<marquee></marquee>
Creates a moving text/ element

Tables

<table> </table>
Creates a table

<tr> </tr>
Sets off each row in a table

<td> </td>
Sets off each cell in a row

<th> </th>
Sets off the table header

<caption></caption>
Adds a topic to the table

<td colspan= “?”>
Sets number of columns a cell should span

<td rowspan=?>
Sets number of rows a cell should span

Forms

<form method= “?” action= “?”> </form>
Defines a form,method is GET or POST and action is the resulting php file.

<input type="text" name=? size=? required>
Creates a one-line text area. Size sets length, in characters.

<input type = “text” placeholder=”?” maxlength= “?” >
To add a hint and give a maximum length to the input

<input type="radio" name=? value=?>
Creates a radio button.

<input type="radio" name=? value=? checked>
Creates a radio button which is pre-checked.

<input type="checkbox" name=? value=?>
Creates a checkbox.

<input type="checkbox" name=? value=? checked>
Creates a checkbox which is pre-checked.

<select name=?> </select>
Creates a dropdown menu

<option>
Sets off each menu item

<textarea name=? cols="x" rows="y"></textarea>
Creates a text box area. Columns set the width; rows set the height.

<input type="submit" value=?>
Creates a submit button. Value sets the text in the submit button.

<input type="reset">
Creates a reset button

HTML comments

<!--Comments goes here-->

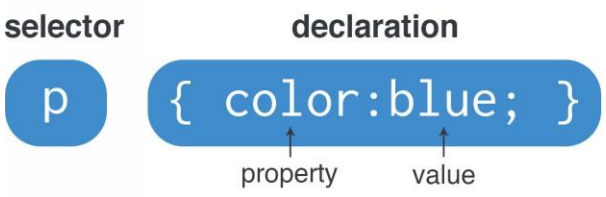
Frames

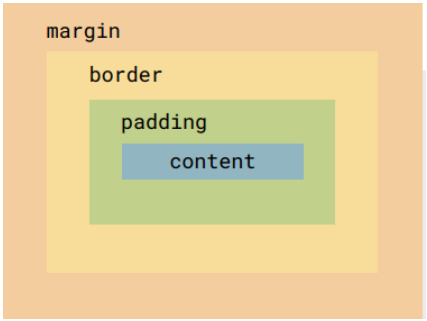
<frameset rows="40%, 60%, >
<frame src="?">
<frame src="?" >
</frameset> - Creates 2 frames in rows

<frameset cols="*, *">
<frame src="? " >
<frame src="? " >
</frameset> - Creates 2 frames in columns

<iframe src="?" title="?"> </iframe>
Adds another webpage as an inline element/frame.

CSS





3 ways to write CSS

(A) Inline CSS

```
<element style="property: value;">
```

(B) Internal CSS

```
<head>
    <style>
        selectors { property: value; }
    </style>
</head>
```

(C) External CSS

```
<head>
    <link rel="stylesheet" href="style.css"
      type= “text/css”>
</head>
```

Types of selectors

elementname – element selector
.classname – class selector
#idname – id selecotor
* - universal selector

Text Properties

p {**word-spacing**: 0.4em;} , #fg {**word-spacing**: -0.2em;}
Defines an additional amount of space between words.

---

p{**letter-spacing**: 0.1em;} h1{**letter-spacing**: -0.1em;}
Defines an additional space between characters.

---

a **{ text-decoration**: none; }
Allows text to be decorated through fiveproperties: underline, overline, line-through, blink, none.

---

h1{**text-transform**:uppercase;} ,p**{text-transform**: none;}
Allows for capitalize, uppercase, lowercase), none.

---

h1 { **text-align:** center; }, p { **text-align**: justify; }
Used to justify text left, center, right, and justify.

P { **text-indent**: 5em; }
Used to specify the amount of indentation prior to the first line of text.

Font Properties

p { **font-family**: "New Century Schoolbook", Times, serif; }
Changes the font of certain words, sentences, paragraphs, etc.

---

h1 { **font-style:** oblique; }, p { **font-style**: normal; }
Changes text: normal, oblique, and italics.

---

h1 { **font-weight**: 800; } or p { **font-weight**: normal; }
Used to specify the weight of the font.

---

h1 { **font-size**: large; } or p { **font-size**: 12pt; }
LI { **font-size**: 90%; } , STRONG { **font-size**: larger; }
Used to modify the size of the displayed font.

---

span { **font-variant**: small-caps; }
Used to display font in normal | small-caps | initial | inherit.

---

p { **line-height**: 200%; }
Used to control the spacing between baselines of text.

---

p **{ font**: italic bold 12pt/14pt Times, serif; }
Used to combine all properties of fonts.(shorthand property)

Color and Background Properties

h1 { **color**: blue; } or H2 { **color**: #000080; }
Changes the color of text.

---

body { **background-color**: white; }
h1 { **background-color**: #000080; }
Sets the background color of an element.

---

body { **background-image**: url(/images/foo.gif); }
p { **background-image**: url(http://www.htmlez.com/bg.png);}
Sets the background image of an element.

List Properties

li.square { **list-style-type**: square; }
li.plain { **list-style-type**: none; }
Specifies the type of list-item marker(<ul>).

---

ol { **list-style-type**: upper-alpha; } (A B C D E..)
ol { **list-style-type**: decimal; } (1 2 3 4 5…)
Specifies the type of list-item marker(<ol>).

---

ul.check { **list-style-image**: url(/checkmark.gif); }
ul li.x { **list-style-image**: url(x.png); }
Specifies the image that will be used as
list-item marker

---

ul { **list-style-position**: inside;}
Determines where the marker is placed in regard to the list item. If the value inside is used, the lines will wrap under the marker instead of being indented. outside is default.

Padding Properties

p { **padding-top**: 20%; }
Describes the amount of space between the top border and the content of the selector.

---

p { **padding-right**: 20 px; }
Describes the amount of space between the right border and the content of the selector.

---

p { **padding-bottom**: 5 em; }
Describes the amount of space between the bottom border and the content of the selector.

---

p { **padding-left**: 15 pt; }
Describes the amount of space between the left border and the content of the selector.

#getinfo { **padding**: 2em 4em 5em 4em; }
Shorthand for the padding-top, padding-right, padding-bottom, and padding-left properties.

Table Properties

table, th, td { **border**: 1px solid black; }
border (Shorthand property)

---

table, th, td {**border style**: dotted}
changes the style of the table borders to dotted, dashed, solid, double or hidden

---

table {**border-collapse**: collapse; }
collapse to a single border

Margin Properties

body { **margin-top**: 5pt; }, DT { **margin-bottom**: 3em; }
Set top, bottom margins of an element by specifying a length

---

p.narrow { **margin-right**: 50%; }, td { **margin-left**: 50%; }
Set right,left margins of an element by specifying a percentage.

---

body { **margin**: 5em; }
Sets the margins of an element for all directions

Border Properties

p **{ border-width**: 10px; }
Used to set the width of an element's border (either all borders, or specifying a border.

---

p { **border-color**: #00000; }
Used to set the color of an element's border.

---

p { **border-style**: dotted; }
Sets style of a border - none, dotted, dashed, solid, double.

p { **width**: 15px; }, H1 { **width**: 35%; }, .foo { **width**: auto; }
Each block-level or replaced element can be given a width, specified as a length, a percentage, or as auto.

---

p { **height**: 15px; }, H1 { **height**: 35%; }, .foo { **height**: auto; }
Each block-level or replaced element can be given a height, specified as a length or as auto.

