

CLASS NOTES OF “FUNDAMENTAL OF IT & C PROGRAMMING ”



Technical Classes

✓ Technical Classes के Course में उपलब्ध फीचर्स

1. सभी कक्षाएं स्मार्ट बोर्ड पर Live
2. विशेषज्ञ शिक्षको द्वारा पठन पाठन
3. Lecture के दौरान डाउट सॉल्विंग
4. Recorded Lecture (VOD)
5. प्रत्येक क्लास का PDF नोट्स
6. SBTE Exam की तैयारी के लिए ब्रह्मास्त्र क्लास
7. Laptop में क्लास देखने की सुविधा

Call/WhatsApp - 93347 89450 / 91555 63777

नोट :-

1. सभी ऑफलाइन कक्षाएं Technical Classes के कैंपस तथा सभी ऑनलाइन कक्षाएं Technical Classes के एप्लीकेशन पर चलेगी।
2. यह नोट्स टेक्निकल क्लासेस के स्टूडेंट्स के लिए है, तथा क्लास करने के बाद अधिक प्रभावी होगा।

By Er. Dhruv Sir

J) Theory Session Outcomes (TSOs) and Units: T2418105

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
<i>TSO 1a.</i> Describe the anatomy of the Computer System. <i>TSO 1b.</i> List the different Input and Output devices. <i>TSO 1c.</i> Identify the different types of memory in computer systems. <i>TSO 1d.</i> Explain communication process between different components of a computer. <i>TSO 1e.</i> Describe the functionalities of a computer system. <i>TSO 1f.</i> Use Internet digital Platforms	Unit-1.0 Basics of Computer System 1.1 Computer System and its Components. - Generation of Computer - Anatomy of Computer Systems - Input and output device - Motherboard - Peripherals - Backend and Front end of System Unit 1.2. Storage device in Computer System - Primary Storage - Secondary Storage 1.3. CPU Components - Register - Control Unit - ALU 1.4. Types of Bus - Address Bus - Data Bus - Control Bus 1.5 Search Engine - Introduction - Search Query - Applications of Internet Digital Platforms (BHIM, Digi-Locker, m-paravian, NPTEL etc.)	CO-1
<i>TSO 2a.</i> Compare various computer network topologies <i>TSO 2b.</i> Differentiate types of networks. <i>TSO 2c.</i> Compare internet and intranet <i>TSO 2d.</i> Explain IP addressing system. <i>TSO 2e.</i> Explain functions of Networking Devices.	Unit 2.0 Basic Network Concepts 2.1 Network Topologies Bus, Mesh, Star, Ring, Hybrid 2.2 Types of Computer Networks LAN, WAN 2.4 Internet & Intranet IP Addressing system and URL, Internet, Intranet, Comparison between Intranet & Internet 2.3 DNS - Introduction, Need - Domain Names & its types 2.5 Networking Devices (Types and use) Switch, Router, Gateway, Modem, Repeater, Wireless Access Point, NIC	CO-2
<i>TSO 3a.</i> Write Algorithm to solve the given problem. <i>TSO 3b.</i> Write simple 'C' program to solve the given arithmetic expressions <i>TSO 3c.</i> Write a simple 'C' Program demonstrating the given data type conversion <i>TSO 3d.</i> Write I/O Statements for the given data.	Unit 3.0 Basics of 'C' Programming and control structures 3.1 Fundamentals of algorithms: Notion of an algorithm. Pseudo-code conventions like assignment statements and basic control structures. 3.2 Flowchart: Flowchart, Symbols of flowchart, Guidelines for preparing Flowchart 3.3 Introduction to C: General Structure of a 'C' program Data Concepts: Character set, tokens,	CO-3

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
<i>TSO 3e.</i> Write a 'C' program using a decision-making structure for two-way branching to solve the given problem. <i>TSO 3f.</i> Write a 'C' program using a decision-making structure for multi-way branching to solve the given problem. <i>TSO 3g.</i> Apply loop statements to solve the given iterative problem in 'C' program. <i>TSO 3h.</i> Use appropriate statements to change the program flow in the given loop.	keywords, Identifiers, Variables, Constant, data types, C operators, Arithmetic operators, Arithmetic expression, declaring variables, and data type conversion. 3.4 Basic Input output: Input and Output statements, using printf() and scanf(), character input/output statements, Input/output formatting, Use of comments 3.5 Decision making and branching: Relational and logical operators, if statement, if else statement, nested if-else, if-else ladder' The switch statement 3.6 Looping: While loop, Do... While loop For loop, Go to statement, Use of break and continue statements	
<i>TSO 4a.</i> Write statements to read, write the given array. <i>TSO 4b.</i> Manipulate the given array of characters and numbers. <i>TSO 4c.</i> Use pointers to access memory locations for solving the given problem.	Unit 4.0 Array and Pointer 4.1 Characteristics of an array, One dimension and two-dimension arrays, Array declaration and Initialization 4.2 Array of characters, Operation on array Character and String input/output Concepts of pointers: declaring, initializing, accessing.	CO-4
<i>TSO 5a.</i> Use the given Library function. <i>TSO 5b.</i> Develop user defined functions for the given problem. <i>TSO 5c.</i> Write 'C' codes to pass the given function parameters using "call by value" and "call by reference" approach. <i>TSO 5d.</i> Write recursive function for the given problem.	Unit 5.0 Concept and Need of Functions Library functions: Math functions, String handling functions, other miscellaneous functions. Writing User defined functions, scope of variables. Parameter passing: call by value, call by reference. Recursive functions	CO-5

Note: One major TSO may require more than one Theory session/Period.

K) Suggested Laboratory (Practical) Session Outcomes (LSOs) and List of Practical: P2418105

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
<i>LSO 1.1.</i> install device driver. <i>LSO 1.2.</i> Install given software on your system. <i>LSO 1.3.</i> Perform Registration process of digital India platform.	1.	1.1 Identify specifications of various types of computer systems available in your institute. 1.2 Install Printer, scanner driver. 1.3 Install any two freeware or open-source software/tool by using web browser 1.4 Use Digital India Platforms: BHIM, Dig-Locker, m-parivahan, NPTEL.	CO-1
<i>LSO 2.1.</i> List various types of networking devices in your Institute. <i>LSO 2.2.</i> make a small local area network.	2.	2.1 Connect two/three computers to form a network using wire/wireless connectivity and configure it.	CO-2

Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24



[Electronics]
[F.I.T with C] → 2nd sem
↓
Fundamental of Information Technology
↓
[Computer]

TECHNICAL CLASSES

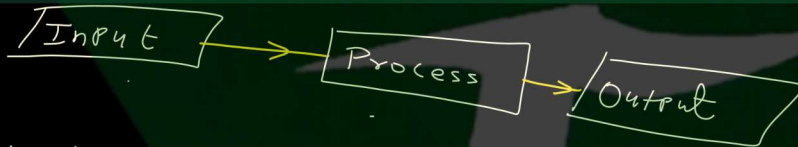
Computer :- It is an Electronic Device which takes input with the help of Input device Process it and gives result as Output with the help of Output device.

Input :- It is the Process in which we give any data, information from outside to a Computer system.



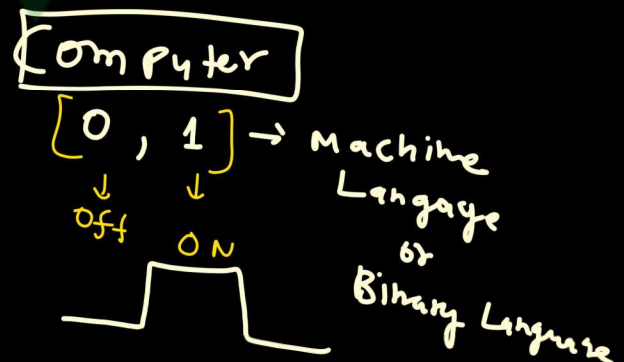
Input Device :- Such type of devices which helps us to give input are known as input devices. Such as

- 1> Keyboard
- 2> Mouse
- 3> Joystick
- 4> Scanner
- 5> Track ball
- 6> Light pen etc



1> Keyboard :- It is a standard input device which is used to take input from user. There are 102-104 keys on it. There are 12 function keys on it.

How a keyboard work ?



ANSI (American National Standard Institute)

It is a Standard Institute of America which has given a set of rules known as ASCII and informed all keyboard making Company that they will have to follow these rules otherwise their keyboard will be benched.

ASCII: [American Standard Code for Information Interchange]

It is the rule given by ASCII in which each and every character on a keyboard has a unique decimal value. e.g.

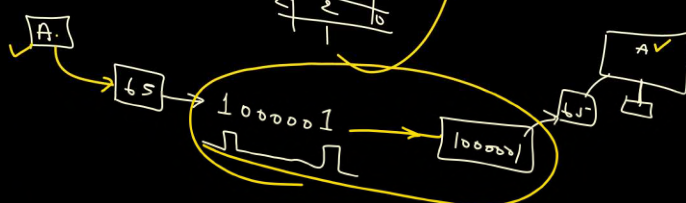
A → 65	a → 97
B → 66	b → 98
...	...
Z	

Decimal number	Binary number
[0 - 9] base 10	[0, 1] base 2
0 1 2 3 4 5 6 7 8 9	

Decimal to Binary Conversion

	2	65	1
2	32	0	
2	16	0	
2	8	0	
2	4	0	
2	2	0	
1	1	1	

1000001





64 32 16 8 4 2 1
1 0 0 1

A → 65 | a → 97

/

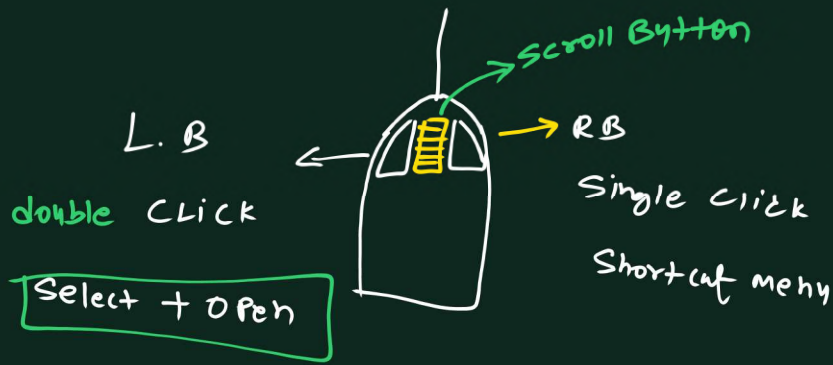
TECHNICAL CLASSES

[Good Evening]

[Happy New Year 2025]

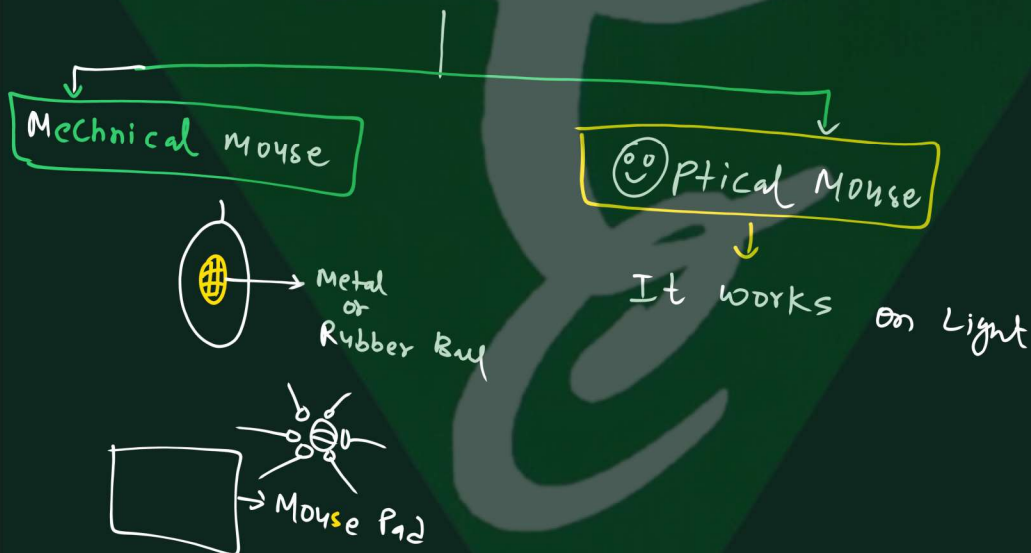
2nd sem

Mouse :- It is an input device
also known as Pointing device. Which is
used to Point & select any object on
Computer Screen.



TECHNICAL CLASSES

There are Two type of Mouse :-



[Good Evening]

Scanner :- It is an input device which is used to convert any physical object (Such as Text, Picture, Characters etc) into digital image so that it can be store in a Computer system for further work.

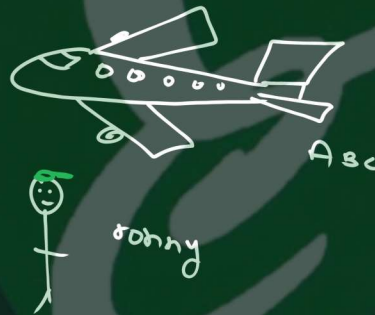


Types of Scanner:-

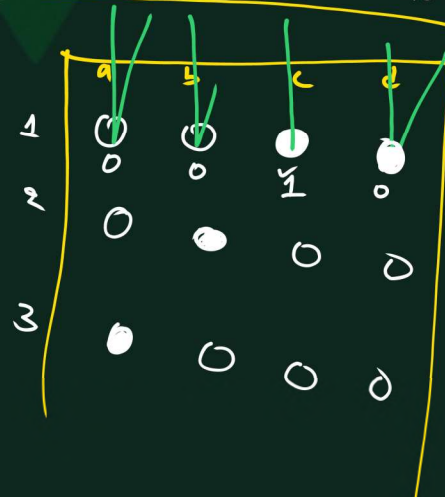
- ① OCR (Optical Character Reader)
- ② OMR (Optical Mark Reader)
- ③ MICR (Magnetic Ink Character Reader)
- ④ Barcode Reader

MICR :-

BANK :- Cheque ?



OMR (Optical Mark Reader)





TECHNICAL CLASSES

Input

Joystick :- It is an input device

which is mainly used for gaming purpose.

It works same as mouse but it is made up of a lever mounted on a base which can be move in all directions.

Track ball :- It is also an input device which works same as Mechanical mouse but it is fixed on a base.

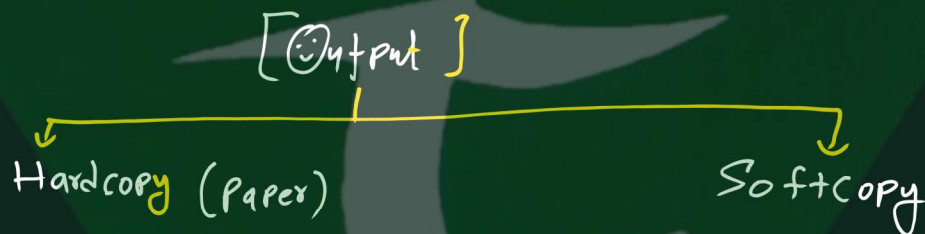


Output devices

Such types of devices which help us to take any output (result) from a computer system to outside is known as output device. eg:-

- ① Monitor
- ② Printer
- ③ Speakers etc

TECHNICAL CLASSES



Monitor :- It is an output device which is used to give any information as result in the form of Softcopy. There are following types of Monitor.