PHP

Basic Syntax

File starts with - **<?php**
Single line comment - **// a comment**
Multi-line comment - **/ comment /**
End of each instruction - **;**
File ends with - **?>**
Outputs data - **echo**

Data types

$boolean1 = true; **// Boolean**
$int = 12; **// integer**
$float = 3.1415926; **// float**
$fruit = 'apple'; **//string**
$arr = array("hello", "world", "!"); **// array**

Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
**	Exponentiation

Logical Operators

And CC	And
Or ||	Or
xor	Exclusive or
!	Not

For loop –

```
# => 12345
for ($i = 1; $i <= 5; $i++) {
        echo $i;
}
```

Comparison Operators

==	Equal
===	Identical
!=	Not equal
<>	Not equal
!==	Not identical
<	Less than
>	Greater than
<=	Less than or equal
>=	Greater than or equal
<=>	Less than/equal/greater than

Control Structure: IF

```
// if something is true do something else
if( $something == true ) {
        //do something;
} elseif( $something == false ) {
        //do something;
} else {
        //do nothing ;
}
```

Control Structure: Loops

Do –

```
$i = 1;
# => 12345
while ($i <= 5) {
        echo $i++;
}
```

Do while -

```
$i = 1;
# => 12345
do {
        echo $i++;
} while ($i <= 5);
```

Foreach Loop

```
$a = ['foo' => 1, 'bar' => 2];
# => 12
foreach ($a as $k) {
        echo $k;
}
```

Functions

```
<?php
function writeMsg() {
        echo "Hello world!";
}
writeMsg(); // call the function
?>
```

Connecting MySQL to PHP

→ mysql_connect()
- opens a non-persistent MySQL connection

→ mysql_close()
- closes a non-persistent MySQL connection

→ mysql_error()
- returns the error description of the MySQL operation

→ mysql_fetch_array()
- returns a row from a recordset as an associative array

→ mysql_query()
- executes a query on a MySQL database

→ mysql_select_db()
- sets the active MySQL database

→ mysql_ASSOC
- columns are returned into the array with fieldname as the array index

→ die()
- display a message
- used to print string messages

Switch statement

```
switch (n) {
        case label1:
                code to be executed if n=label1;
                break;
        case label2:
                code to be executed if n=label2;
                break;
        case label3:
                code to be executed if n=label3;
                break;
                ...
        default:
        code to be executed if n is different from all
        labels;
}
```

Create connection

```
$conn = new mysqli($servername, $username,
$password, $dbname);
```

Check connection

```
if ($conn->connect_error) {
        die("Connection failed: " . $conn-
        >connect_error);
};
```

Close connection

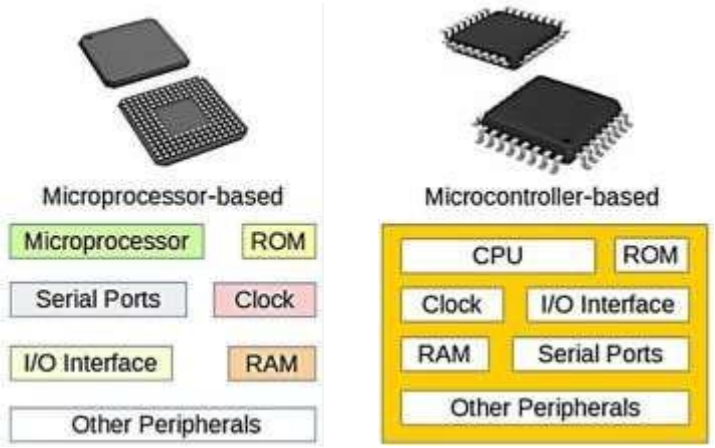
```
$conn->close();
```

Unit-11(IOT)

Embedded System

Embedded system is a computer system embedded into some other system such as a refrigerator, washing machine, car, etc. It also follows Input, Process and Output (IPO) model.
Input → Sensors capture the state
Process → Processor processes
Output → Performs through actuators

The **microcontroller** is a single chip containing a CPU, memory, I/O ports, and other peripherals.
Most embedded systems are microcontroller based because they **do not require expensive, powerful microprocessors** to implement their basic functionalities.

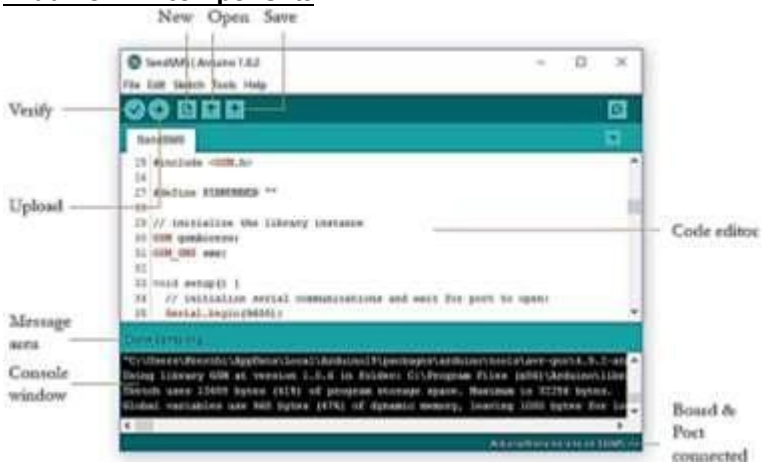


Examples for Microcontrollers
- micro:bit
- EasyPIC
- Arduino board
- Raspberry Pi board

Arduino - open-source, low cost, easy-to-use hardware, and software platform with Cross platform support. Offers extensive official and community support and Extensive availability of software libraries.

Arduino Uno Board

Arduino IDE components



Microcontroller based Development Systems vs. microprocessor-based systems

Microprocessor	Microcontroller
Used in general-purpose systems like PCs	Used in embedded systems like appliances
Only CPU; external memory C I/O required	CPU, memory, and I/O are built-in
Larger circuit, not suitable for compact systems	Compact and ideal for small devices
Higher system cost and power consumption	Lower cost and low power usage
Lacks power-saving features	Has power-saving modes
Based on Von Neumann architecture	Based on Harvard architecture
Requires external bus for peripherals	Uses internal bus
High-speed operation	Operates at lower speeds (up to 200MHz)

Key Differences between Microprocessor and Microcontroller:

- A **microprocessor** contains only the CPU, while a **microcontroller** has a CPU, memory, and I/O ports all on a single chip.
- **Microprocessors** are mainly used in personal computers, whereas **microcontrollers** are designed for embedded systems.
- **Microprocessors** connect memory and peripherals via an external bus; **microcontrollers** use an internal bus for control.
- **Microprocessors** typically follow the **Von Neumann** architecture, while **microcontrollers** use the **Harvard** architecture.
- Microprocessors are more complex, costly, and handle many instructions; microcontrollers are simpler, cheaper, and handle fewer instructions.

Basic Arduino Programming Guidelines

1. Every instruction must end with a **semicolon (;)**.
2. Conditional (if) and loop structures (for, while) as well as functions must be enclosed in **curly braces {}**.
3. **Single-line**: // comment here
   Multi-line: /* comment block */

4. void setup() runs once when the program starts.
   void loop() runs repeatedly.
5. Use pinMode(pin, mode) in setup() to set a pin as INPUT, OUTPUT
6. Use digitalRead(pin) for digital input.
   Use analogRead(pin) for analog input.
7. digitalWrite() is used to send output signals to a digital pin.

- Arduino code is **case-sensitive**

Pinmode: This identifies the pin on which the inputs and output must be given.

```
void setup ()
{
      pinmode(2, OUTPUT);
}
```

```
void setup ()
{
      pinmode(3, INPUT);
}
```

DigitalWrite: This command is used to change the voltage of I/O pins of Arduino board.

```
Ex: digitalWrite(2,HIGH)
    digitalWrite(2,LOW)
```

Begin: define the number of bits should be transmitted when communicating with a communication device

Ex : Serial.begin (9600)

Delay: to change the me frame on which a task will be performed.

```
Ex: digitalWrite(2,HIGH);
    delay(1000);
    digitalwrite(2,LOW)
```

If:

```
if (condition)
{
        statement(s)
}
```

For

```
for (initialization;condition;increment)
{
        statement(s);
}
```

While

```
while(condition)
{
        statement(s)
}
```

IoT (Internet of Things)

A network of interconnected smart systems where everyday objects communicate and act autonomously to improve convenience and comfort in life.