① CRT [Cathod Ray Tube]

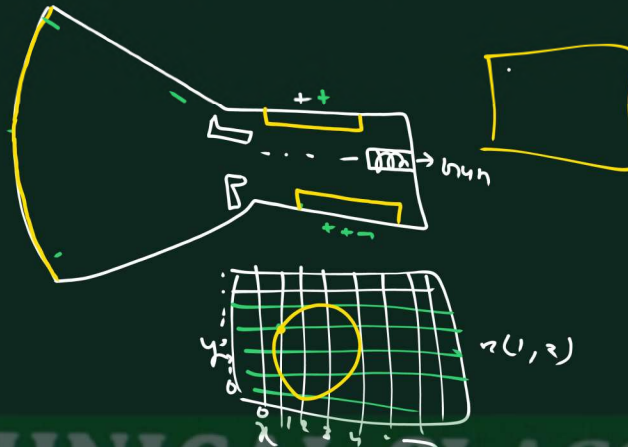
② LCD [Liquid crystal display]

③ LED [Light Emitting diode]

T.F.T

Thin film transistor

① C.R.T (Cathod Ray Tube)



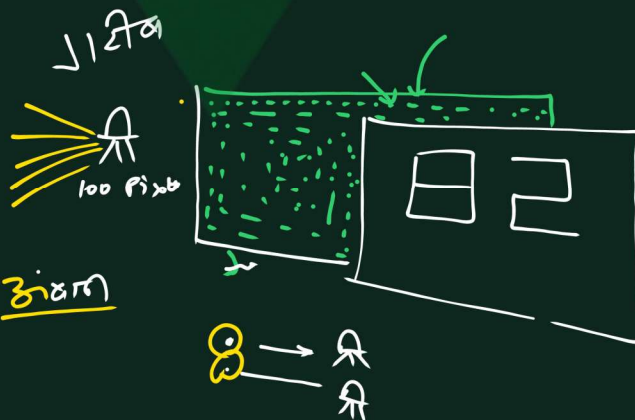
TECHNICAL CLASSES

Recap

[Good Evening]

Monitor
C.R.T

L.C.D :

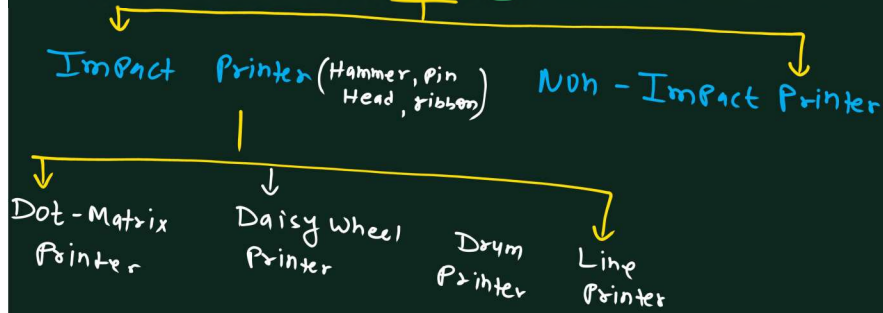


L.E.D :-

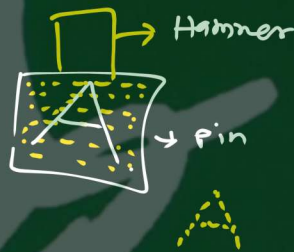


Printer :- It is used to take any information from a computer system as output on a paper.

[Printer]



① Dot Matrix Printer



2. > Daisy Wheel Printer





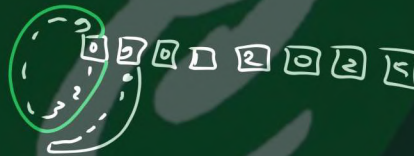
③ Drum Printer

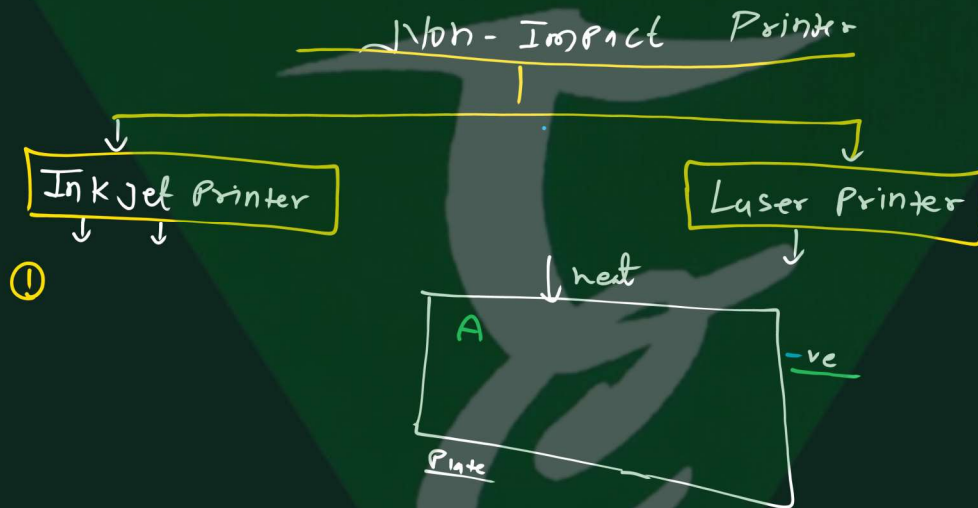
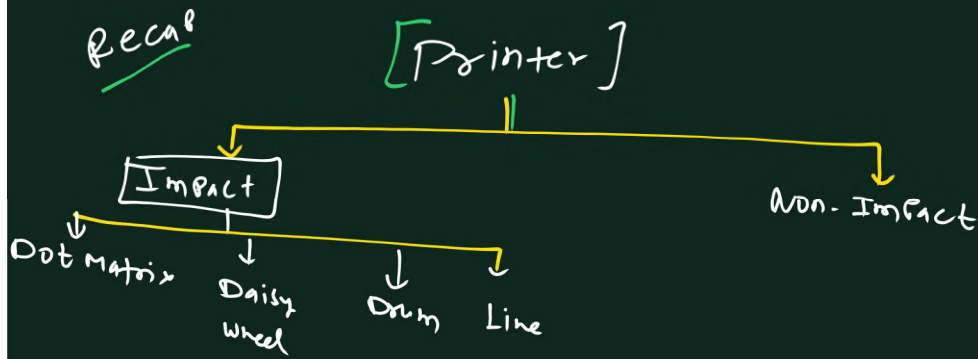
a	a	a	a	a	a	a	a
b	b	b	b	b	b	b	b
c	c	c	c	c	c	c	c

Line Printer

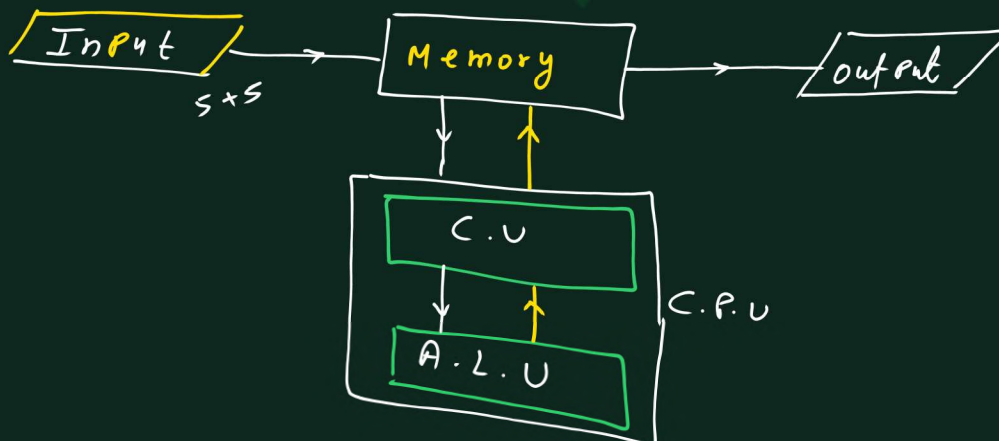
a	a	a	a	a	a	a	a
b	b	b	b	b	b	b	b
c	c	c	c	c	c	c	c

a p p l e





[Block Diagram of Computer]



Unit - 1.0

1.2 :- Storage devices in Computer system
or
Memory

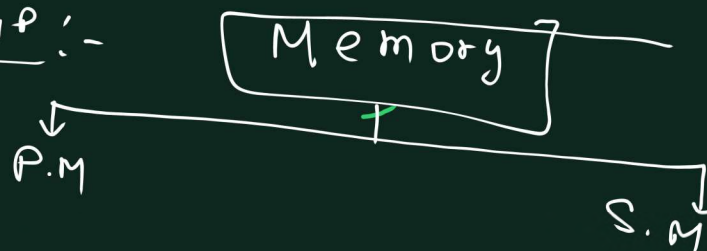
Memory: It is a Place where a Computer stores any instructions, information, data for a short time or for long time for future use.

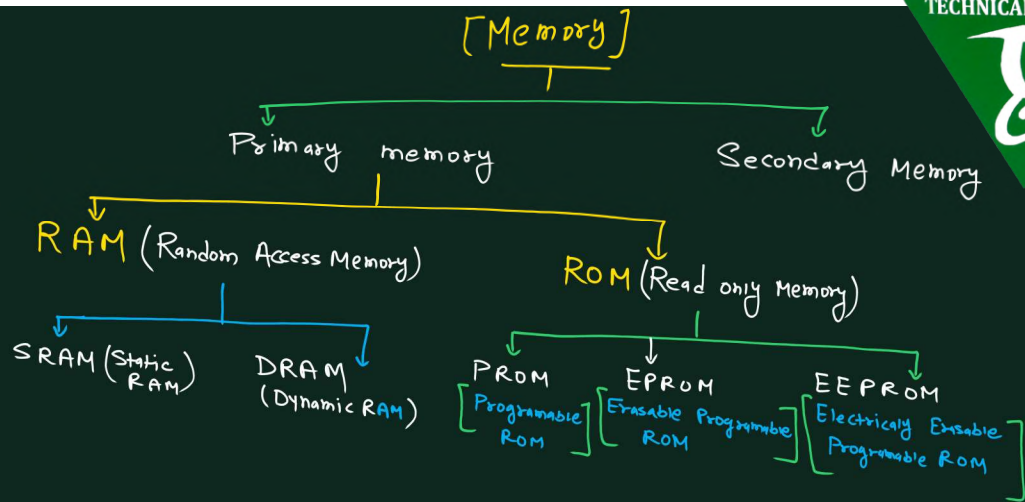
There are Two types of Memory



[Good Evening]

Recap :-

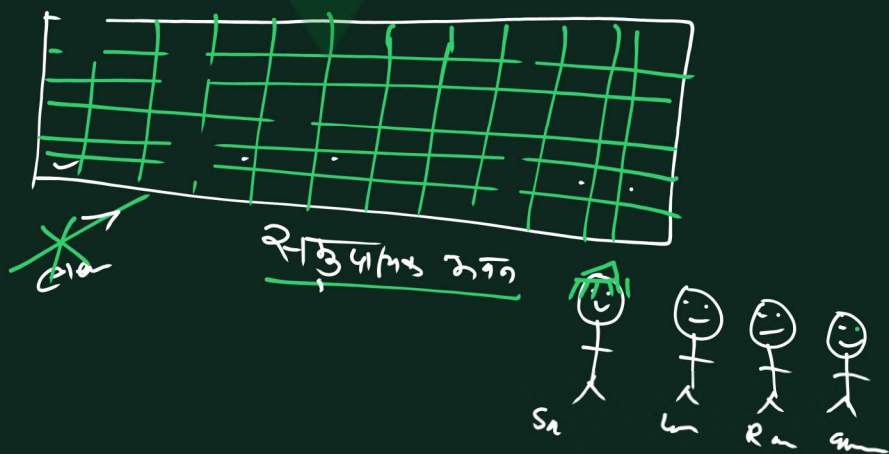




TECHNICAL CLASSES

RAM (Random Access Memory) : It is a

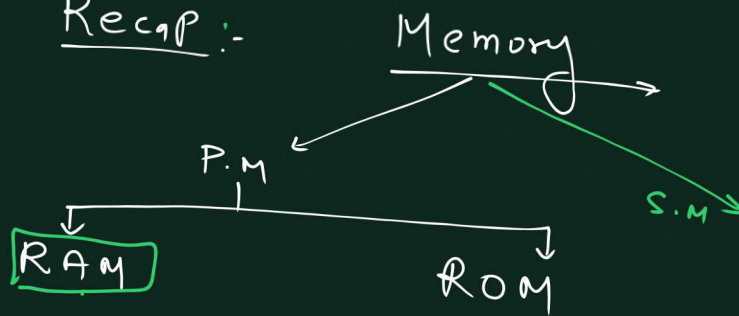
Primary memory of our Computer system made up of Semiconductor Materials. Any input first takes place on RAM. It is a volatile memory because as and when electricity turns off it deletes its content. It is fastest memory which can store some GB of information.





[Good Evening]

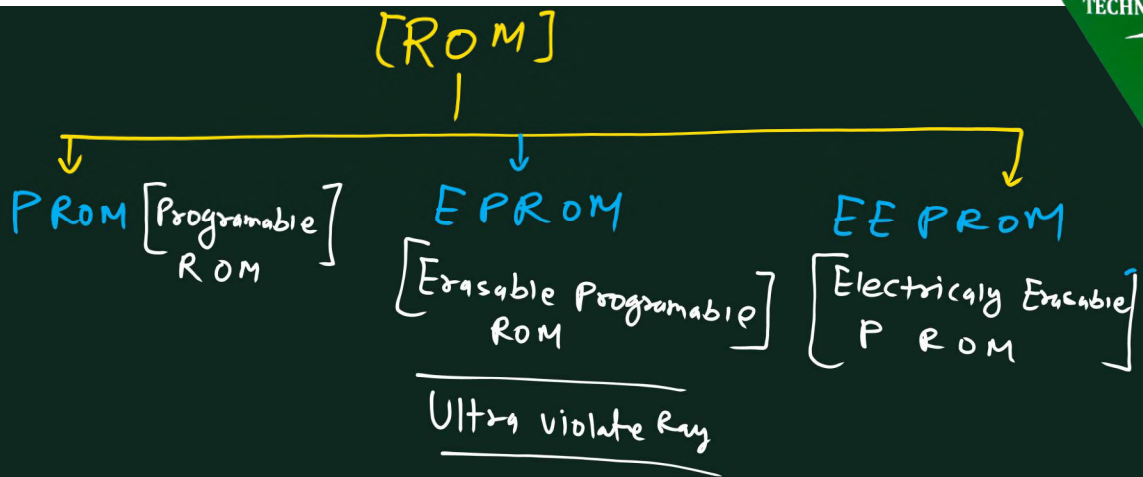
Recap :-



TECHNICAL CLASSES

Primary Memory :- It is the memory space in our computer system where any input firstly takes place.

ROM (Read only Memory) :- It is also a primary memory but it is Non-volatile in nature because it holds basic Input output Programs which is essential to open our computer system.



TECHNICAL CLASSES

Q.→ Which of the following is a Secondary memory ?

- a.→ RAM
- b.→ ROM
- c.→ EPROM
- d.→ None ✓

Q.→ Which is a volatile memory ?

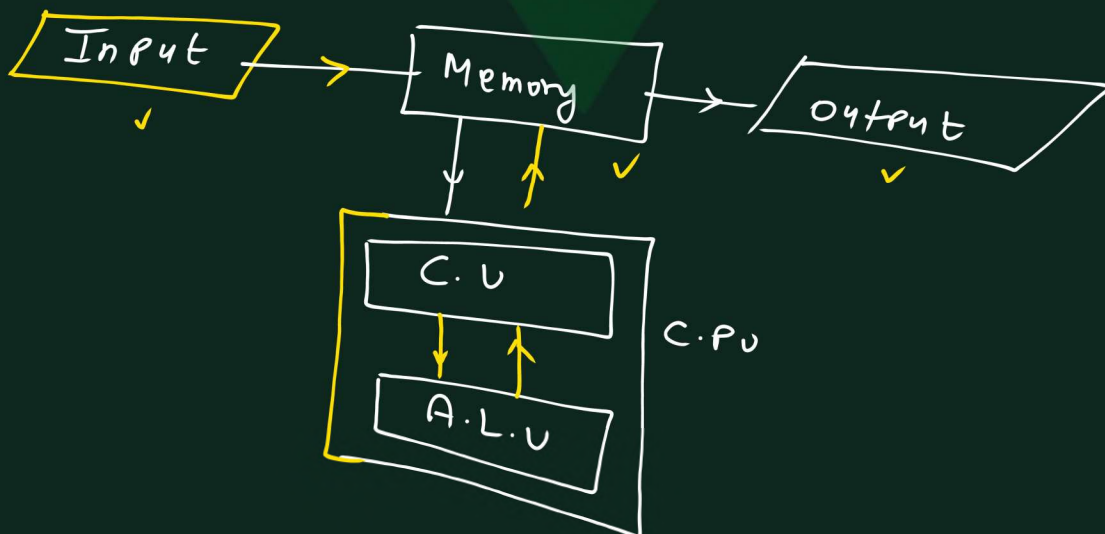
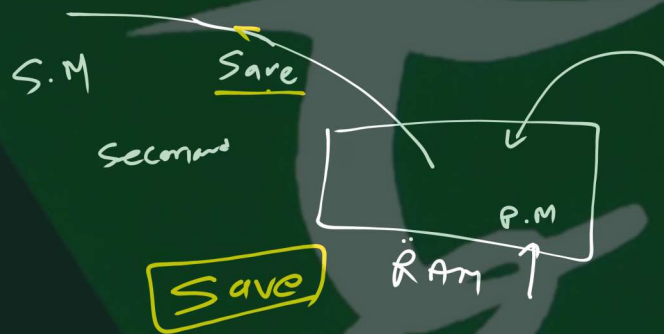
- a.→ ROM
- b.→ RAM ✓
- c.→ Harddisk
- d.→ None



[Secondary Memory]

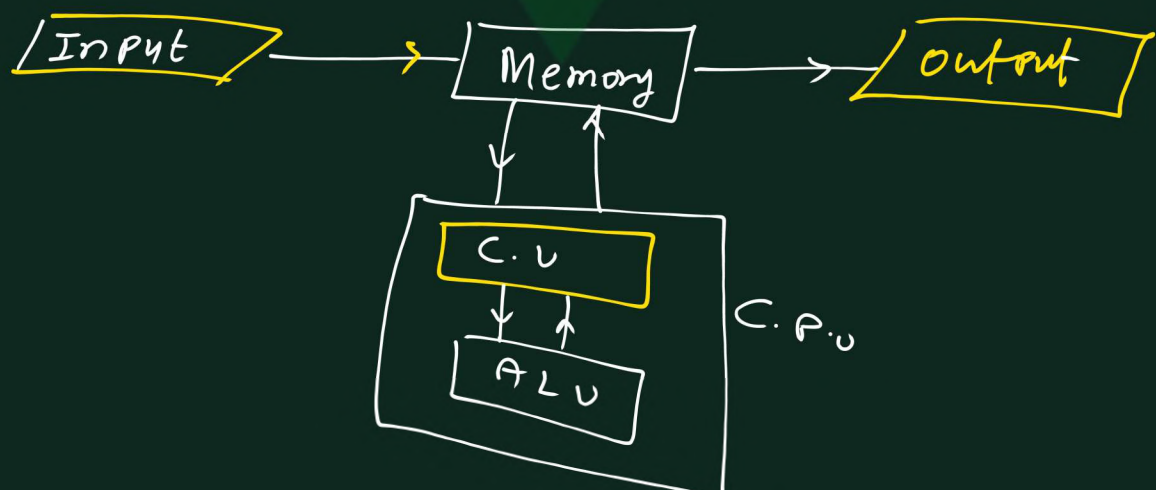
It is Permanent memory of our Computer System where we store data, information for future use. Example:- Harddisk, Pen drive, CD, DVD, Magnetic tape etc

TECHNICAL CLASSES





TECHNICAL CLASSES





TECHNICAL CLASSES

CPU (Central Processing Unit)

It is the Primary Component of a Computer system that performs most of the Processing inside a Computer. It is made up of Silicon material and can be divided into Three Parts.