Smart World

A vision where **autonomous, interconnected smart systems** work together seamlessly—for example, a smart alarm clock communicating with a smart kettle or refrigerator.

Things

Everyday physical objects, ranging from **wristwatches to cars and buildings**.

When embedded systems are connected to the **Internet**, they can:

- **Interact** with each other, **Communicate** with users
  Form a large network called the **Internet of Things**

IoT is enabled by technologies like IPv6 for large address space, cheaper and faster networking, compact and affordable sensors, and efficient, low-power processors and storage. These advancements make connecting and controlling smart devices easier and more practical.

Major components of IOT

1) Smart devices C sensors
2) IOT gateway
3) Cloud
4) Analytics
5) User Interface

Examples for IOT

Smart Watch
Google Home devices
Smart door locks
Smart Gardening
Video doorbells
Personal Assistant

Future Systems Using IoT

1. **Smart transportation** – Automated control based on vehicle size.
2. **Environmental management** – Automated control and improvement.
3. **Healthcare** – Monitoring patients, automatic medicine dispensing.
4. **Construction** – Observing and managing activities remotely.
5. **Machine communication** – Transmitting data between machines.
6. **Vehicle tracking** – Real-time location tracking.
7. **Agriculture** – Automatic water supply based on need.
8. **City monitoring** – Smart city surveillance and management.
9. **Remote home control** – Operate lights, motors from a distance.
10. **Smart infrastructure** – Homes, outlets, schools, cities.

Challenges of IoT

- Data confidentiality and copyright concerns.
- Insufficient research and updates.
- Complex hardware requirements.
- Dependence on stable power supply.

Disadvantages of IoT

- **Privacy issues** – User data may be exposed.
- **Security issues** – Risk of unauthorized access.

To construct an embedded system, it is essential to follow some steps as given below:

- Construct the schematic diagram and assemble hardware
- Design firmware
- Develop firmware
- Compile firmware and Uploade machine code

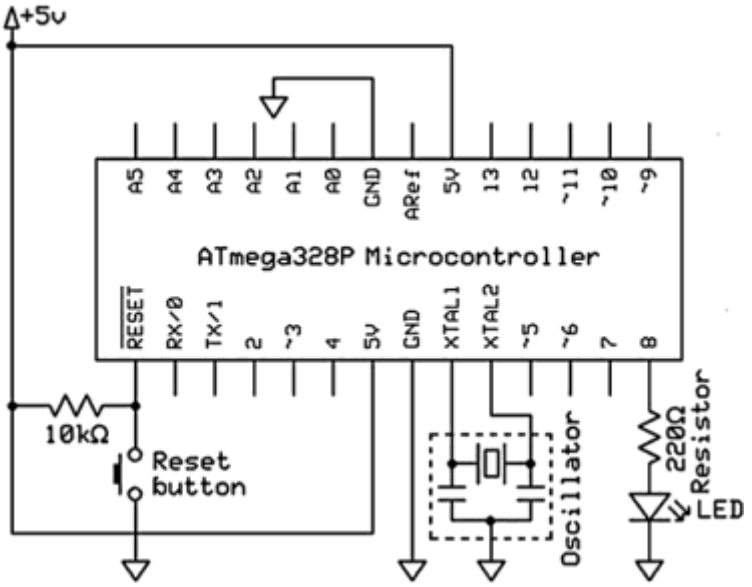
Examples for Embedded systems using Arduino

SYSTEM 1: Blinker

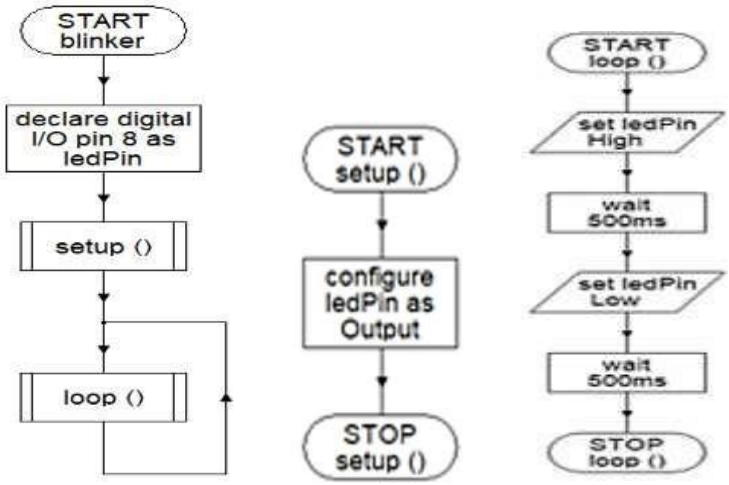
Required components

1 × Arduino Uno microcontroller-based development board
1 × LED
1 × 220Ω Resistor

Schematic Diagram



Flowchart



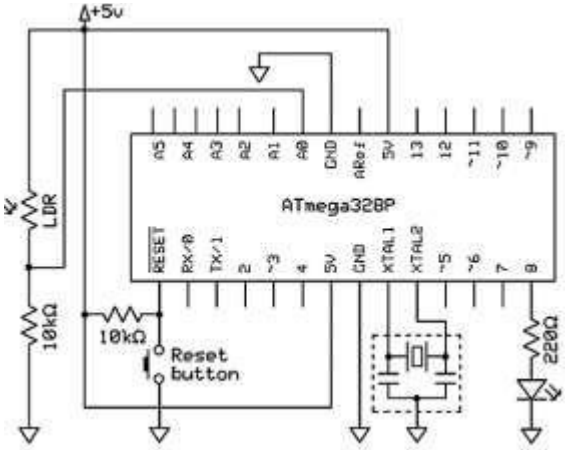
```
// blinks an LED every ½ a second
const int ledPin = 8;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(500);
  digitalWrite(ledPin, LOW);
  delay(500);
}
```

SYSTEM2: AutoLight

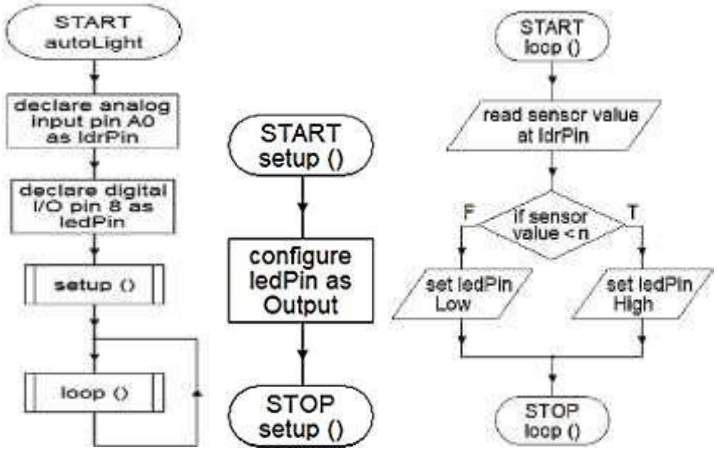
Required components

1 × Arduino Uno microcontroller-based development board
1 × LED
1 × 220Ω Resistor
1 × Light Dependent Resistor (LDR)
1 × 10Ω Resistor