- a. → C.U c. → Register
- b. → A.L.U

It is Also known as Brain of Computer.

C.U [Control Unit] :- It is the Part of

CPU which is responsible for managing and directing the Operations of a Computer System. It works same as a manager which



doesn't perform any task by self but it **directs** others to do their task. CU is also known as **Heart** of a computer because it pumps instructions to all parts of a computer.

TECHNICAL CLASSES



Q.7 Which of the following is **known as Heart** of a computer?

a. > ALU

b. > CPU

c. > CU ✓

d. > User

Q. Which is known as Brain of a Computer?

a) C.U

b) ALU

c) CPU ✓

d) None

TECHNICAL CLASSES

A.L.U (Arithmetical logical Unit) ∴ It is very important Part of a CPU in which all types of arithmetic (+, -, ×, /) and Logical (>, <, >=, <=, =) operations are performed.

Register ∴ It is a small high speed storage location within the CPU. It is mainly used to hold data, instructions, or memory address that CPU is currently processing. So that our CPU work fast. A register can store some bit of data only.



TECHNICAL CLASSES

Unit - 1.1

[Generation of Computer]

The generation of Computer refers to the evolution of Computer technology over time.

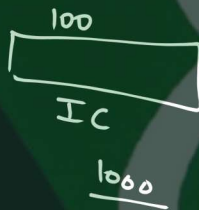
It is mainly categorized in Five generations, based on their technology.

1st Generation (1940 - 1956) :- Vacuum Tube



Computer Generation

1st	2nd	3rd	4th	5th
Vacuum Tube	Transistor $\frac{1}{100}$ 1000 fast	IC Integrated Circuits	LSI, VLSI Large Scale I.C → 1000 →	A.I



Antanomy of Computer System

Antanomy of a computer system means different parts that makes a computer system. A computer consists mainly two parts.

[Compute]

Hardware (हार्डवेयर)

→ It is the working part of a computer.

→ It can be seen or touched

Eg: CPU, Motherboard, RAM
HDD/SSD, Input/output unit
etc.

Software (सॉफ्टवेयर)

→ It gives instruction to hardware to perform task

→ It cannot be seen only we can feel it.

Eg:- System Software (Operating system)
Application Software
(game, Antivirus, M.S word etc)

Motherboard :-

The motherboard is the main (single) circuit board in a computer system that connects all hardware components. It allows communication between CPU, RAM, Storage devices and other peripheral devices.

Key Component of a motherboard

1> CPU socket :-

2> RAM slot

3> Power connectors

4> Storage connectors

5> USB Ports

6> Audio Ports

7> Networking Ports

8> PCI Slots (Graphic cards)



Peripherals

It is any external or internal device that connects to a computer but it is not the part of core system. It means **Peripherals** helps to extend computer's functionality.

Example :

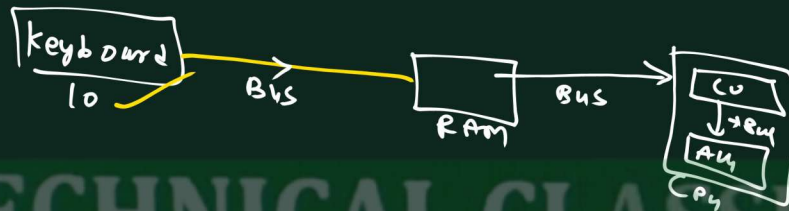
Input Peripherals :- Keyboard, Mouse, Joystick, Webcam
Scanner etc

Output Peripherals :- Monitors, Printers, Speakers etc

Storage Peripherals :- USB, External Hard disk, memory card etc

Bus : It is a Communication System

that transfers data between different Components of a Computer system.



It consists of physical connections such as wires, circuits, optical fiber etc that allows data transfer.

Type of Buses in Computer

1. Data Bus : It is used to carry actual data between computer memory, CPU and peripheral devices. It is subset of system bus.

Key feature of data bus

* Bidirectional : It means it allows data to flow both to and from CPU and memory.

2. > Address Bus :- It transfers memory address that Processor uses to read and write. It mean it is a Pathway in Computer system which carries memory address from CPU to another component.

* It is Unidirectional.

3. > Control Bus :- This bus is used to send Control signals between peripherals so that they can Coordinate.

* It is Bidirectional.

[Search engine]

It is a tool or service available on internet that help us to search or find any information ^{or website} on internet by using some text, Picture etc. **Example**

Google, Bing, Yahoo, DuckDuckGo etc



Search Query : It is a word or phrase that we type in search engine to find information. A search engine process these queries and returns results.

TECHNICAL CLASSES

Application of Internet Digital Platform

① BHIM : (Bharat Interface for Money)

It is a digital payment app developed by NPCI (National Payments Corporation of India) to perform fast, reliable and secure digital transactions. So that it promotes Cashless transactions in India. It was launched in 2016.

* Digi-Locker :-

It is a cloud based digital platform launched by GOI ^{Govt. of India}. It allows a user to store, access & share official government issued document securely online.



MParivahan :-

It is a mobile application developed by Ministry of Road transport and highway Govt. of India. It provides digital access to vehicle registration Certificate, Driving Licence and other transport related service.

NPTEL [National Programme on Technology Enhanced Learning]

it is an online platform powered by IITs under the ministry of Education, Govt. of India that provides free high quality courses in engineering, Science and management.

Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24

Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24



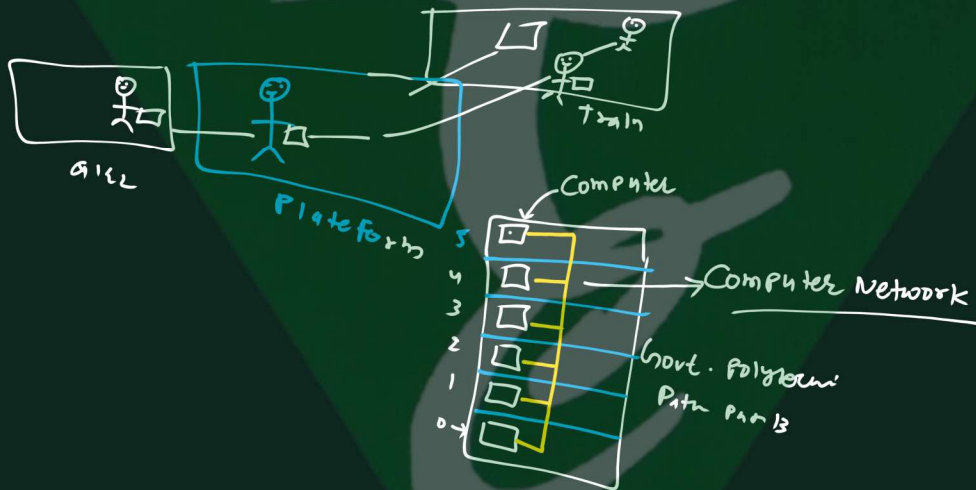
Unit-2

Basic Network Concept

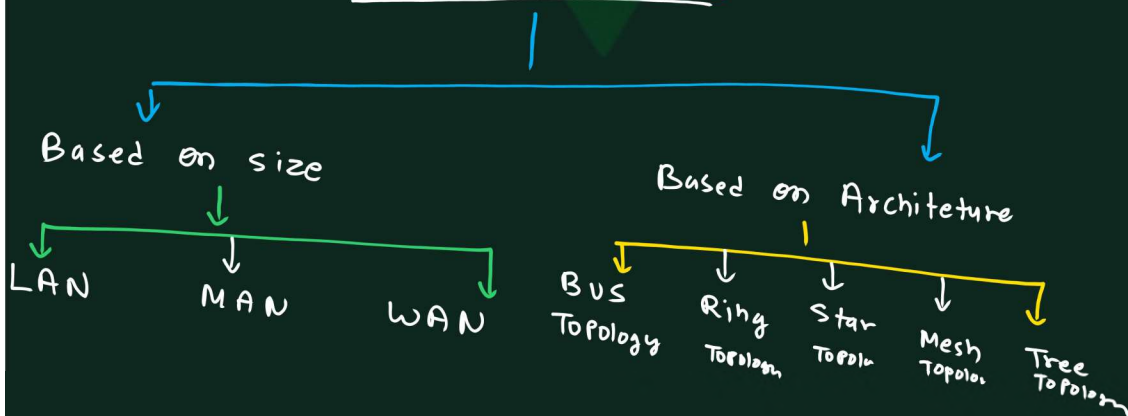


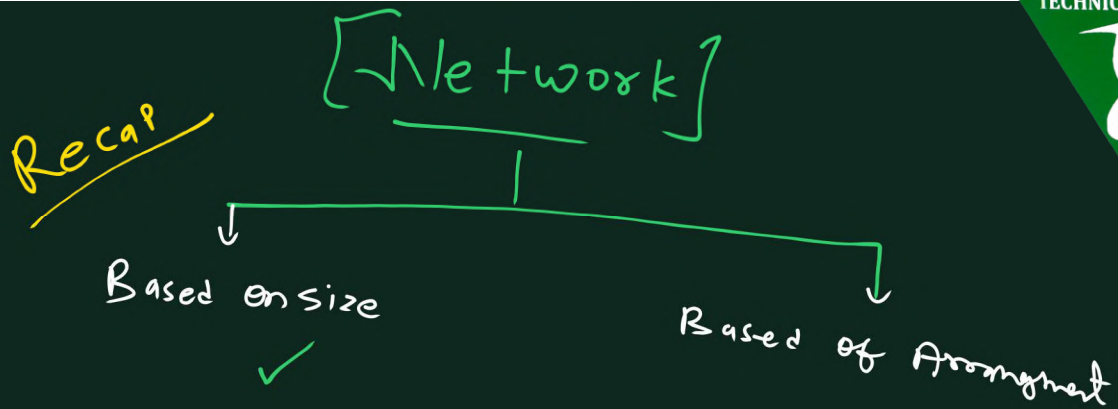
Network :- When two or more than two Computers or any networking devices are Connected with each other to Share their information with the help of Wire or WireLess Communication. It is known as Network.

अर्थ -> Network



Types of Network





TECHNICAL CLASSES

There are 3 categories of a Network based on size.

- i) LAN (Local Area Network)
- ii) MAN (Metropolitan Area Network)
- iii) WAN (Wide Area Network)

LAN (Local Area Network) :- It is a type of network which is used to connect two or more than two Networking devices such as Computer, Printer etc within a small geographical area (1 km). eg :- College campus, Office building etc.



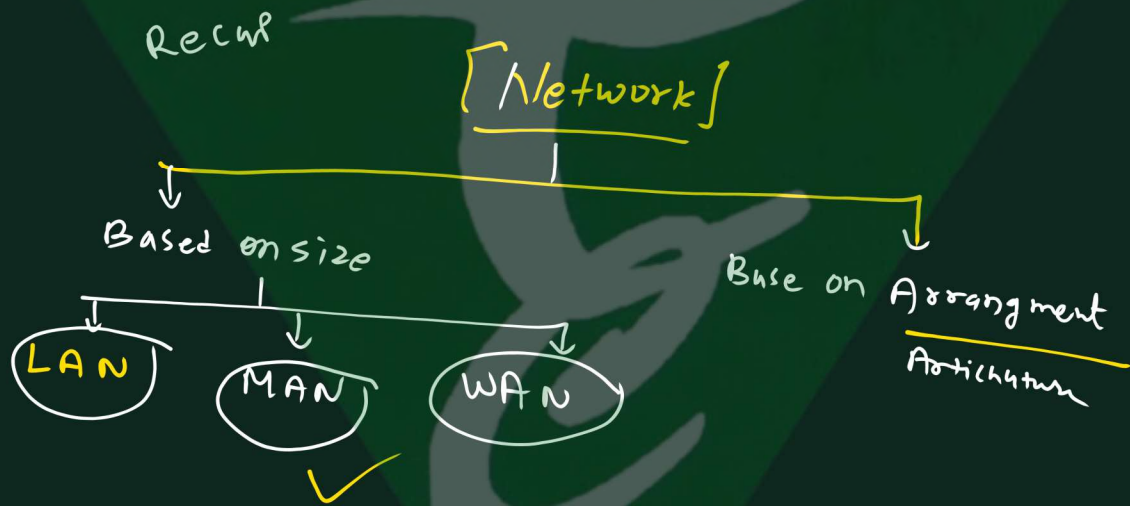
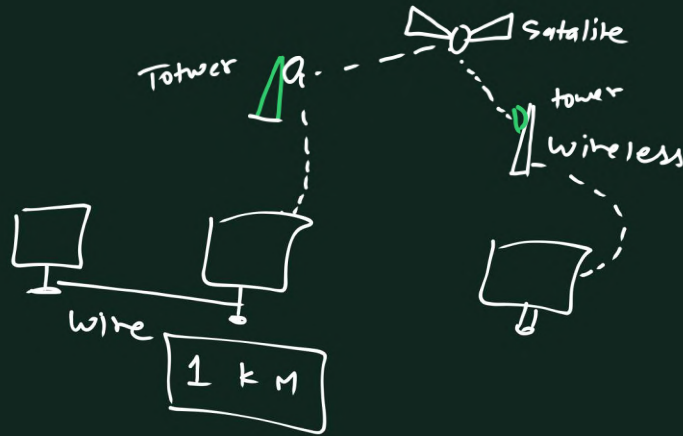
(ii) MAN (Metropolitan Area Network) :- It is also

a Network in which **two** or more than two **networking** devices are connected with each other to **Share** their resources (Files). It is **bigger** than LAN and **Covers** one or more than one cities. Such as **Bank** Network.



WAN (wide Area Network) :- It is such type of Network which covers a large geographical area such as multiple countries. It is bigger than **MAN**. eg: Internet





Network Topology :-

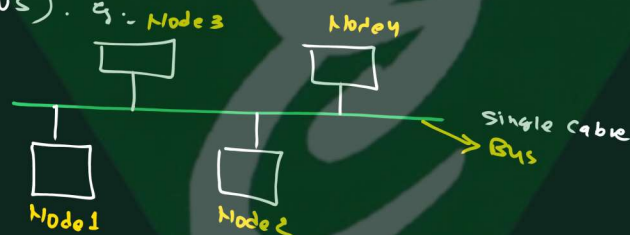
Network topology means an arrangement or structure of computers and other networking devices which are connected in a network. There are following types of topology.