Schematic Diagram



Flowchart



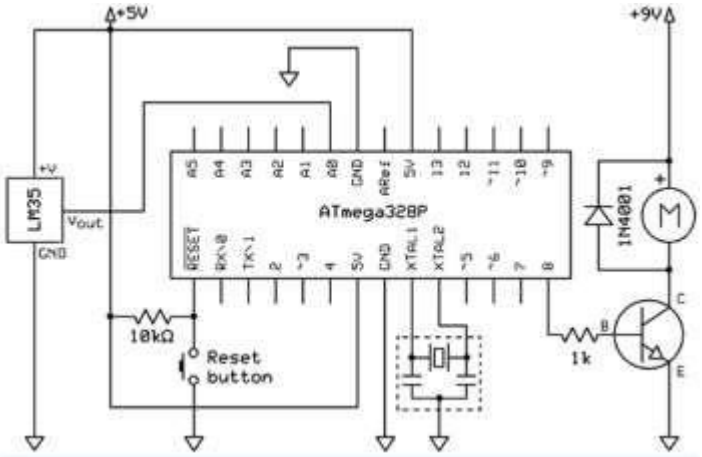

```
// switches an LED on and off depending on light intensity
const int ldrPin = A0;
const int ledPin = 8;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  int sensorValue = analogRead(ldrPin);
  if (sensorValue < 150)
    digitalWrite(ledPin, HIGH);
  else
    digitalWrite(ledPin, LOW);
}
```

SYSTEM3: AutoFan

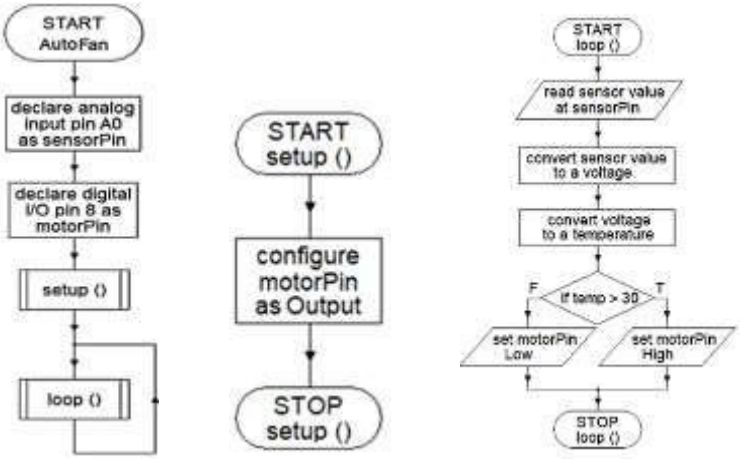
Required components

1 × Arduino Uno microcontroller-based development
1 × 9 Volts DC Motor C 1 × LM35 Temperature Sensor
1 × BC547 Transistor C 1 × 1kΩ Resistor
1 × 1N4001Rectifier Diode

Schematic Diagram



Flowchart



```
// switches a motor of a fan on and off depending on room temperature
const int sensorPin = A0;
const int motorPin = 8;

void setup()
{
  pinMode(motorPin, OUTPUT);
}

void loop()
{
  int sensorValue = analogRead(sensorPin);
  float voltage = value * 5.0 / 1024;
  float temp = voltage * 100;
  if (temp > 30)
    digitalWrite(motorPin, HIGH);
  else
    digitalWrite(motorPin, LOW);
}
```

```
// triggers an alarm when a door is opened
const int switchPin = 9;
const int buzzerPin = 8;

void setup()
{
  pinMode(switchPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop()
{
  int switchState = digitalRead(switchPin);
  if (switchState == LOW)
    tone(buzzerPin, 262);
  else
    noTone(buzzerPin);
}
```

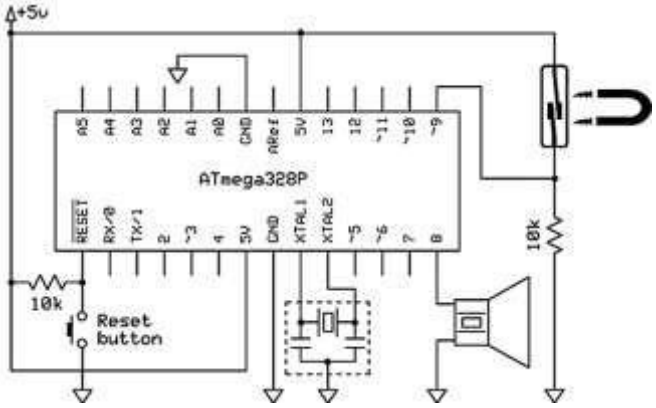
SYSTEM4: Door-Alarm

Required components

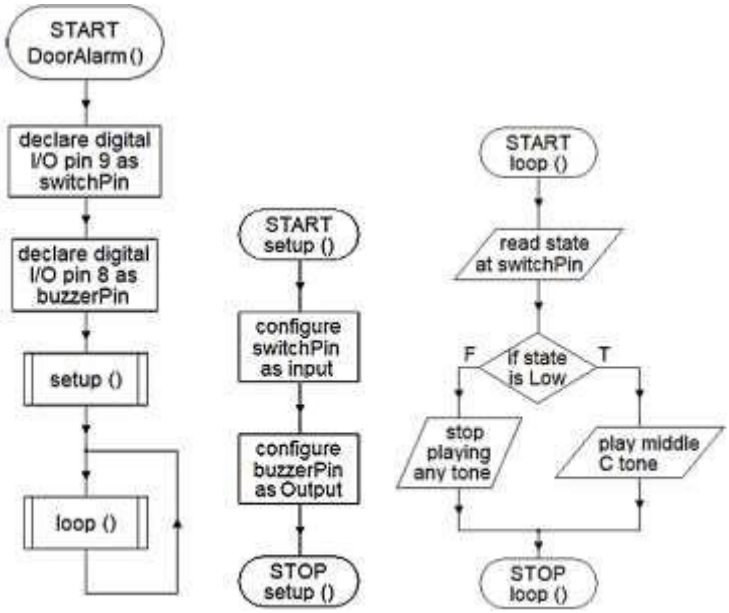
1 × Arduino Uno microcontroller-based development Board , 1 × Piezo Buzzer , 1 × Reed Switch

1 × 10kΩ Resistor

Schematic Diagram



Flowchart



Unit-12(E-Commerce)

Role of ICT in Modern Business

- ICT is a key enabler for business success.
- Helps solve problems like inefficiency, low productivity, strong competition, and limited market reach.
- Started with computerizing manual processes, now essential for all businesses.
- Improves productivity, decision-making, and customer service.
- Enables innovative business models and expansion into global markets.
- No modern company can succeed without ICT.

Digital Economy

- A digital economy is where economic activities happen through digital technologies.
- ICT advancements, especially the Internet (Web 2.0), drive this transformation.
- Examples:
    - Reading newspapers online, sharing, and discussing content.
    - Online shopping with digital payments/cryptocurrencies.
    - Freelancing opportunities like translation jobs.
- Digital economy involves production, distribution, and consumption of goods/services in digital form.
- Internet technology enables digital products (e-books, music, video, software, etc.) to be created and shared globally.
- Also enables IoT (Internet of Things) – connecting people with devices (e.g., smart appliances).
- Reduces barriers of geography, culture, and society → leading to globalization and a "global village."

New Business Methods in the Digital Economy

The rise of the digital economy has introduced innovative business methods that reshape how buyers and sellers interact. These methods make use of internet technology to improve efficiency, reduce costs, and create win–win opportunities for both sides.

New Business Methods in the Digital Economy:

- Group Purchasing
- Reverse Auction
- E- Market Place

1. Reverse Auctions

A reverse auction is an online bidding process where buyers announce their needs (goods/services) and sellers compete by offering the lowest possible price. Unlike traditional auctions where buyers raise the price, in reverse auctions the price decreases as sellers underbid each other.

Why followed?

- To ensure cost efficiency in large purchases.
- To create high competition among suppliers, leading to better deals.
- To increase transparency and fairness in procurement.

Examples:

- Government e-procurement systems (e.g., for medical supplies, office equipment).
- Large corporations sourcing raw materials or IT services through online bidding platforms.

Benefits:

- Buyers → Lowest possible cost.
- Sellers → Equal chance to compete for large contracts.

2. Group Purchasing (Collective Buying)

Group purchasing is a method where many customers combine their demand to buy a product/service together. By purchasing in bulk, they receive discounted prices from sellers.

Why followed?

- To reduce individual costs by taking advantage of bulk discounts.
- To give small buyers purchasing power equal to big buyers.
- To help businesses quickly increase sales volumes.

Examples:

- Groupon or MyDeal.lk: users get discounts when a minimum number of people purchase the same deal.
- Office supply companies giving discounts to schools or universities that order in bulk.

Benefits:

- Customers → Lower prices, better value.
- Businesses → Quick customer acquisition and higher sales volume.

3. e-Marketplace (Electronic Marketplace)

An e-Marketplace is an online platform that connects multiple buyers and sellers to trade goods and services. It serves as a virtual marketplace where transactions happen digitally.

Why followed?

- To reach wider markets without physical limitations.
- To provide a transparent platform where customers can compare prices, quality, and reviews.
- To reduce marketing and distribution costs for businesses.

Examples:

- Amazon, eBay, Alibaba (global scale).
- Daraz.lk (Sri Lanka's leading e-commerce platform).

Benefits:

- Customers → Convenience, variety, and easy price comparison.
- Businesses → Expanded market reach, more visibility, and new revenue channels.

Business Models in the Digital Economy

Organizations can be categorized based on how they use ICT and online platforms for their operations. Three major types are: **Pure Brick, Brick-and-Click, and Pure Click**.

1. Pure Brick Organizations

- These are traditional businesses that operate only through physical stores/offices.
- They do not use online platforms for selling or interacting with customers.
- Customers must visit the physical location to purchase products or services.

Why followed?

- Suitable for businesses where physical interaction is necessary (e.g., medical care, salons).
- Some customers prefer the trust, touch-and-feel experience, and personal service of physical shops.

Examples:

- Kapruka Bakeries (local outlets only, without online system).
- Sathosa (state-owned supermarket chain – traditional outlets).
- Small-town bookshops, salons, or clothing stores that rely purely on physical presence.

Advantages:

- Personal interaction with customers.
- Strong local presence and customer loyalty.

Limitations:

- Limited market reach (only local).
- Higher operating costs (rent, staff, utilities).

2. Brick-and-Click Organizations

- Businesses that operate in both physical outlets and online platforms.
- Customers can either visit the store or order products/services online.
- This model combines the trust of physical stores with the convenience of online access.

Why followed?

- To reach a wider customer base (local + online).
- To provide flexibility: customers can choose how they want to shop.
- To stay competitive in the digital economy.

Examples:

- Keells Super → Physical supermarkets + Keells online shopping app.
- ODEL → Fashion retailer with physical stores + online store.
- Wijeya Newspapers → Printed newspapers + online news portals (e.g., Daily Mirror Online).

Advantages:

- Wider reach (local customers + global audience).
- Better customer convenience.
- Improved brand trust (physical store presence).

Limitations:

- High cost of maintaining both systems (physical and digital).
- Requires good logistics and delivery management.

3. Pure Click Organizations

- Businesses that operate only through online platforms.
- No physical outlets—everything from browsing to payment and delivery happens digitally.

Why followed?

- Lower operational costs (no need for rent/staff for physical stores).
- Ability to reach national or even global markets easily.
- Increasing demand for online convenience and cashless payments.

Examples:

- PickMe (ride-hailing C food delivery app).
- Kapruka.com (Sri Lanka's first e-commerce platform).
- Ikman.lk (classified ads website).
- Daraz.lk (online marketplace).

Advantages:

- Cost-effective operations.
- Unlimited market reach (beyond physical boundaries).
- 24/7 availability.

Limitations:

- No physical interaction → trust issues for some customers.
- Dependent on internet connectivity and digital literacy.

Business Functions and the Role of ICT

ICT plays a vital role across all business functions, enabling efficiency, accuracy, and global connectivity. Below are the key areas:

1. Accounting and ICT

- ICT automates financial record keeping and reporting.
- Software like QuickBooks, Tally, Sage helps in preparing balance sheets, income statements, and tracking expenses.
- Reduces human error and improves accuracy in financial decisions.

Examples: Automated payroll systems, online tax submissions.

2. Human Resource (HR) and ICT

- ICT supports recruitment, training, and employee management.
- HR software helps in storing employee records, managing leave, payroll, and performance appraisals.
- Online platforms enable e-recruitment and virtual training programs.

Examples: LinkedIn for recruitment, online HR portals.

3. Production and ICT

- ICT supports automation, quality control, and scheduling in production.
- Systems like Computer-Aided Manufacturing (CAM) and Computer-Aided Design (CAD) help improve efficiency.
- Sensors and IoT devices ensure real-time monitoring of production lines.

Examples: Apparel factories using automated sewing machines with ICT systems.

4. Marketing s Sales and ICT

- ICT enables digital marketing, e-commerce, and customer relationship management (CRM).
- Companies use websites, social media, and email to reach global customers.
- CRM software tracks customer behavior and personalizes marketing campaigns.

Examples: Daraz.lk for online sales, Facebook/Instagram ads.

5. Supply Chain Management and ICT

- ICT ensures smooth flow of materials from suppliers to customers.
- Tools like ERP systems track inventory, manage logistics, and optimize warehouse operations.
- Reduces delays and improves cost-efficiency.

Examples: Keells supermarkets tracking stock and deliveries via ICT systems.

6. Business Communication and ICT

- ICT enables instant and effective communication within and outside organizations.
- Tools: Emails, video conferencing, instant messaging, and collaborative platforms.
- Enhances global teamwork and decision-making.

Examples: Zoom/Teams for meetings, Slack for internal communication.

Secure Payment Mechanisms

Secure payment mechanisms are systems and technologies that ensure safe, reliable, and fraud-free financial transactions in the digital economy. They protect both buyers and sellers by using authentication, encryption, and trusted third parties.

1. Payment Gateways

- A payment gateway is a digital service that acts as a bridge between customers, merchants, and banks.
- It encrypts sensitive data (like card details) and securely transmits it to banks for authorization.
- Ensures transactions happen in real-time with minimal risk of fraud.

Examples (Sri Lanka s Global):

- PayHere, WebXPay (Sri Lanka)
- Stripe, Razorpay, 2Checkout (Global)

Why Important?

- Protects card details from being exposed.
- Builds customer trust in e-commerce.

2. Secure Credit Card Payments

- Credit card payments use secure protocols to prevent misuse.
- Mechanisms used:
    - SSL (Secure Socket Layer): Encrypts information sent over the internet.
    - CVV numbers: Extra security beyond the card number.
    - 3D Secure Authentication (e.g., Verified by Visa, MasterCard SecureCode): Requires OTP (One-Time Password) or PIN for confirmation.

Example:

Paying with Visa/MasterCard on Daraz.lk using OTP verification.

Why Important?

- Prevents identity theft and unauthorized transactions.
- Enhances customer confidence in online shopping.

3. Third Party Systems (e-Wallets s Payment Processors)

- These are trusted intermediaries that handle transactions on behalf of users.
- Users store money in a digital wallet or link their bank accounts/cards, and the system processes payments securely.

Examples:

- Global: PayPal, Apple Pay, Google Pay.
- Sri Lanka: eZ Cash, Genie, mCash.

Why Important?