- ① Bus Topology
- ② Ring Topology
- ③ Star ,,
- ④ Mesh ,,
- ⑤ Tree ,,
- ⑥ Hybride ,,

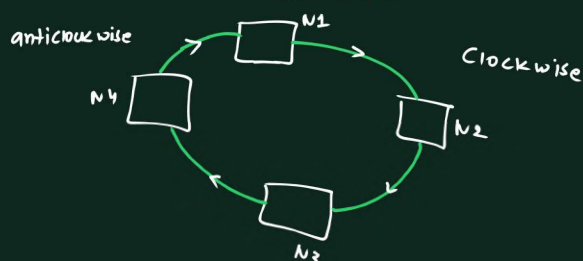
① Bus Topology

In Bus Topology all Networking devices are connected with a single Central cable (known as Bus). eg.:



② Ring Topology

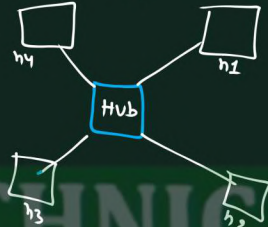
In this topology nodes are connected in Circular fashion and each device is connected with two other devices. eg.:





(iii) Star topology

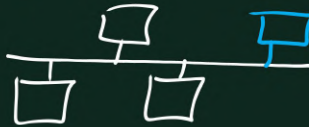
In star topology all nodes are connected to a central device known as Hub or Switch. Data from any device firstly reaches to this hub then it is forwarded. e.g.,





Recap :-

① Bus :-



② Ring :-



③ Star



iv) Mesh Topology

In this topology each and every networking devices(nodes) is connected with each other with a Separate Networking cable.

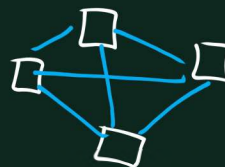


formula for no of wire used in Mesh topology

$$n \times (n-1) / 2 \quad n = \text{node}$$

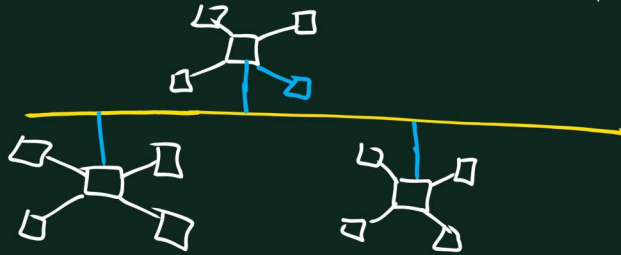
Q. There are 4 nodes in Mesh Topology find no of wires.

$$\begin{aligned} & 4 \times (4-1) / 2 \\ & = 4 \times 3 / 2 = 12 / 2 = 6 \end{aligned}$$



④ Tree topology

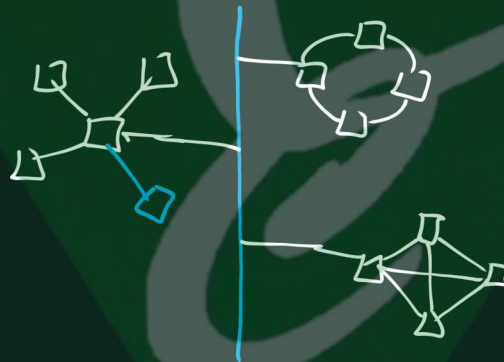
It is Combination of **Bus** Topology and Star topology and Looks Like a tree with branches.



TECHNICAL CLASSES

⑤ Hybride Topology

It is Combination of two or more **than** 20 networking Topology.



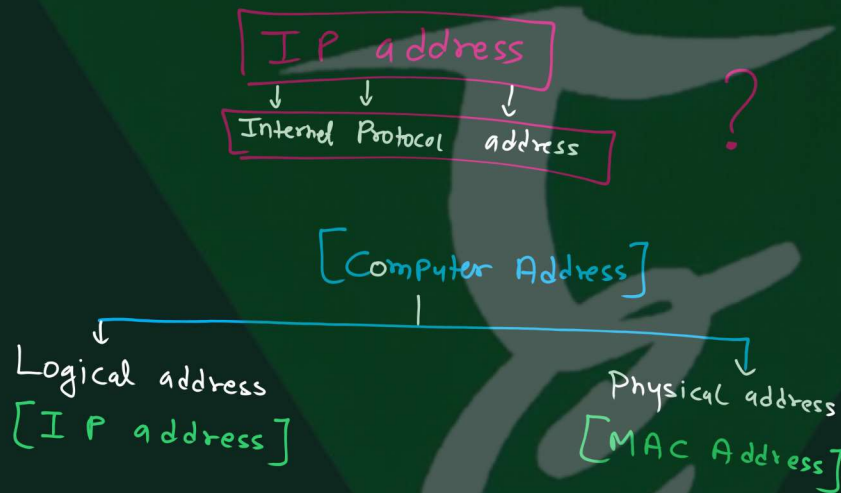
[Internet & Intranet]

Internet :- It is network of networks

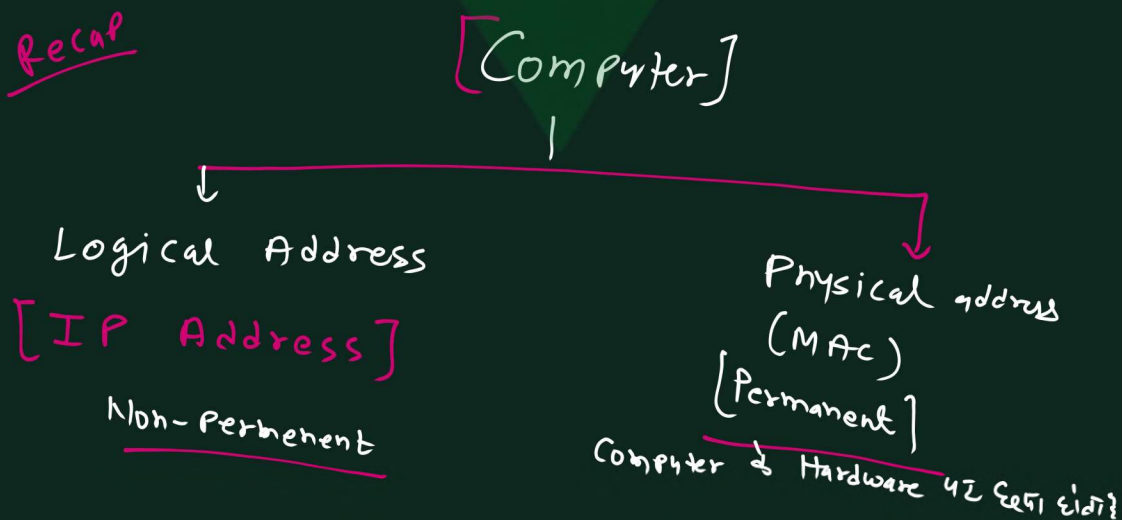
it means it is a global network of computers that Communicate with each other using Standard Protocols. It allows users to access and Share information through email, browser, **video**, text etc.



Intranet :- It is a Private network used in an organization, such as Bank, School, college or govt. agency etc. It works like internet but it is restricted to authorized user only. It is a secure network and data is prevented from any theft.



Recap





IP Address

An IP address [Internet Protocol] is a unique numerical label given to each and every devices (computer, mobile etc) connected to a computer network or Internet. It is a logical address which can be changed. There are two types of IP Address.

$$\dot{I} > I_{PV4}$$

ii) IPv6

IPV6

→ It's Length is 32 bits

It's Length is 128 bits

→ It is divided in 4 octet
binary number
separated by a decimal point.

It is divided in 8 octet
of Hexadecimal numbers

→ Range is from 0 to 255.

Range from 0 to FFFF (65536)

⇒ Only 4 billions address can be generated.

⇒ 340 trillion (2^{120}) address can be generated.

eg:- 192.168.2.6

C9: 3F8B:1806:4552:100:2AF6:21:4FF:an

$0.255 \cdot 0.255 \cdot 0.255 \cdot 0.255$
 $0 \cdot 0 \cdot 0 \cdot 0$
 $1 - 0.255$
 0.255

0-9

0-1

0 — 15151515

0 - 9 A B C D E F

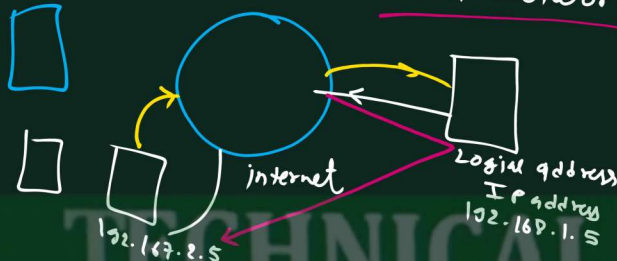
↓ ↓ ↓ ↓ ↓ ↓
10 11 12 13 14 15

[F.I.T.C.P]

Recap:-

[I.P address] :- Internet Protocol

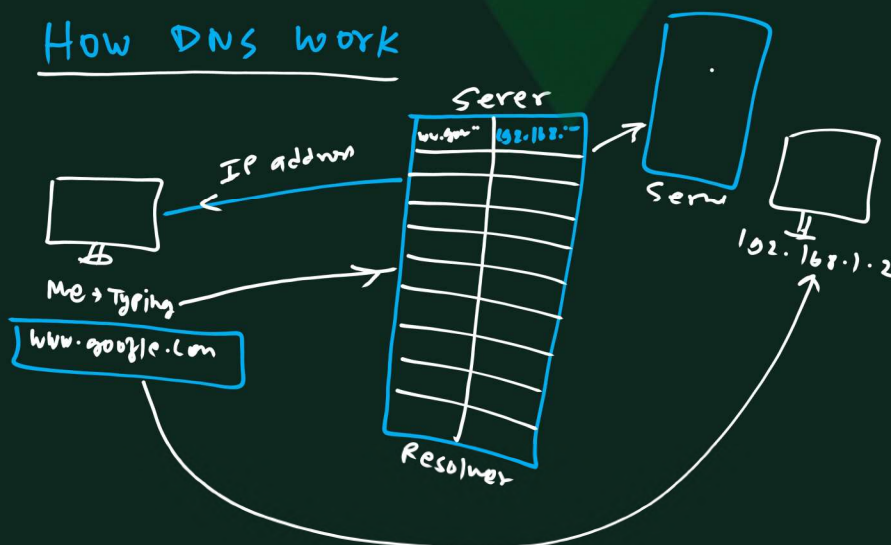
Logical address → on a Network



D.N.S [Domain Name system]

It is a decentralized system that translates human readable domain names for example (www.google.com) into IP address (192.168.1.5). So that it allows a computer to easily locate on internet.

How DNS work



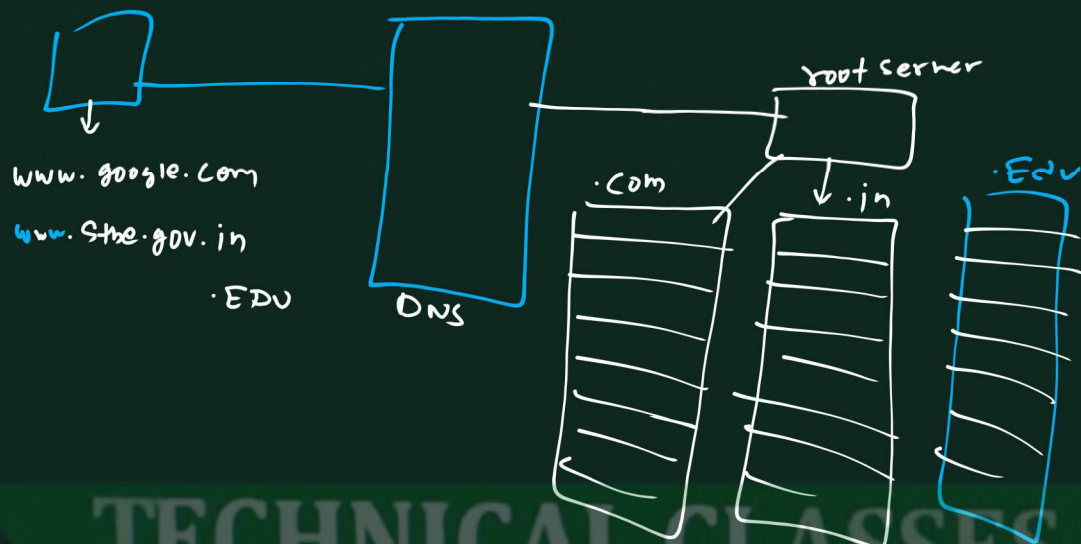
Types of DNS Server

DNS operates through a hierarchical system and there are different types of DNS servers. Each having a unique purpose. These are;

① Resolver DNS : It is the first step in DNS Lookup Process. It receives queries from Client Computer and finds the address by asking other DNS.

2.7 Root servers :- In these servers DNS names are stored according to their types for example .com, .edu, .gov, .in. etc. There are 13 root servers.

Letter address :- Fajrwa kumar, S/o dhekhwa
Rajirngel, Patna, Bihar → root server



* Networking Devices *

Networking Device is a Hardware that connects two or more than two computers or peripherals on a Network. There are following types of Networking devices.

1. > Switch
2. > Router
3. > Gateway
4. > Modem
5. > Repeater
6. > NIC



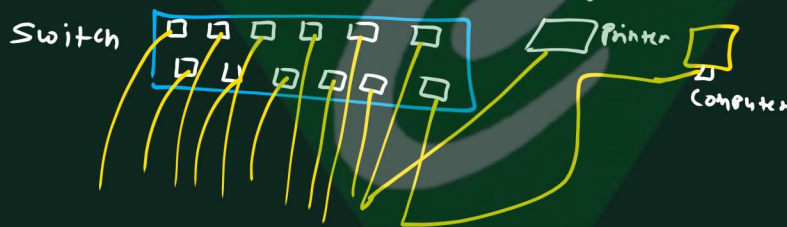
[F.I.T.CP]

Recap :- Network → DNS ↓

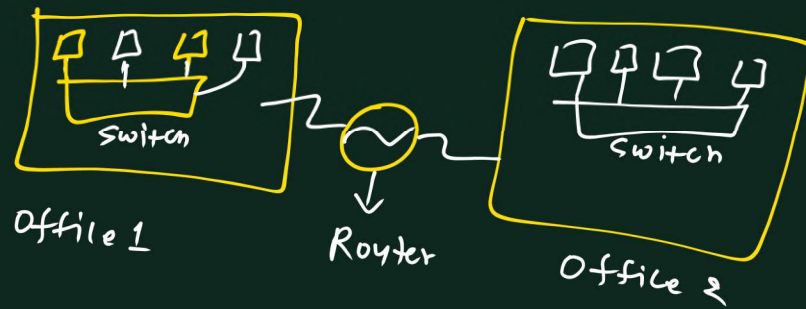
Networking devices

TECHNICAL CLASSES

① Switch :- It is a very important Networking device used to Connect multiple devices within the Same LAN. Such as Computer, Printer, server in a office.



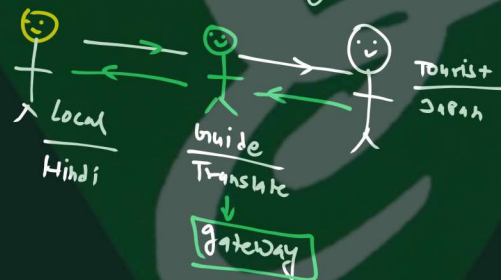
Router :- It is a critical networking device that is used to Connect two or more than two different Networks and forwards data between them.



TECHNICAL CLASSES

Gateway : It is a networking device or software that acts as a bridge between two different

Networks when both are using different protocols.



MODEM (Modulator-Demodulator) :- It is a Networking device that enables digital data to be transmitted

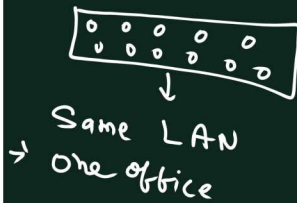
over analog Communication Line such as phone line.



[Good Morning]

Recap :- Networking Devices.