- Reduces the need to share card/bank details directly with merchants.
- Convenient, fast, and globally accepted.

4. Mechanisms Ensuring Secure Transactions

(a) Data Encryption

- Converts sensitive data (like card numbers, passwords) into coded form.
- Even if hackers access the data, they cannot read it without a decryption key.
- Example: SSL/TLS encryption used in all major banking and e-commerce websites.

(b) Virtual Currencies

- Digital form of money issued by private entities.
- Can be used for online transactions, often limited to specific platforms.
- Example: Virtual credits in online games, reward points, gift cards.

(c) Cryptocurrencies

- A type of virtual currency that uses blockchain technology to ensure secure and decentralized transactions.
- Not controlled by banks or governments.
- Example: Bitcoin, Ethereum, Litecoin.
- In Sri Lanka, cryptocurrency trading is not officially legalized, but awareness is growing.

Analyses the relationship between ICT and business operations

E-Commerce

E-commerce refers to buying and selling of goods and services electronically through the internet.

E-Commerce Includes:

- Online shopping platforms (e.g., Daraz, Amazon).
- Online banking and digital payments.
- Online ticket booking (flights, buses, cinema).
- Online education and e-learning platforms.
- Online delivery services (PickMe, Uber Eats).

Focus: Primarily deals with transactions of goods and services online.

E-Business

E-business is a broader concept that includes not only buying/selling but also all business processes carried out using ICT.

E-Business Includes:

- E-commerce activities (online buying/selling).
- Supply chain management (inventory C logistics).
- Customer relationship management (CRM systems).
- Online recruitment and HR management.
- Business communication (email, video conferencing).

Focus: Covers all aspects of running a business digitally, not just sales.

Types of E-Commerce Transactions

B2B – Business to Business

- Transactions between two businesses.
- Example: A Sri Lankan tea exporter selling bulk tea to an international distributor.
- **Benefit:** Builds strong supply chain relationships.

B2C – Business to Consumer

- Businesses sell directly to customers online.
- Example: Daraz.lk selling electronics, clothes, and groceries to individuals.
- **Benefit:** Convenience for customers, wider market for businesses.

C2C – Consumer to Consumer

- Consumers sell products/services to other consumers via online platforms.
- Example: ikman.lk where individuals sell used vehicles, phones, or property.
- **Benefit:** Easy marketplace for second-hand goods.

C2B – Consumer to Business

- Consumers offer products or services to businesses.
- Example: A freelance graphic designer in Sri Lanka offering services to companies via Fiverr/Upwork.
- **Benefit:** Businesses get flexible services; individuals earn income.

B2E – Business to Employee

- Businesses provide services/products directly to employees through digital platforms.
- Example: Company portals offering discounted health insurance or e-learning facilities for employees.
- **Benefit:** Improves employee satisfaction and loyalty.

G2C – Government to Citizen

- Government provides digital services to the public.
- Example: Sri Lanka's **LankaGate** and **e-Services Portal** for paying utility bills, taxes, or exam applications online.
- **Benefit:** Saves time, improves efficiency, reduces paperwork.

Internet and Business

Online marketplace

An online marketplace is an e-commerce platform that acts as a meeting point for multiple sellers and a large pool of buyers. Instead of managing their own websites, sellers can open virtual stores inside the marketplace and directly reach global customers. Amazon's marketplace is a classic example where individual merchants can sell products, while Rover.com focuses on pet-related services. In Sri Lanka, Daraz.lk functions as a leading online marketplace. This model benefits sellers by giving them access to a wide customer base and benefits buyers by providing them with variety and convenience in one place.

Virtual Storefront

A **virtual storefront** uses digital technology to display products online or on electronic screens in public places. Tesco is famous for setting up such storefronts at airports and subway stations, where customers could scan QR codes of products using a mobile app, place their order instantly, and have items delivered later. Similarly, any online shop within a marketplace like Daraz or Amazon is also a virtual storefront. This method provides convenience to customers while allowing companies to present their products in innovative ways without maintaining large physical stores.

Information brokers

Information brokers are businesses that help customers filter and compare information across many sources. Instead of manually visiting multiple websites, an information broker automates the process and provides results in an easy-to-understand way. For example, comparison sites can help customers identify which store sells a particular book or gadget at the lowest price. By saving time and effort, information brokers add significant value in a world overflowing with information.

Content Provider

A content provider is a business that supplies digital content such as news, music, videos, photos, or games. Content providers earn money by charging for downloads, offering subscriptions, or running advertisements. Apple's iTunes Store is one of the most famous examples that changed the music industry. Today, streaming platforms such as Netflix and Spotify have taken content provision even further. In Sri Lanka, local examples include online newspapers like Daily Mirror Online and streaming platforms like Hiru TV Live. These services make it possible for consumers to access entertainment and information instantly from anywhere in the world.

Portal

A Portal is an entry point to e-commerce and online services. It aggregates content from multiple providers and allows users to search, browse, and access various categories of goods, services, or information from one location. Portals today also offer personalization by making recommendations based on user profiles and displaying targeted advertisements. Yahoo is a well-known example, while in Sri Lanka, LankaGate serves as a government portal providing citizens access to services such as bill payments, exam applications, and more.

Online service providers

Online service providers are businesses that deliver services through the internet, ranging from communication to file storage, banking, healthcare, and entertainment. Google is an excellent example, offering services like Gmail, Google Drive, and YouTube. Facebook also functions as a service provider by offering social networking to millions of users worldwide. In Sri Lanka, online service providers include Dialog's digital services, as well as online banking platforms like Sampath Vishwa. These services increase convenience, reduce costs, and make global access possible.

Virtual Communities

virtual communities are groups of people who gather and interact online because of shared interests, opinions, or experiences. These communities provide a space for members to share ideas, build relationships, and strengthen loyalty to brands or causes. Examples include Facebook groups, Reddit communities, and online student forums. In Sri Lanka, many Advanced Level student groups function as virtual communities where learners share study materials and support each other. Virtual communities therefore play a major role in shaping consumer opinions and brand engagement in the digital age.

Revenue Models in Digital Business

Subscription as a Revenue Model

In this model, customers pay a recurring fee (monthly, yearly, or weekly) to access a product or service. It ensures continuous income for the business.

Examples include Netflix, Spotify, Microsoft 365, and in Sri Lanka, Dialog ViU and subscription-based online newspapers.

Sales as a Revenue Model

This is the most traditional revenue model where businesses earn money by selling products or services directly to customers. In e-business, sales happen through online shops and marketplaces.

Examples include Amazon, Daraz.lk, and Kapruka.com.

Advertising as a Revenue Model

Some companies offer free or low-cost services but earn income by displaying advertisements from third parties. Businesses with a large user base rely on this model.

Examples include Facebook, YouTube, Google, and Sri Lankan news sites like Daily Mirror Online.

Transaction Fee (or Commission) as a Revenue Model

Here, a company acts as a platform for transactions and charges a fee or commission for each completed transaction.

Example, PayPal charges for online money transfers, Daraz charges sellers a commission on sales, and ride-hailing apps like PickMe earn through commissions from drivers.

Advantages of E-Business

- Global reach – can sell to customers worldwide.
- 24/7 availability – always open for customers.
- Lower operational costs – fewer physical stores/staff needed.
- Provides customer data for better decision-making.
- Convenience and wide variety for customers.

Disadvantages of E-Business

- Security risks – hacking, fraud, identity theft.
- No personal interaction – cannot touch/see products before buying.
- Depends on internet access and digital literacy.
- Very high competition online.
- Delivery and logistics challenges.