① Switch



② Router



③ Gateway

⇒ Software
⇒ Hardware

Translator

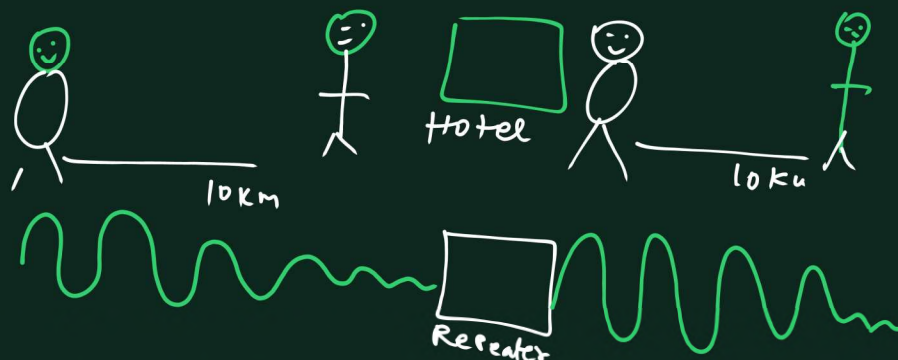
④ Modem



TECHNICAL CLASSES

* Repeater *

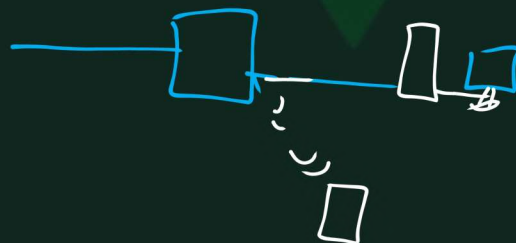
It is a networking device which is used to extend the range of a network by amplifying or regenerating the signal. so that data can be travel to a longer distance.



Wireless Access Point (WAP) :- It is a network device that allows wireless devices such as mobile, Laptop, tabs etc to connect with a wired network using WI-FI. It acts as a bridge between wireless and wired LAN.

NIC (Network Interface Card)

It is a hardware component that enables a computer to connect with a network. It provides physical interface for both wired & wireless network.



Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24

Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



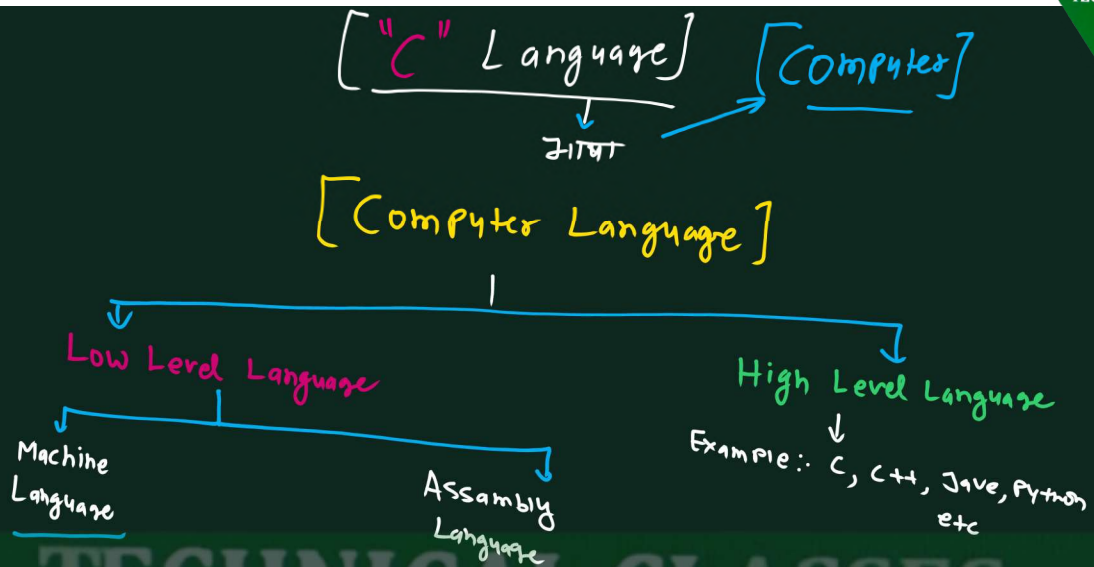
Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24



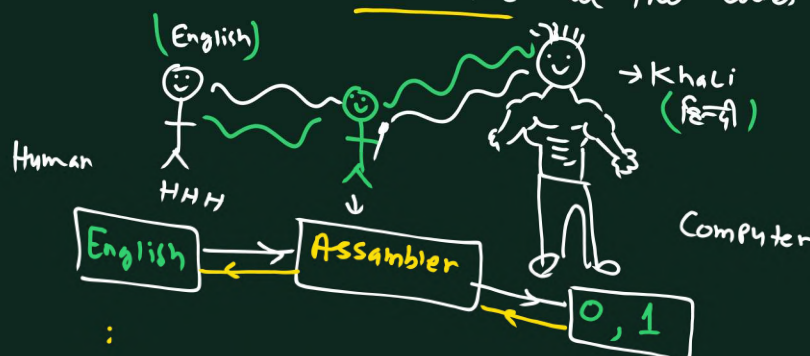
① Machine Language : It is directly understandable by a computer system because it uses 0 (off) 1 (ON) for programming but it is hard for a human to understand.

Example:- $A \rightarrow 65 \rightarrow 1000001$

0101010010

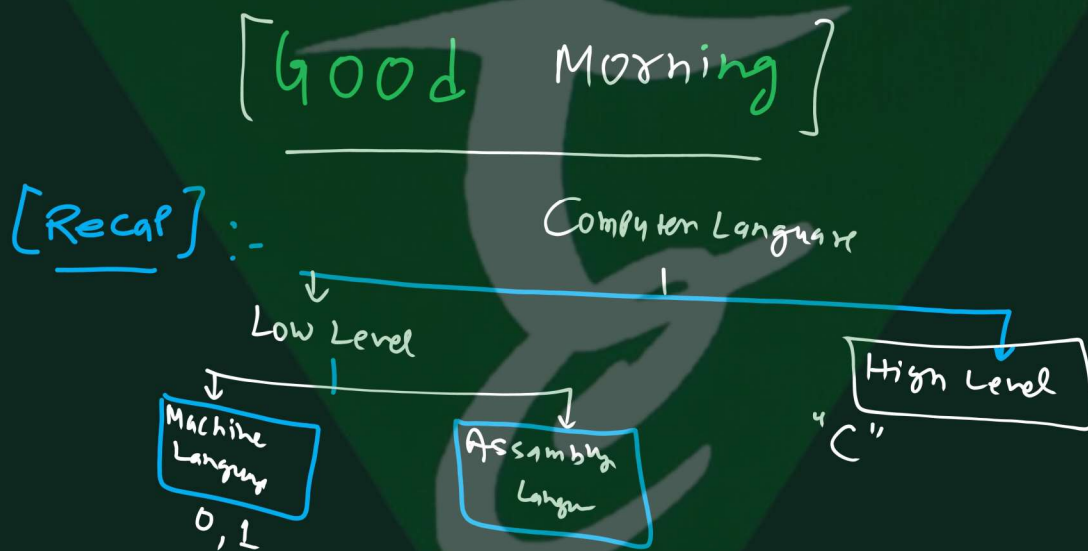
② Assembly Language :- It is english like codes which having a special meaning for computer. A

Programmer have to memorize all the codes for work.



③ High Level Language :- It is english like statement used for programming. It is easy to write and understand because the english code is attached with daily life. we don't have to remember it.

Example :- C, C++, Java, Python etc



Steps :-> Learn Programming Language (C)

↓

Write Programs (set of instruction)

↓

Developing Software (set of Programs)

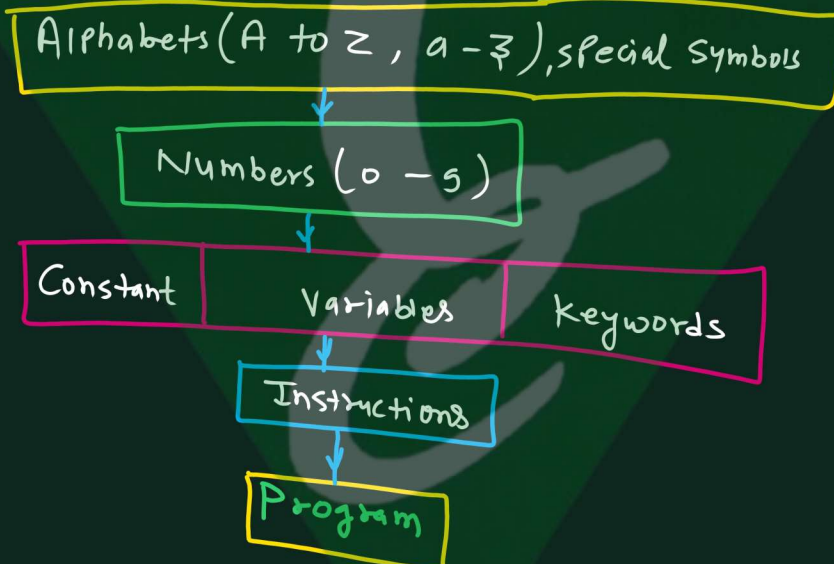
for Computer



[History of C]

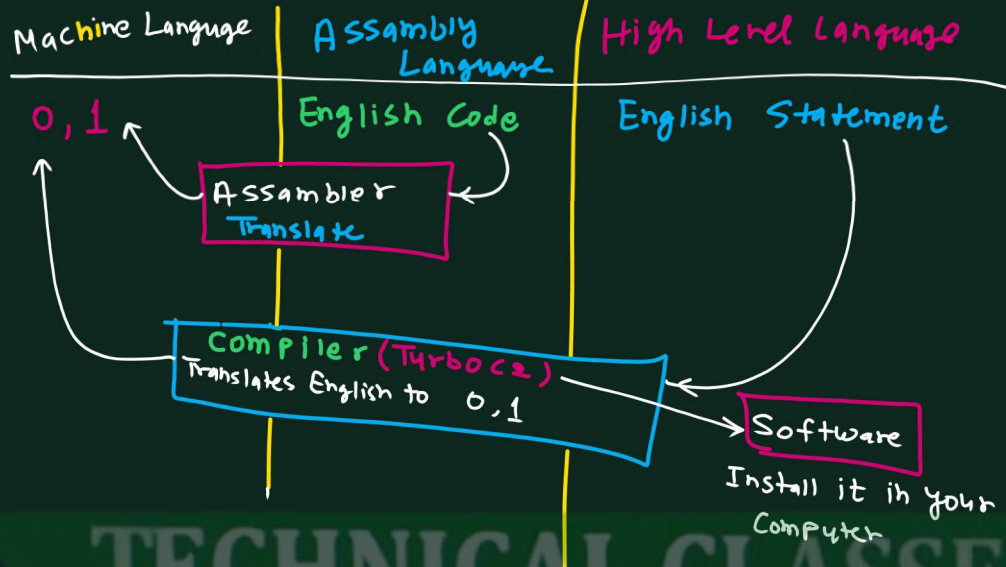
C is a "High Level" Programming Language developed by Denish Ritchie in 1972 at Bell Lab USA. The Programming Language used before C was B and BCPL was used before B.

[Steps to Learn C Language]



Keywords :- Keywords are reserved words (group of Characters) meaning of which is already given in

C Compiler. There are 32 types of keywords in C Language. Example: Printf, scanf, int, float etc.



[Hir]

12.00



Avihasn



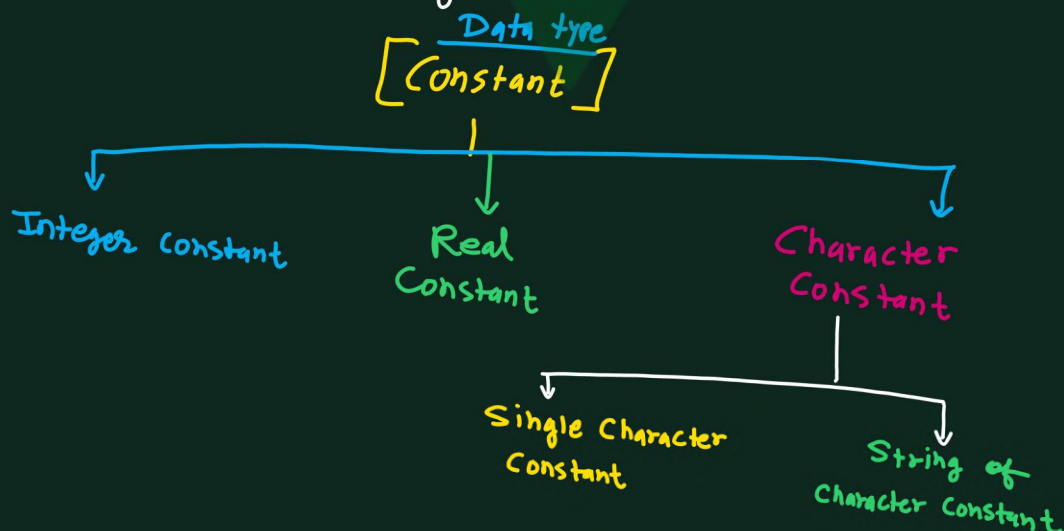
Recap :- Keywords :- $\frac{32}{\text{bit}}$

TECHNICAL CLASSES

[Data type in c / Constant in c]

Data type means the type of input given by a user in a program. So that we should know how much space it will take in memory.

There are following types of data type in c





Constant :- Constant is a value which doesn't changed during execution of a Program.

There are 3 types of Constant

① Integer Constant

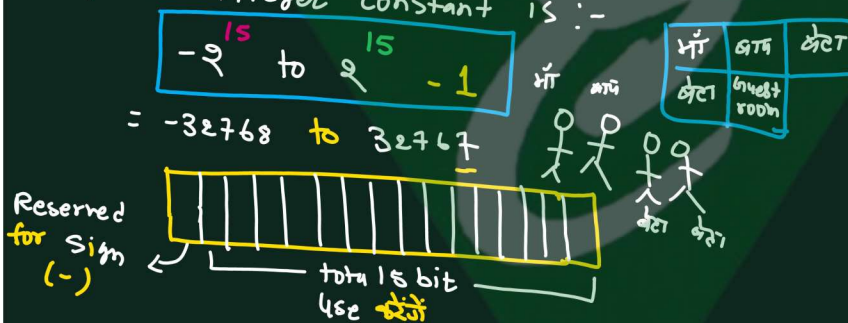
② Real Constant

③ Character Constant

① Integer Constant :- An integer Constant is a number (0-9) which doesn't have any decimal point.

It takes 2 bytes (16 bits) inside RAM. The range

of an integer constant is :-



Q.7 Which of the following is an integer constant ?

① 32869 ✗

② 50.00 ✗

③ 846.93 ✗

④ None ✓

-32768 to 32767

1286 ✓

1286.5 ✗

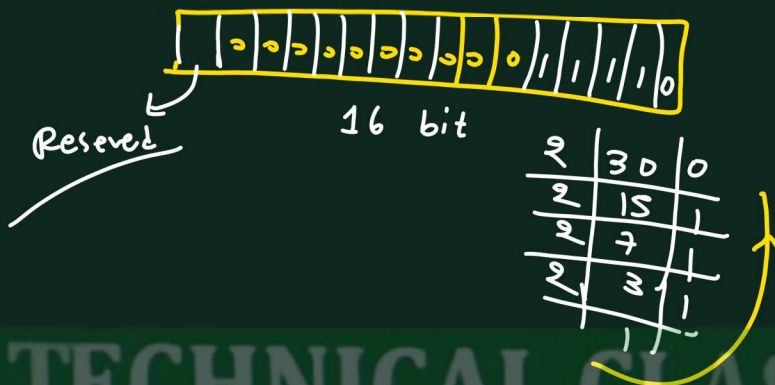
-2868 ✓

32768 ✗

-32769 ✗

Example :- you input 30

↓ Convert into Binary



0 1 2 3 4 5

-5 -4 -3 -2 -1 0

-5 -4 -3 -2 -1 0 1 2 3 4 5

X

[Good Morning]

Recap :- Datay type or Constant

① integer constant :-

Q:-> An integer constant takes 16 bits

Q:-> range -2^{15} to $2^{15} - 1$ 2 byte

$= -32768$ to 32767

② Real Constant : Real Constants are numbers (0-9)

which must have a decimal point. It takes

4 bytes = 32 bits inside RAM. The range of

$$\text{Real Constant} = -2^{31} \text{ to } 2^{31} - 1$$

$$-2147483648 \text{ to } 2147483647$$

Example:-

55.50 ✓

50.00 ✓

120 ✗

214567.00 ✓

2147483649.50 ✗

③ Character constant : Any number, symbol, Alphabets are known as characters

① Single Character constant : Any single character

enclosed within single inverted comma ' ' is known

as single character constant. It takes 1 byte =

8 bits inside RAM. Its range is

$$-2^7 \text{ to } 2^7 - 1 = -128 \text{ to } 127$$

Example :-

'2'	✓
5	X
'21'	X
'a'	✓
"a"	X
#	X
'#'	✓

② String of Character constant :- When more than one characters are enclosed within double quote

" " are known as String of Character constant.