E-Commerce Trends

- Becoming more social – sales through Facebook, Instagram, TikTok.
- Mobile-based (M-Commerce) – shopping and payments through apps.
- Personalization C AI – product recommendations, targeted ads, chatbots.
- Omni-channel integration – combining physical stores + online platforms.
- Digital payments – mobile wallets, QR codes, cryptocurrencies.
- Subscription models – Netflix, Spotify, Microsoft 365.

E-Marketing

E-Marketing, also known as online marketing or internet marketing, refers to using digital platforms and electronic media to promote products and services. This includes activities such as promotions, advertising, public relations, and after-sales support through the internet.

For example, many Sri Lankan businesses send email newsletters with product catalogs and special offers that directly influence customers' purchasing decisions. Social media platforms like Facebook, Instagram, and TikTok are widely used by companies such as ODEL, Singer, and Keells to announce promotions, introduce new arrivals, and interact with customers. Social media also acts as a digital form of word-of-mouth marketing, where customers share, like, and comment on posts, spreading awareness to wider audiences. Online advertising campaigns, such as those run by Dialog Axiata or PickMe, also help firms to instantly reach thousands of customers.

The main advantages of e-marketing are that businesses can quickly adapt to customer feedback, make an instant impact, directly target specific market niches (e.g., youth, professionals), and measure responses in real-time through analytics tools.

Mobile Marketing

Mobile marketing is a powerful extension of e-marketing that uses smartphones and mobile devices to reach customers. Since most Sri Lankans now browse the internet through their phones, businesses are shifting from desktop-based to mobile-first marketing strategies.

Firms use various mobile channels such as SMS, WhatsApp, Viber, Facebook Messenger, and mobile apps to send promotional messages and personalized offers. For example, Sampath Bank and Commercial Bank send SMS alerts and promotional loan offers directly to customers' phones. Food delivery services like PickMe and Uber Eats use push notifications on their apps to offer discounts and attract customers during lunch or dinner hours.

A key feature of mobile marketing is location-based services. For instance, local restaurants in Colombo can use geoadvertising to promote lunchtime offers only to customers within a certain distance. Similarly, Daraz and Keells online shopping apps send personalized app notifications for special sales. Mobile marketing is closely linked to m-commerce (mobile commerce), where customers directly shop using mobile apps. In Sri Lanka, most banks, insurance firms, and telcos such as Dialog and Mobitel have mobile apps that allow bill payments, fund transfers, and service requests conveniently.

Database Marketing

As more people engage in online activities, businesses collect large amounts of customer data, including personal details, browsing habits, and preferences. Database marketing refers to using this data to target customers with personalized offers.

For example, an online store like Daraz.lk collects information on what products customers browse or add to their wishlist. Later, they send personalized emails or app notifications reminding customers about these items or offering discounts to encourage purchase. Similarly, PickMe uses ride history data to send customers special coupons for their frequently traveled routes.

Database marketing is often combined with data mining techniques, which help businesses understand customer behavior. For instance, by analyzing which products students purchase around the exam season, Vijitha Yapa Bookshop's online store or MD Gunasena's online portal can promote relevant books directly to them. This makes marketing more effective than sending the same generic message to everyone.

Unit-13(New Trends & Future Directions of ICT)

Today ICT is shaping our lives faster than ever. From communication to business, almost every activity depends on ICT. Because of this, scientists and engineers are constantly researching new technologies and how they can be applied in real life. To study how these new technologies grow, researchers often use a tool called the **Gartner Hype Cycle**, which shows the journey of a new technology from its beginning until it becomes widely used.
According to this model, new technologies pass through five stages:

1. Innovation Trigger – the first introduction of a technology.
2. Peak of Inflated Expectations – people expect a lot, sometimes more than the technology can deliver.
3. Trough of Disillusionment – failures and limitations reduce excitement.
4. Slope of Enlightenment – success stories appear and realistic uses are discovered.
5. Plateau of Productivity – technology becomes mature and widely adopted.

Some of the key emerging technologies identified in recent hype cycles include **Artificial Intelligence (AI), Blockchain, Internet of Things (IoT), Quantum Computing, and Smart Robots**.

Intelligent and Emotional Machines

The idea of intelligent and emotional machines is not new; it has been imagined in myths, movies, and science fiction for centuries.
For example, the car "KITT" in the TV series *Knight Rider* could talk and even show emotions.

Modern intelligent machines are designed to think, respond, and sometimes even recognize emotions. Emotional computing allows machines to detect human feelings through speech or facial expressions and respond accordingly. Although people have long built automatic machines called "automatons" (that perform tasks by themselves), the real progress in artificial intelligence only began in the mid-20th century.

Artificial Intelligence (AI)

AI is the ability of machines to perform tasks that normally require human intelligence, such as understanding language, learning from experience, solving problems, and making decisions.
Researchers describe two main forms of AI:

- Weak AI (Narrow AI):
  Machines that are very good at one specific task but cannot perform outside that area. Examples include IBM's Deep Blue (chess-playing computer), Watson (question-answering system), and AlphaGo (a program that mastered the game of Go).

- Strong AI (General AI):
  A system that could think like a human and work in multiple fields. For example, a machine that can both drive a car and perform surgery. Researchers are still working on achieving this level, often called Artificial General Intelligence (AGI).

There is also the idea of **Artificial Super Intelligence (ASI)**, where machines could go far beyond human intelligence, but this remains theoretical.

AI Techniques

Some important methods used in AI include:

- **Search techniques** – finding solutions, e.g., winning moves in a game.
- **Expert systems** – using stored knowledge in "if–then" rules to give advice.
- **Natural Language Processing (NLP)** – understanding and responding to human language.
- **Speech recognition** – converting spoken words into text.
- **Computer vision** – recognizing and interpreting images and videos.
- **Machine learning** – finding hidden patterns in data, e.g., predicting if a bank customer may default on a loan.
- **Neural networks / Deep learning** – powerful systems inspired by the human brain, good at image and voice recognition.
- **Genetic algorithms** – solving problems using "survival of the fittest" approaches.
- **Fuzzy logic** – controlling systems with flexible rules (e.g., "if the room is hot, run the fan fast").

Man-Machine Coexistence

One common fear is that intelligent machines may replace humans in jobs. However, the real goal of ICT and AI is **coexistence**, where humans and machines work together. Machines can handle repetitive or heavy tasks, while humans focus on creativity, emotions, and complex decisions.
For example, **Asimo**, a humanoid robot created by Honda, was designed to work with humans in workplaces and assist with tasks. This shows how man and machine can collaborate.

Machine-to-Machine Coexistence (M2M)

Apart from working with humans, machines are also learning to work with other machines. This is called **machine-to-machine coexistence**. It is a core part of the **Internet of Things (IoT)**, where devices communicate with each other automatically.
For example, smart electricity meters can send readings directly to the power company without human involvement. Cars can alert service centers about maintenance needs, or in healthcare, wearable devices can send health updates to hospital systems.

Software Agents

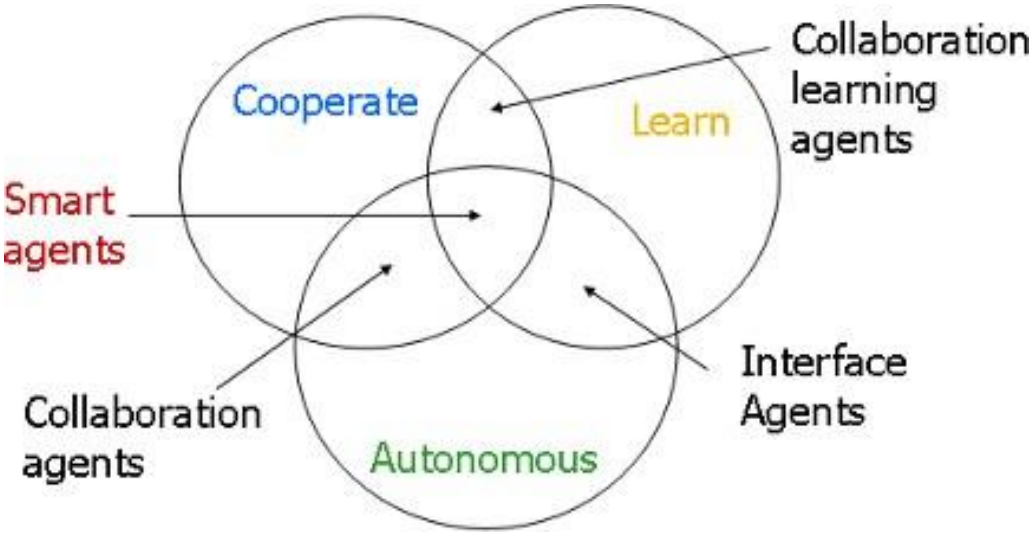
A software agent is a piece of software that works on behalf of a human user or another software application. Instead of the user doing everything manually, the agent can assist, automate tasks, and make decisions.

A good example is Cortana in Windows or Siri in iOS. These virtual assistants can search for files, recognize voice commands, set reminders, and even predict user needs. While they don't have "true" intelligence, they show weak AI abilities such as voice recognition, natural language understanding, prediction based on past behavior, and reasoning.

Key Features of Software Agents:

- Autonomous: can take decisions by themselves without the interference of the human user
- Proactive: can initiate actions on its environment
- Reactive: can react to events in the environment
- Cooperative: can cooperate with other agents and humans
- Able to learn: can learn by observation, experience, and data
- Social ability: can interact and communicate with others complying to the social rules and norms

These features actually help to differentiate a software agent from an object in software. A good classification of software agents is depicted in Figure below.



Multi-Agent Systems (MAS)

A single agent may not be powerful enough to solve complex problems. That's why sometimes multiple agents are combined to work together in what is called a **Multi-Agent System (MAS)**.

A **MAS** is a group (network) of agents that interact with each other to solve problems collectively. Each agent in a MAS is:

- **Autonomous** – self-aware and able to make independent decisions.
- **Limited in scope** – each agent only knows its specific task, not the whole system.
- **Decentralized** – no single agent controls the system; they work together in coordination.

For example, if a customer searches for airline tickets on a booking site, different agents will connect to different airline databases, gather results, and then send them to a broker agent. The broker filters and organizes the information and passes it to the **interface agent**, which shows the results to the user.

Applications of Agent Systems

Software agents are widely used today in many fields:

- **Virtual Assistants**: Cortana (Windows), Siri (iOS), and Google Assistant help users by managing reminders, answering queries, and searching information.
- **E-Commerce**: Agents search for products across multiple websites, compare prices, and recommend the best options to users (e.g., Daraz search filters, hotel/flight booking sites).
- **E-Learning**: Agents provide personalized recommendations to students, such as suggesting additional resources based on their progress.
- **Banking**: Agents can track transactions, detect fraud, and provide personalized financial advice through chatbots.
- **Online Booking Systems**: Travel and hotel booking platforms use multi-agent systems to automatically contact merchants, check availability, and confirm reservations.

What's an agent diagram?

A quick sketch that shows software agents (and any humans/systems they work with), their roles, and who talks to whom (messages/requests/results). It's perfect for Multi-Agent Systems (MAS) like search/booking/price-comparison apps.

- Agent = rounded rectangle (write its role inside)
- Human user = stick figure or rectangle labeled "User"
- External system = rectangle with double vertical edges
- Data store = cylinder
- Message = arrow with a short verb–noun label (e.g., "send query", "return results")

Step-by-step method

1. **State the goal**
   Write one line at the top:
   "Goal: find the best deal and place an order" (or whatever your scenario is).
2. **List participants**
   - Humans (e.g., User)
   - Internal agents (e.g., InterfaceAgent, BrokerAgent, RecommenderAgent)
   - External agents/systems (e.g., MerchantAgent A/B/C, PaymentGatewayAgent)
   - Data stores if needed (e.g., ProductDB, ProfileDB)
3. **Assign roles**
   Next to each agent, jot a tiny note: *collect input*, *aggregate offers*, *rank & filter*, *process payment*, *notify user*.
4. **Lay out layers left→right**
   - Left: User
   - Middle: your core agents (interface/broker/recommender)
   - Right: external agents/systems (merchants, gateways)
     This left-to-right flow keeps arrows tidy.
5. **Draw the boxes**
   Use rounded rectangles for each agent; label clearly (e.g., *BrokerAgent (A,P,C)* to show abilities).
6. **Connect with arrows (name each message)**
   - From User → InterfaceAgent: "search criteria"
   - InterfaceAgent → BrokerAgent: "normalized query"

What is an Autonomous Agent?

An agent that can work without user interference is called an autonomous agent. Autonomous agents operate independently, making decisions and performing tasks without needing continuous human control or input.

Beyond von Neumann Computers

Most of today's computers are designed using the von Neumann architecture. This model separates the computer into components such as the processor, memory, input, and output. It has been very successful for decades and is still the foundation of modern computers.

However, as technology advances, the limitations of von Neumann computers are becoming more visible. According to Moore's Law, the processing power of microprocessors doubles every 18 months, but this cannot continue forever because:

- Packing more transistors generates too much heat, which is difficult to manage.
- There are physical limits to how small transistors can become.

Because of this, scientists are now researching new models of computing to go beyond the limitations of von Neumann systems. Two of the most important approaches are Quantum Computing and Nature-Inspired Computing.

Quantum Computing

Traditional computers store and process data using bits (0s and 1s). In contrast, quantum computers use qubits, which can exist in multiple states at once due to the principles of quantum physics.

This ability allows quantum computers to perform calculations much faster and more efficiently than normal computers, especially for very complex problems like cryptography, weather forecasting, or drug design.

- Example: While a traditional computer tries possible solutions one by one, a quantum computer can try many solutions at the same time.

Thus, quantum computing is seen as a future technology that can break through the limits of current processors.

Nature-Inspired Computing

Scientists are also inspired by nature when creating new computing techniques. By observing how living organisms behave, process information, and adapt to their environment, researchers develop algorithms that can solve human problems more effectively. This field is called nature-inspired computing or bio-inspired computing.

Examples:

- **Neural Networks –**
  Inspired by the human brain, used in AI for image recognition, speech recognition, etc.

- **Swarm Intelligence –**
  Inspired by how ants, bees, or birds work together. For example, bees collectively decide the best location for their hive. In computing, this is used for routing strategies (like Google Maps traffic optimization), simulating crowd movements, and team-building algorithms.

- **Evolutionary Computation (Genetic Algorithms) –**
  Based on the principle of "survival of the fittest." Used in optimization problems where the best solution is gradually evolved.

- **Artificial Immune Systems –**
  Inspired by the human immune system to detect abnormalities, e.g., in computer security systems.

- **Membrane Computing –**
  Inspired by the functioning of living cells, where communication and processing happen inside and between cells.

- **Intelligent Communication Systems –**
  Designed by studying natural patterns like how diseases spread, which helps in building faster and more robust communication networks.

Ubiquitous Computing

Ubiquitous computing (also called pervasive computing) is a model of computing where technology is embedded everywhere around us and becomes a natural part of daily life. The word "ubiquitous" means *present everywhere*. The idea is that computers are not limited to desktops or laptops but are integrated into everyday objects and environments in such a way that people may not even realize they are using a computer.

The concept was first introduced by Mark Weiser in the early 1990s. His vision was that computers should "disappear into the background" and allow people to focus on their tasks instead of the technology itself.

Examples of Ubiquitous Computing

- **Smart Homes:** Smart bulbs, fans, and appliances controlled by mobile apps
- **Wearables:** Smartwatches like Apple Watch