Example:-

"Patha" ✓	"12" ✓	"2" X
Patha X	"@#" ✓	
'Patha' X	12 X	

[Good Morning]

Recap :- Constant :-

integer :- 2 byte = 16 bits = -32768 to 32767

Real :- 4 byte = 32 bits = -2147483647 to 2147483646

Character :- Singl = 'a'

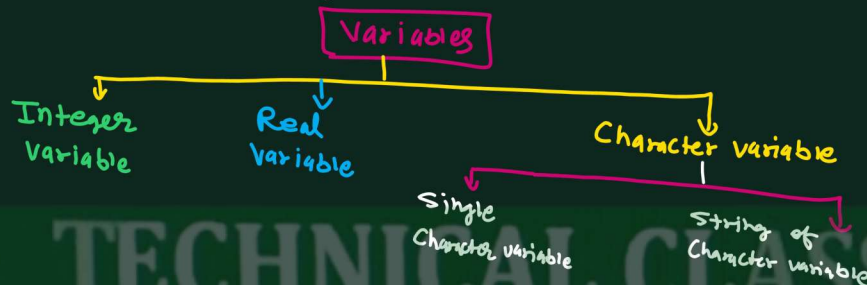
String = "Apple"



[Variables]

Variables are **memory** Location reserved on **RAM**. It is used to store Constants.

There are **3** types of Variable also.



① Integer variable :- Integer variables are used to store integer constant only. It takes **2** byte inside RAM. We use int keyword to declare an integer variable. The format Specifier of an integer variable is %d.

Real variable : Real Variables are used to store real **constant** only. It takes **4 bytes** inside RAM. We use float keyword to declare it. The format Specifier of real variable is %f.



Character variables

① Single Character Variable : It is

Used to store Single Character Constant only.

We use Char keyword to declare it. Its
format specifier is %c. It takes 1 byte.

TECHNICAL CLASSES

String of Character Variable : It is used

to store String of Character Constant only

We use Char keyword to declare it. Its
format specifier is %s.

[Good morning]

f I T C P

Recap :-

Data type or Constant	Variables	
integer	int	%d
Real	float	%f
Character	Char	%c
	Char	%s



Variable name declaration or Identifier

It is the name given to the memory

Location reserved on RAM.

Example :- `int a;`

`a` → variable name
`10` → Value
 Variable name in Memory `1024` → Memory address

TECHNICAL CLASSES

Rules of variable name declaration

- * It may be combination of characters, digits and special symbols.
- * but it must start with character only.
- * Only one special symbol known as underscore (`_`) can be used
- * No any blank space in a single name.

Example :-

`int a;` ✓

`int a5;` ✓

`int 5a;` ✗

`int b@;` ✗

`int b_a;` ✓

`int a b;` ✗

`int a, b;` ✓

`int ab;`

`a` `b`

`ab`

[Basic Input and output Statement]

There are following functions to Perform I/O

① Printf (); This function is used to Print any Message or result on output screen.

Syntax:- `Printf("Your message");`

Example:-

```
#include <stdio.h>
void main()
{
    Printf("welcome");
}
```

Note :- definition of Printf function is written in `#include <stdio.h>`

[Structure of a C Program]

① Header file

```
#include <stdio.h>
```

② main function (`void main()`
`int main()`)

```
void main()
```

③ { Open bracket

```
{
```

④ Body of a Program

```
Printf("welcome");
```

⑤ } Closing bracket

```
}
```



① Header file :- It is the place where Library files are written. It is the top most part of our program. Example

#include <stdio.h>

↓ ↓

Pre-processor Library file
directive

The definition of printf & scanf function is written in <stdio.h> Standard Input Output header file. So that we have to use it for working of printf and scanf

② void main or int main

every program must have a single main function.

③ {

④ }



TECHNICAL CLASSES

[F I T C P]

Recap :-

Printf :-

Syntax:- Printf ("_____");

eg:- int a = 5;

Printf ("%d", a);

Scanf function :- This function is used to take input from a keyboard. It means when the user will give any input this function will catch it and stores in a variable which is declared on RAM.

Syntax:-

scanf("Format specifier", &Variable name);

Example :-

scanf("%d", &a);

↓
address of
Variable

TECHNICAL CLASSES

Q. > W.A.P in C to add any given two numbers.

#include <stdio.h>

void main()

{

int a, b, c;

clrscr();

printf("Enter first no to add = ");

scanf("%d", &a);

printf("Enter second no to add = ");

scanf("%d", &b);

c = a + b;

printf("add = %d", c);

}

150	60	210
a	b	c

Enter first no to add = 150
Enter second no to add = 60
add = 210

After writing a program

① Compile → Alt + F9

② Run → Ctrl + F9

③ View result → Alt + F5



TECHNICAL CLASSES



[Good Morning]

Recap :-

Printf → Output

scanf → Input

clrscr(); :- This Function is used to clear previous result from our output screen so that new program will run on clean screen.

This function is declared in `#include <conio.h>`

TECHNICAL CLASSES

getch(); :- This function is used to take

one character from user otherwise it holds

output screen. So that we use it in our program

and it saves our time to Press Alt + F5

It is defined in `#include <conio.h>`

Note:- if we use Ctrl + F5 button it will

do both work at a time

① Compile

② run

Q. Which key is used to run as well as compile a program

⇒ Ctrl + F5

How to write a programCooking

- ① वर्तन , Basic
- ② सामान
- ③ पकाना
- ④ परोचना

Programming

- ① **Basic**, variable declare
- ② Input
- ③ Process
- ④ output

TECHNICAL CLASSES

Q.7 W.A.P in C to find S.I of any given P, r, t.

#include <stdio.h>

#include <conio.h>

void main()

{

 float P, r, t, si;
 clrscr();

Enter Principal = 5000

Enter rate = 5

Enter time = 4

The S.I = 1000

printf("Enter Principal=");

scanf("%f", &P);

printf("Enter rate=");

scanf("%f", &r);

printf("Enter time=");

scanf("%f", &t);

5000

P

2

r

3

t

30

si

Enter Principal = 500.00

Enter rate = 5

Enter time = 3

The S.I = 30



Q. > W.A.P in C to Convert temp from f to c

$$Si = (P * r * t) / 100;$$

```
printf("The S.I = %f", si);
```

```
getch();
```

```
}
```

Q. > W.A.P in C to convert F to °C

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
float f, c;
```

$\boxed{-40}$ $\boxed{-40}$
f c

```
clrscr();
```

```
printf("Enter temp in F");
```

```
scanf("%f", &f);
```

```
c = 5 * (f - 32) / 9;
```

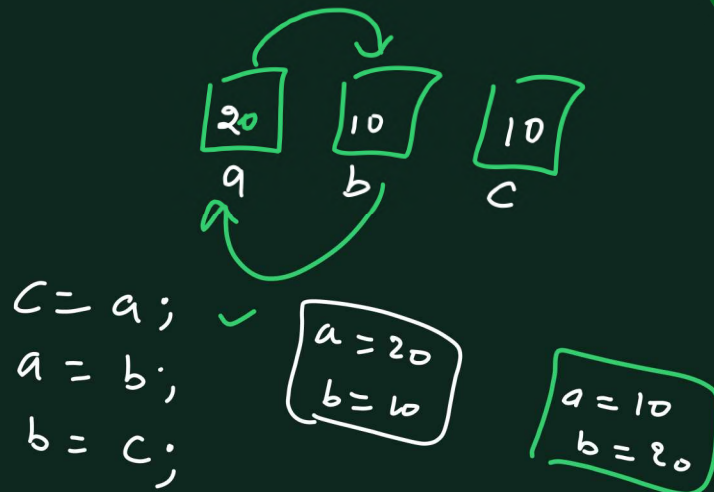
```
printf("After conversion temp = %f", c);
```

```
getch();
```

```
}
```




Swapping



TECHNICAL CLASSES

[Conditional Statements]

It is used in a program when we have to take a decision according to given Condition.

There are following types of Conditional statements

- (i) if
- (ii) if else
- (iii) Nesting of if else
- (iv) else if Ladder



if Statement :- It checks the condition

and if condition is true it works.

Syntax:-

```

if (condition)
{
    Statement ;
}

```

True
it will work

TECHNICAL CLASSES

Q.7 W.A.P in C to find given no is positive

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
{
```

```
    int n;
```

```
    clrscr();
```

```
    printf("Enter any no to check Positive=");
```

```
    scanf("%d", &n);
```

5
n

Enter any no to check Positive 5

```
    if (n > 0)
```

```
    {
```

```
        printf("given no = %d is positive", n);
```

```
    }
```

```
    getch();
```

```
}
```



ii) if else statement :- It is used in a program when we have to check a condition and given result in both (True or false) condition.

Syntax:-

```

if (condition)
{
    Statement ;
}
else
{
    Statement
}
  
```

Diagram illustrating the execution flow of an if-else statement:

- The condition is evaluated. If it is **True**, the flow goes to the **do this** block under the **if** condition.
- If the condition is **false**, the flow goes to the **do this** block under the **else** condition.

[Good morning]

[Recall] Conditional statement

① if

② if else

Q. > W.A.P in C to find biggest between any given two numbers.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
    int a, b;
```

```
    clrscr();
```

```
    printf("Enter any number = ");
```

```
    scanf("%d", &a);
```

```
    printf("Enter any second number");
```

```
    scanf("%d", &b);
```

```
    if (a > b) 50 > 20
    {
        printf("biggest no = %d", a);
    }
```

```
    else
    {
        printf("biggest no = %d", b);
    }
```

```
    getch();
}
```

50
a

20
b

[Nesting of if else]

When we put an if else statement

within another if else statement it is

known as Nesting of if else.

Syntax:-

```
if (condition)
{
    if (condition)
    {
        Statement;
    }
    else
    {
        Statement;
    }
}
else
{
    if (condition)
    {
        Statement;
    }
    else
    {
        Statement;
    }
}
```


V.V.I:- W.A.P in to find biggest between any given 3 no.s

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b, c;
    clrscr();
    printf("Enter any 3 no to check biggest");
    scanf("%d %d %d", &a, &b, &c);
```

100 50 40
a b c

```
if (a > b)
{
    if (a > c)
    {
        printf("A is biggest");
    }
    else
    {
        printf("C is biggest");
    }
}
else
{
    if (b > c)
    {
        printf("b is biggest");
    }
    else
    {
        printf("C is biggest");
    }
}
getch();
```

iv. Else if Ladder:- It is a multi conditional statement which is used in a program when we are having more than 3 conditions.

Syntax:-

```

if (Condition)
{
    Statement;
}
else if (Condition)
{
    Statement;
}
else if (Condition)
{
    Statement;
}
...
else
{
    Statement;
}

```

Q. > W.P.C to Print Month name according to given month value.

```

#include <stdio.h>
#include <conio.h>
void main ( )
{
    int h;
    clrscr();
    printf("Enter any month value = ");
    scanf("%d", &h);

    if (h == 1)
    {
        printf("Jan");
    }
    else if (h == 2)
    {
        printf("Feb");
    }
    else if (h == 3)
    {
        printf("March");
    }
    ...
    else if (h == 12)
    {
        printf("Dec");
    }
    else
    {
        printf("Wrong input");
    }
    getch();
}

```

15
h



Operator :- It is a Symbol used to Perform any arithmetic or Logical Operations in a Program. Example: $+$, $-$, $*$, etc.

TECHNICAL CLASSES

There are following types of Operators in C.

- (i) Arithmetical Operator
- (ii) Relational "
- (iii) Logical "
- (iv) Ternary " , or Conditional Operator
- (v) Increment / Decrement Operator
- (vi) Arithmetical Assignment "
- (vii) Bitwise "

(i) Arithmetical Operator :- It performs arithmetic Calculation in a Program. These are

(i) $+$:- Add

(ii) $-$:- Sub

(iii) $*$ $\rightarrow$ Multiplication

(iv) $/$ $\rightarrow$ Divide

(v) $\%$ $\rightarrow$ Modulus or Remainder Operator



$\%$:- This operator is used to find remainder of given expression.

eg:-

$$10 \% 3 = 1$$

$$10 \% 2 = 0$$

3) 10 (3
9
1 शेषफल

TECHNICAL CLASSES

Note 1 :- If numerator is Less than denominator

then result will be always numerator.

example

$$2 \% 3 = 2$$

$$10 \% 15 = 10$$

Note 2 :- Modulus operator doesn't work with float value.

Q. > $6.6 \% 2 = \text{error}$

Tyroboc2

C++

(i) Relational Operator :-

This Operator is used to Compare the given expression and after comparing it returns **True** (1) or **False** (0) according to given value.

TECHNICAL CLASSES

There are following types of relational operators

- (i) $>$ $\rightarrow$ greater than
- (ii) $<$ $\rightarrow$ Less "
- (iii) $>=$ $\rightarrow$ greater than or equal to
- (iv) $<=$ $\rightarrow$ Less " " " "
- (v) $==$ $\rightarrow$ Is equal to
- (vi) $!=$ $\rightarrow$ Not equal to

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int a = 5, b = 2, c;
    clrscr();
    c = (a > b);
    printf(" %d", c);
    getch();
}
```



#

#

void main()

{

```
int a = 5, b = 2, c;
clrscr();
```



c = (a == b);

```
printf("%d", c);
getch();
```

}

TECHNICAL CLASSES

[Good Evening][Recap] :- [😊] Operators

① Arithmetic operator

② Relational

Logical operator :- Logical operatorsare used to combine two or morethan two relational operators and return either True(1) or False(0) according to given expression.



There are 3 types of Logical operator

① Logical AND (&&)

② Logical OR (||)

③ Logical Not (!)

TECHNICAL CLASSES

① Logical AND (&&) : It returns true only if all value of expression is True. Otherwise it returns false.

exp 1	exp 2	result
T	T	T ✓
T	F	F
F	T	F
F	F	F

Er. Gaurav
Mr. Er. Gaurav
Lipistick ✓ AND ✓ cyciter

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
int a = 5, b = 4, c;
```

```
clrscr();
```

```
c = (a > b) && (b == 4);
```

```
printf("%i", c);
```

```
getch();
```

```
}
```

```
c = 1
```

(ii) Logical OR (||) :- It returns false only if all condition is false otherwise it returns True.

exp1	exp2	Result
T	T	T
T	F	T
F	T	T
F	F	F

eg:-

```

void main()
{
    int a=5, b=4, c;
    clrscr();
    c = (a < b) || (b == 4);
    printf("%d", c);
    getch();
}

```

$(5 < 4)$ is False (F) and $(4 == 4)$ is True (T).
 Since one condition is True, the result of the OR operation is True (T).

Logical Not (!) :- It is used to reverse the result of any relational expression.

exp1	Result
T	F
F	T



```
#
#
void main()
{
    int a = 5, b;
    clrscr();
    b = !(a > 4)
    printf("%d", b);
    getch();
}
```

| !(5 > 4)
 T

0

(iv) Conditional or Ternary Operator :-

Conditional or Ternary operator is used in a program when there is a condition and we have to take decision according to given condition and to perform a task.

Syntax:-

False
True

(Condition) ? Value 1 : Value 2 ;

The compiler first checks the condition & if given condition is True the compiler will go in 1st part (Value 1) otherwise it will transfer to second part (Value 2)

Q. > W.P.P in C to find biggest betw given two numbers.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b;
    clrscr();
    printf("Enter any two no to find biggest = ");
    scanf("%d %d", &a, &b);
    (a > b) ? printf("A is biggest") : printf("B is biggest");
    getch();
}
```

100 20
a b

Enter any two no to find biggest =
100 20
A is biggest

Increment operator :- $(++)$:- Increment

operator is used to increase the value of an operand by one. eg:-

```
#include <stdio.h>
void main()
{
    int a = 5;
    a++;
    printf("%d", a);
}
```

6
a

6
a

There are two type of increment operator

(i) Preincrement operator $(++a)$

(ii) Postincrement operator $(a++)$

Note 1 :- Pre increment and Post increment operator works same when they are independently used in a program.



Pre increment

```
#
#
void main( )
{
    int a = 5;
    clrscr();
    ++a;
    printf("%d", a);
    getch();
}
```

Diagram showing memory address 9 containing value 6. A box around 6 has a checkmark below it.

Post increment

```
#
#
void main( )
{
    int a = 5;
    clrscr();
    a++;
    printf("%d", a);
    getch();
}
```

Diagram showing memory address 9 containing value 6. A box around 6 has a checkmark below it.

Note-2 :- Pre increment and Post increment works

differently if they are used with another expression.

Pre increment

```
#
#
void main( )
{
    int a = 5, b;
    clrscr();
    b = ++a;
    printf("%d", b);
    getch();
}
```

Sit -1

Diagram showing memory address 9 containing value 6. An arrow points from this box to another box at address 6 containing value 6, which also has a checkmark below it.

Post increment

```
#
#
void main( )
{
    int a = 5, b;
    clrscr();
    b = a++;
    printf("%d", b);
    getch();
}
```

Sit 2 →

Diagram showing memory address 9 containing value 6 and memory address b containing value 5. A box around 5 has a checkmark below it.

Operator (part 7)

Bitwise operator :- Bitwise operators are used to perform bit level programming in various places such as Microchip programming. These are



1. Bitwise AND (&)

2. Bitwise OR (|)

3. Bitwise XOR (^)

4. Bitwise Not (~)

5. Bitwise Left Shift (<<)

6. Bitwise Right Shift (>>)

1. Bitwise AND (&)

It firstly converty Decimal value to binary and then returns 1 only if all bits are 1 otherwise it returns 0.

eg:-

```
#include <stdio.h>
void main()
{
    int a = 5, b = 4, c;
    clrscr();
    c = a & b;
    printf("%d", c);
    getch();
}
```

Binary Conversion

a = 5	→	0 1 0 1
b = 4	→	0 1 0 0
a & b	→	0 1 0 0

Decimal Conversion

4



2.7 Bitwise OR (|)

→ It firstly converts the decimal value into binary then returns 0 only if all values are 0 otherwise it returns 1.

eg:-

```

##
##
void main()
{
    int a=5, b=4, c;
    clrscr();
    c = a | b;
    printf("%d", c);
    getch();
}

```

a = 5 →	0 1 0 1
b = 4 →	0 1 0 0
a b →	0 1 0 1
	5
	c

←

3.7 Bitwise XOR (^)

→ In this operation result will be 1 only if corresponding bits of the operands have different value.

eg:-

a →	0 0 1 1
b →	0 1 1 0
	0 1 0 1



TECHNICAL CLASSES

भा 23

Simple Program → Conditional Program →

if, if else, nesting
[else if Ladder]

Termin LC

Switch statement :- It is a multidirectional

branching statement which works same as

else if Ladder but it is easier than that.



Syntax:-

```

Switch(value)
{
    case value1:
        statement;
        break;
    case value2:
        statement;
        break;
    ...
    default:
        statement;
}

```

Annotations:
 - An arrow points from 'Switch(value)' to 'value' with the label '→ value'.
 - An arrow points from 'case value1:' to 'value1' with the label '→ value1'.
 - An arrow points from 'case value2:' to 'value2' with the label '→ value2'.
 - An arrow points from 'default:' to 'default' with the label '→ default'.

Q. WAP in C to print day of week by taking number from 1 to 7.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
    int n;
    clrscr();

```

```
    printf("Enter value to print week day = ");

```

```
    scanf("%d", &n);

```

Enter value to print = 2

2
n

```
Switch(n) {
```

```
    case 1:
        printf("Sunday");
        break;

```

```
    case 2:
        printf("Mon");
        break;

```

```
    case 3:
        printf("Tue");
        break;

```

```
    case 4:
        printf("Wed");
        break;

```

```
    case 5:
        printf("Thur");
        break;

```

```
    case 6:
        printf("Fri");
        break;

```

```
    case 7:
        printf("Sat");
        break;

```

```
    default:
        printf("sorry wrong input");
    }
    getch();
}
```

* Loop *

Looping statement is used in a program when we have to perform similar type of task again & again.

TECHNICAL CLASSES

There are 3 type of loop in C

- ① for loop
- ② while loop
- ③ do while loop

① for loop :- It is used to perform repeating type of task in a program. It works by checking the condition and if condition is True then it work. So that it is also know as Pre-Test Loop.

Syntax:-

```

for (intilization ; Condition ; increment/decrement)
{
    Statement ;
}

```

Diagram illustrating the flow of a for loop: The loop starts with 'intilization', then checks 'Condition'. If 'True', it executes 'Statement' and then 'increment/decrement', looping back to 'Condition'.

Example :-

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    clrscr();
    for (i = 1 ; i <= 5 ; i++)
    {
        printf("\n welcome");
    }
    getch();
}

```

6
i

welcome
welcome
welcome
welcome
welcome

Q.7 W.A.P in C to print table of any given no.

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int n, i, t;
    clrscr();
}

```

```

printf("Enter table number =");
scanf("%d", &n);
for(i=1; i<=10; i++)
{
    t = n * i;
    printf("%d", t);
    getch();
}

```

Q.→ W.A.P in C to Print factorial of any given no.

[Good Morning]

Recap :-

[Loop]

1.→ for ✓

2.→ while

3.→ do while

[While Loop]

It is Same as for Loop used in a Program to Perform **repetative** type of **task**. It first Checks the Condition and if given Condition is TRUE it work. So that it is also known as

PreTest Loop.

Syntax:- :-

```

initialization ;
while (condition)
{
    Statement ;
    increment / decrement
}

```

Example

```

#include <stdio.h>
#include <conio.h>

void main()
{
    int i = 1;
    clrscr();
    while (i <= 5)
    {
        printf("welcome");
        i++;
    }
    getch();
}

```

In for Loop

```

#include <stdio.h>
#include <conio.h>

void main()
{
    int i;
    clrscr();
    for (i = 1; i <= 5; i++)
    {
        printf("welcome");
    }
    getch();
}

```



Q.7 W.A.P in C to print Fibonacci series upto 13.

0 1 1 2 3 5 8 13

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
    int f, s, t;
```

```
    clrscr();
```

```
    f = 0;
```

```
    s = 1;
```

TECHNICAL CLASSES

```
    t = f + s;
```

```
    printf("%d %d", f, s);
```

```
    while(t <= 13)
```

```
    {
        printf("%d", t);
```

```
        f = s;
```

```
        s = t;
```

```
        t = f + s;
```

```
    }
```

```
    getch();
```

```
}
```

8
f

13
s

21
t

0 1 1 2 3 5 8 13

Q. W.A.P in C to reverse a given number.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n, s, rn = 0;
    clrscr();
    printf("Enter any number to reverse=");
    scanf("%d", &n);
```

```
while (n > 0)
{
    0 > 0
```

```
    s = n % 10; 1 % 10
```

```
    rn = (rn * 10) + s;
           321
```

```
    n = n / 10;
```

```
}
printf("Reverse of given no = %d", rn);
getch();
}
```

123
10 321
n rn

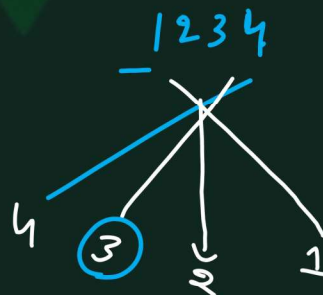
1
s

10) 1 (0

321

4 = 1234 % 10

123 = 1234 / 10





["C" Language]

Recap:- [while Loop]

T.T.1:- Fibonacci series

Reverse number

TECHNICAL CLASSES

Q> W.A.P in C to check given number is Palindrome.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
    int n, rn=0, c, s;
```

```
clrscr();
```

```
printf("Enter Any number to check Palindrome");
```

```
scanf("%d", &n);
```

```
c = n;
```

```
while(c > 0)
```

```
{
```

```
s = c % 10;
```

```
rn = (rn * 10) + s;
```

```
c = c / 10;
```

```
}
```

```
if(rn == n)
```

```
{
    121 == 121
```

```
printf("given no = %d is Palindrome", n);
```

```
}
```

```
else
```

```
{
```

```
printf("given no = %d is Not Palindrome", n);
```

```
}
```

```
getch();
```

```
}
```

Q.7 W.A.P in C to check given no is Armstrong or not. ex:-

$$\begin{aligned}
 & \boxed{153} \\
 & 1^3 + 5^3 + 3^3 \\
 & = 1 + 125 + 27 \\
 & = \boxed{153} \quad 3^3 + 5^3 + 1^3
 \end{aligned}$$

TECHNICAL CLASSES

```

#include<stdio.h>
#include<conio.h>
void main( )
{
    int n, Ar=0, s, c;
    clrscr( );
    printf("Enter Any number to Check Armstrong=");
    scanf("%d", &n);
    c = n;

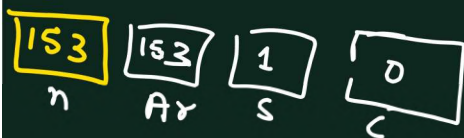
```

```

while(c > 0)
{
    s = c % 10;
    Ar = Ar + s * s * s;
    c = c / 10;
}

if (Ar == n)
{
    printf("given no = %d is Armstrong", n);
}
else
{
    printf("given no = %d is Not Armstrong", n);
}
getch();
}

```



$$\begin{array}{r}
 10 \overline{) 153} \\
 \underline{10} \\
 53 \\
 \underline{50} \\
 3
 \end{array}$$



TECHNICAL CLASSES

["C" Language]

[do while Loop]

Recap:-

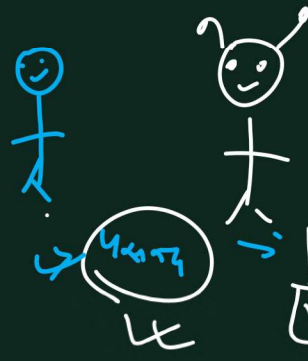
[while Loop]

[for Loop]

①

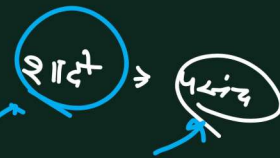
②

[Pretext Loop]



do while

4444 आ (9999)





[Do-while Loop]

It is also a Looping Statement used in a Program to Perform repetitive type of tasks. It firstly Performs the task and then Checks the Condition. So that it is also known as Post test Loop.

Syntax:-

```
do
{
    Statements;
    ;
    increment / decrement;
}
while(Condition);
```

Note:- do while loop runs at least one time even if given Condition is false.

example :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i=1;
    clrscr();
    do
    {
        printf("welcome");
        i++;
    }
    while(i > 5);
    getch();
}
```

2
i

welcome
one time

```
int i=1;
clrscr();
while(i > 5) X no time
{
    printf("welcome");
    i++;
}
```



g> V.V.I W.A.P in c to make a calculator.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b, c;
    char ch;
    clrscr();
    printf("Enter your choice=");
    scanf("%c", &ch);
    printf("Enter any two no to perform=");
    scanf("%d %d", &a, &b);
```

```
switch(ch)
{
    case '+':
        c = a + b;
        printf("Add = %d", c);
```

```
        break;
```

```
    case '-':
```

```
        c = a - b;
```

```
        printf("Sub = %d", c);
        break;
```

```
    case '*':
```

```
        c = a * b;
```

```
        printf("Mul = %d", c);
```

```
        break;
```

```
    case '/':
```

```
        c = a / b;
```

```
        printf("Div = %d", c);
```

```
        break;
```

```
    default:
```

```
        printf("Sorry wrong choice");
}
```

```
    getch();
```

```
}
```

1. A
ch

50
a

20
b

30
c



Q.7 W.A.P in C to check given character is vowel or consonant.

TECHNICAL CLASSES

[Good morning]

Flowchart, Algorithm, Pseudocode

①

②

③

[Program]



Flow Chart :- It is a graphical representation of any Problem, Algorithm.

It is a visual representation of a Process showing the sequence of steps and decision points. It uses Symbols, Shapes to show the steps.

TECHNICAL CLASSES

There are following types of symbol used in flowchart.

i)  :- Start/stop

ii)  :- Input/output

iii)  :- Process

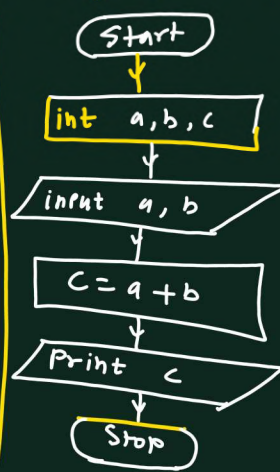
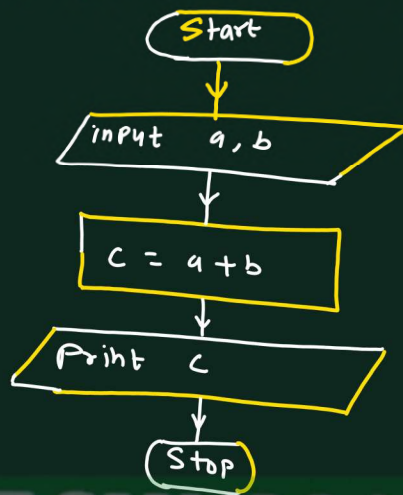
iv)  :- Decision

v)  :- Flow

vi)  :- Looping

vii)  :- Connector

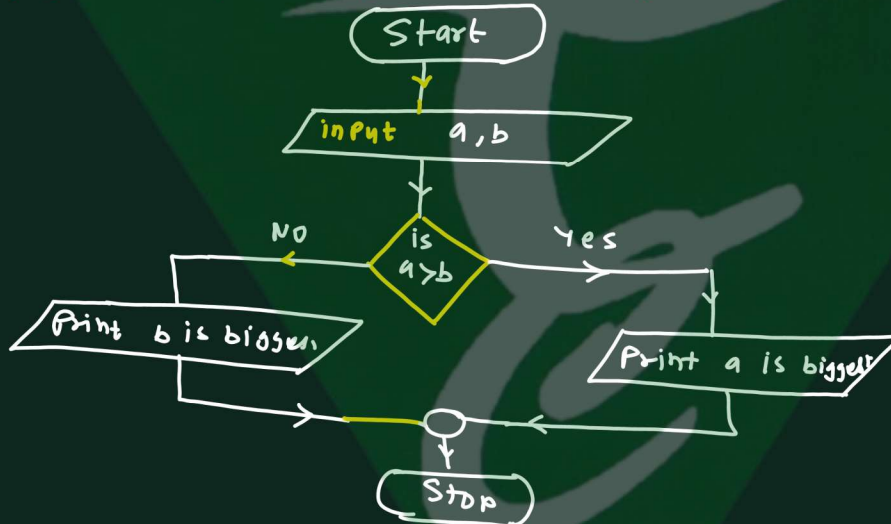
Q7 Draw a flowchart to add any given two nos.



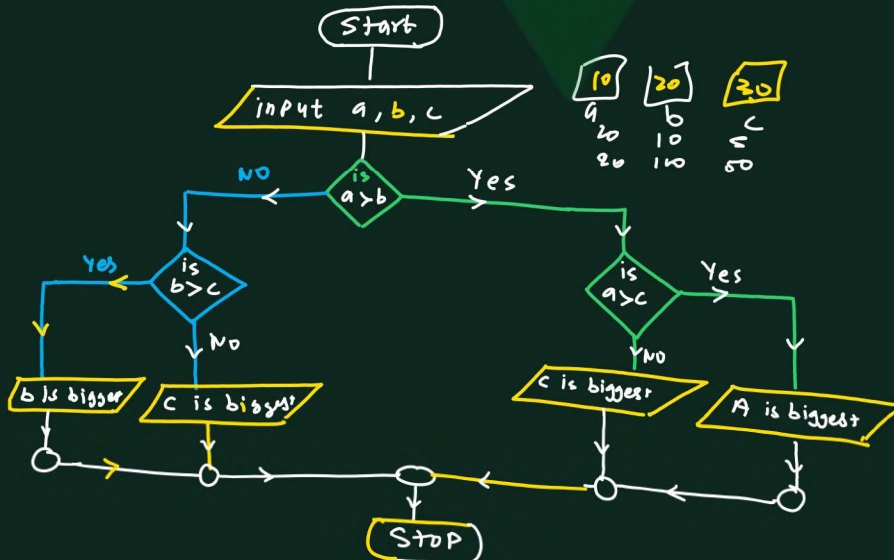
```

//
// void main()
//
-> int a, b, c;
-> printf(
    scanf(
-> c = a + b;
-> printf
-> return(
    )
    )
    
```

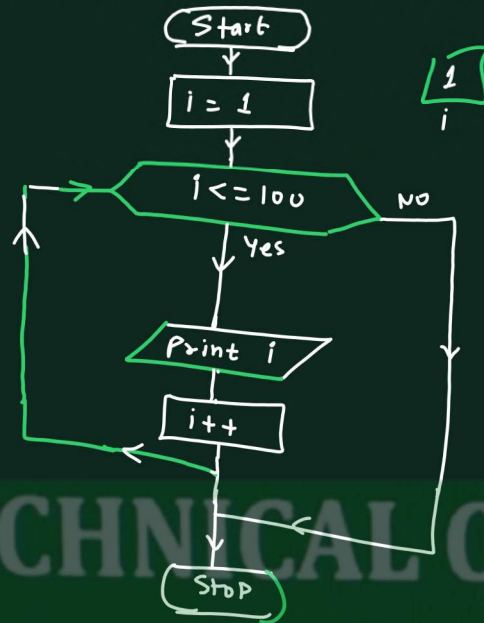
Q7 Draw a flowchart to find biggest between any two no.



Q7 Draw a flowchart to find biggest between 3 nos.



Q. > Draw a flow chart to Print 1 to 100



[Algorithm]

It is step by step solution of any problem

Written in english statement.

Q. > Write an algorithm to add any given two numbers.

Step 1 :- Start

Step 2 :- Input in a, b

Step 3 :- $C = a + b$

Step 4 :- Print C

Step 5 :- Stop



Q.7 write an algorithm for greatest betwn 3 no.

Step 1:- Start

Step 2:- input three numbers a, b, c

Step 3:- if $A > B$ and $A > C$ then print
A is greatest

Step 4:- Otherwise if $B > A$ and $B > C$ then print
B is greatest

Step 5:- Otherwise print C is greatest

Step 6:- Stop

* Pseudo Code *

It is Combination of Algorithm and Programming

Codes.

Example:-
Start
input a, b
 $C = a + b$
Print c
Stop

Q.7 write Pseudo Code for Greatest between three numbers.

Start

Input a, b, c

if $a > b$ and

if $a > c$

Print "A is biggest"

else

Print "c is biggest"

else if $b > c$

Print "b is biggest"

else
Print "C is biggest"

End

Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24

Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24

[Array]

An array is a secondary data type used to store similar type of data at a single memory location. It stores data in a sequence. It is also known as homogenous data structure.

[Types of Array]

One dimensional Array
or
Single dimensional Array

Two dimensional Array
or
double dimensional Array

① One dimensional array:- It is used to store similar type of data at a single place with a single name.

Syntax:- `datatype Arrayname[size];`

Example:- `int a[5];` Subscript value

In memory :-



Note:- An array Index is always starts with 0.

Input In an Array:-

for input we use `scanf` function with `for` loop.

Output in an array

for output we use `printf` function with `for` loop.

TECHNICAL CLASSES

or W.A.P in C to declare an array of size 5 & perform I/O.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[5], i;
    clrscr();
    for(i=0; i<5; i++)
    {
        printf("\n Enter elements =");
        scanf("%d", &a[i]);
    }
    printf("\n Elements are =");
    for(i=0; i<5; i++)
    {
        printf("\n %d", a[i]);
    }
    getch();
}
```

In memory:-

15	25	30	8	12
a[0]	a[1]	a[2]	a[3]	a[4]



[Good Morning]

Recall :- Array :- Sequential, Secondary data store

Homogeneous D.S

Index :- 0

I/O :- for loop

eg:-
int a[5];



Q. > W.A.P in C to find biggest element in an array of size 5.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    int a[5], i, b=0;
```

```
    clrscr();
```

```
    for(i=0; i<5; i++)
```

```
    { printf("Enter elements=");
```

```
      scanf("%d", &a[i]);
```

```
    }
```

```
    for(i=0; i<5; i++)
```

```
    { if(a[i] > b)
```

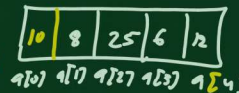
```
      { b=a[i];
```

```
      }
```

```
    } printf("The biggest element = %d", b);
```

```
    getch();
```

```
}
```



```
Enter elem = 10
Enter elem = 8
The biggest = 25
```

Hint

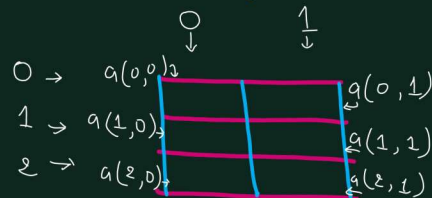
Q. > W.A.P in C to find addition of all elements in an array of size 5.

* Two dimensional array :- It is used to store similar type of data in row & column manner (only show)

Syntax:- data type arrayname [row size] [column size];

ex :- int a [3] [2];

In memory



TECHNICAL CLASSES

I/O:- for I/O on 2 dimensional Array we use Nesting of for loop.

Q:- W.A.P in C to perform I/O in a 2 dimensional array of size 3x2.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
    int a[3][2], i, j;
    clrscr();
```

```
for(i=0; i<3; i++)
```

```
{
    for(j=0; j<2; j++)
```

```
{
    printf("Enter elements=");
    scanf("%d", &a[i][j]);
```

```
}
}
```

$a(0,0)$	10	8	$a(0,1)$
$a(1,0)$	15	5	$a(1,1)$
$a(2,0)$	9	18	$a(2,1)$

```

10 8 10
15 5 10
9 18 10

```



```

printf("In Elements are ");
for(i=0; i<3; i++)
{
    for(j=0; j<2; j++)
    {
        printf(" %d\t", a[i][j]);
    }
    printf("\n");
}
getch();

```

$a(0,0)$	10	8	$a(0,1)$
$a(1,0)$	15	5	$a(1,1)$
$a(2,0)$	9	18	$a(2,1)$

10 8
 15 5
 9 18

Array part-3

Recall :-

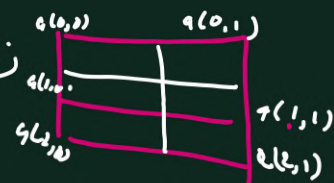
Array → one dimensional Array :-

int a[5];



• Two dimensional Array

int a[3][2];





Q. > W.A.P in c to Add given two matrix of size 2x2

```
#include<stdio.h>
#include<conio.h>

void main()
{
    int a[2][2], b[2][2], c[2][2], i, j;
    clrscr();
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            printf("Enter elements in first Matrix=");
            scanf("%d", &a[i][j]);
        }
    }
```

a(0,0)	10	5	a(0,1)
a(1,0)	8	12	a(1,1)

Enter element in form = 10
Enter element in form = 5
Enter element in form = 8
Enter element in form = 12

```
for(i=0; i<2; i++)
{
    for(j=0; j<2; j++)
    {
        printf("Enter elements into second Matrix=");
        scanf("%d", &b[i][j]);
    }
}
```

b(0,0)	2	3	b(0,1)
b(1,0)	5	6	b(1,1)

```
for(i=0; i<2; i++)
{
    for(j=0; j<2; j++)
    {
        c[i][j] = a[i][j] + b[i][j];
    }
}
printf("\n The addition = \n");
for(i=0; i<2; i++)
{
    for(j=0; j<2; j++)
    {
        printf("%d\t", c[i][j]);
    }
    printf("\n");
}
getch();
}
```

a(0,0)	10	5	a(0,1)
a(1,0)	8	12	a(1,1)

b(0,0)	2	3	b(0,1)
b(1,0)	5	6	b(1,1)

c(0,0)	12	8	c(0,1)
c(1,0)	13	18	c(1,1)

12 8
13 18



Previous H.W :-

Q> Addition of all array elements

① variable $s=0$, $a[5]$, i

② Input

```
for (i = 0; i < 5; i++)
{
    s = s + a[i];
}
```

Print s

TECHNICAL CLASSES

Q> W.A.P in C to print transpose of a matrix 3×2

Recap :-

Two dimensional Array

Addition =

[Array Part - 4]* Multiplication of Two Matrices. 2×2

2	3
1	4

a

4	1
2	3

b

$2 \times 4 + 3 \times 2$ 14	$2 \times 1 + 3 \times 3$ 11
$1 \times 4 + 4 \times 2$ 12	$1 \times 1 + 4 \times 3$ 13

c

TECHNICAL CLASSES

Q. > W.A.P in C to Multiply given two matrix of size 2×2

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int a[2][2], b[2][2], c[2][2], i, j, k;
    clrscr();
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            printf("Enter elements in first Matrix=");
            scanf("%d", &a[i][j]);
        }
    }
```

$a(0,0)$	2	3	$a(0,1)$
$a(1,0)$	1	4	$a(1,1)$

Enter element in form = 2
 2
 3
 1
 4

```
for(i=0; i<2; i++)
{
    for(j=0; j<2; j++)
    {
        printf("Enter elements into second Matrix=");
        scanf("%d", &b[i][j]);
    }
}
```

	$b(0,0)$	4	1	$b(0,1)$
$b(1,0)$	2	3	$b(1,1)$	



```

for(i=0; i<2; i++)
{
    for(j=0; j<2; j++)
    {
        c[i][j] = 0;
        for(k=0; k<2; k++)
        {
            c[i][j] = c[i][j] + a[i][k] * b[k][j];
        }
    }
}

```

a(0,0)	2	3	a(0,1)
a(1,0)	1	4	a(1,1)

b(0,0)	4	1	b(0,1)
b(1,0)	2	3	b(1,1)

c(0,0)	14	11	c(0,1)
c(1,0)	12	13	c(1,1)

```

    getch();
}
printf("\n Multiplication = \n");
for(i=0; i<2; i++)
{
    for(j=0; j<2; j++)
    {
        printf("%d\t", c[i][j]);
    }
    printf("\n");
}
getch();
}

```

Q. > W.A.P in C to transpose a matrix of size 2x3

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int a[2][3], i, j;
    clrscr();
    for(i=0; i<2; i++)
    {
        for(j=0; j<3; j++)
        {
            printf("Enter elements = ");
            scanf("%d", &a[i][j]);
        }
    }
}

```

2	4	6
3	6	9

2	3
4	6
6	9



```
printf("Transpose = \n");
```

```
for(i=0; i<3; i++)
```

```
{
    for(j=0; j<2; j++)
    {
        printf("%d\t", a[j][i]);
    }
    printf("\n");
}
```

```
    {
        for(j=0; j<2; j++)
```

```
    {
```

```
        printf("%d\t", a[j][i]);
```

```
    }
```

```
    printf("\n");
```

```
getch();
```

```
}
```

$a(0,0)$	$a(0,1)$	$a(0,2)$
2	4	6
$a(1,0)$	$a(1,1)$	$a(1,2)$
3	5	9

2	3
4	6
5	9

TECHNICAL CLASSES

Q. > W.A.P in C to print transpose of a matrix 3x2

[Good Morning]

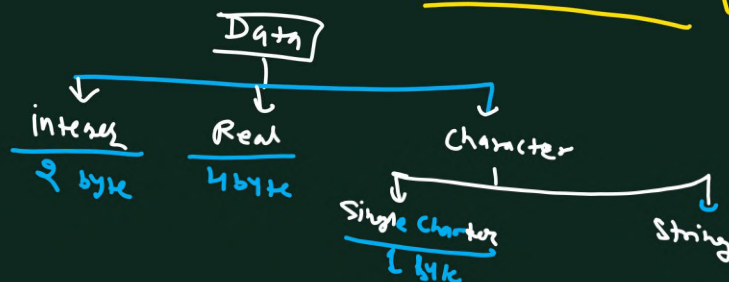
Recap :-

2D Array :-

Matrix Addition V.V.7

Matrix Multiplication (T.T.I)

" Transpose V.I





[String]

String is an array of characters which is used to store more than one characters under a single variable name.

Syntax:- `datatype name [size];`

Example:-

`Char ch[5];`

In Memory →

A	B	C	Δ	'\0'
---	---	---	---	------

 Note:- When we declare a string in Memory Compiler automatically puts a NULL value ('0') at the last of string.
 NULL [zero]

Input Operations :- We can use `scanf` function for input in C but it will stop taking input after a blank space. So that we use `gets` function to take input with space.

eg:-

<code>Char ch[5];</code> <code>scanf("%s", ch);</code>	<code>Char ch[5];</code> <code>gets(ch);</code>
---	--

Output Operation:- We may use `printf` function for output but there is a special function named `puts` for taking output from a string

Example :-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char ch[20];
    clrscr();
    printf("Enter your name=");
    gets(ch);
    printf("your name=");
    puts(ch);
    getch();
}
```

* Some String handling function in C *

- ① Strlen :- It returns Length of a string.
- ② Strcpy :- It copies one string to another
- ③ Strcat :- It appends (Join) one string to other.
- ④ Strrev :- It reverse the given string.
- ⑤ Strcmp :- It compares two strings and returns zero(0) if both are same.

Note:- To use these function we must use header file
#include<string.h>

Q:- W.A.P in C to check given string is Palindrome or not

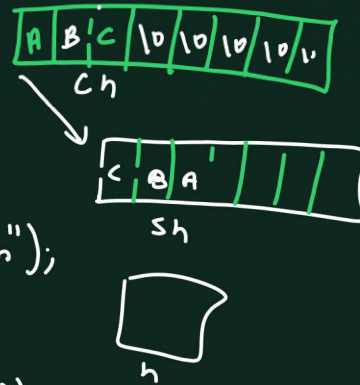
```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char ch[50], sh[50];
    int n;
    clrscr();
    printf("Enter any string to check Palindrome=");
    gets(ch);
```



```

strcpy(sh, ch);
strrev(sh);
n = strcmp(sh, ch);
if (n == 0)
{
    printf("given string is Palindrome");
}
else
{
    printf("given string is Not Palindrome");
}
getch();
}

```



Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24

Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24

Recap

String :- Array of characters
 '\0'

Input → gets

Output → puts

← string.h

TECHNICAL CLASSES

* Structure *

It is a secondary data type which is used to store different types of value under a single variable name. Such as record of a student.

So that it is also known as **Heterogeneous** data structure.

[name → string
 roll → int
 age → float]

Syntax:-

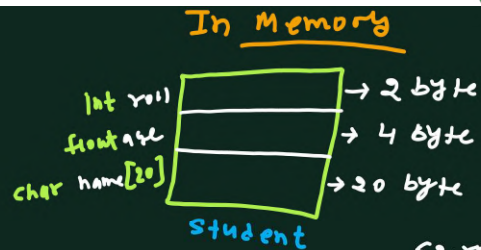
```

struct <structure name>
{
    structure member variable1;
    "      "      " 2;
    "      "      " 3;
    :
};

struct <structure name> structure variable1, variable2, ...;
```

Example: `struct student`
`{`
`int roll;`
`float age;`
`char name[20];`
`};`

`struct student s1, s2, s3;`



TECHNICAL CLASSES

[Input/output in a structure]

We use `scanf` & `printf` function with `(.)` dot operator.

Q> W.A.P In C to demonstrate I/O in a structure

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
struct student
```

```
{
```

```
int roll;
```

```
float age;
```

```
};
```

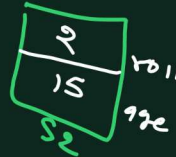
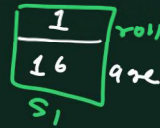
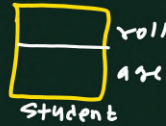
```
struct student s1, s2;
```



```

void main()
{
    clrscr();
    printf("Enter first student details=");
    scanf("%d %f", &s1.roll, &s1.age);
    printf("Enter second student details=");
    scanf("%d %f", &s2.roll, &s2.age);
    printf("details of first student");
    printf("roll=%d age=%f", s1.roll, s1.age);
    printf("details of second student");
    printf("roll=%d age=%f", s2.roll, s2.age);
    getch();
}

```



Q.7 This structure will take 10 byte

```

struct s
{
    int a, b, c;
    float b;
}

```

Note:- All member variables of a structure takes their own memory space.

* Union *

It is same as structure but different in term of memory. In structure all member variables takes their own space but in Union the **biggest** will take memory and other will share same.



Syntax:-

```
Union <Union name>
{
    Union variable1;
    Union variable2;
    :
    :
};

Union <Union name> v1, v2, ..., vn;
```

TECHNICAL CLASSES

Example

```
Union Student
{
    int roll;
    float age;
};

Union Student S1, S2;
```

In memory



Student
Total Size = 4 byte
In Union the space is given to the biggest. remaining will share the space.

Recap

Structure	Union
Each variable have own memory	The biggest will take
Struct keyword	Union
all variable will show their value	The recent will show the value

* Function *

It is a **Sub Program**. It means it is a **block** of statement that is written to perform any task. We use function in a Program to break down a Large Program into **various small parts**. So that it can be easily managed.

Error finding & Correction becomes easier.

By using function in a Program we can save time by using previous codes. So that we can **reuse** the previous block of statement.

There are two types of Functions
[Function]

System defined
Function

Example:-

```
printf ( );
scanf ( );
sort ( );
pow ( );
```

User defined
function

① System defined function : These function are already written in C compiler. It means we don't have to write what & how it will work. We only use it by calling it.

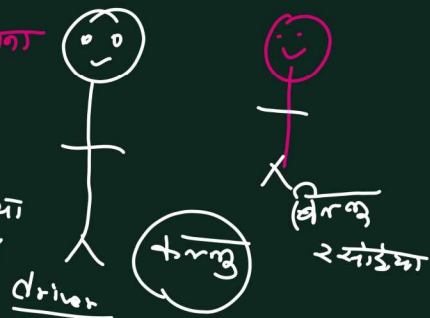
Example :- `clrscr();` | `printf( );`
`getch();` | `scanf( );`

User defined function :- User defined functions are written by a programmer to perform any specific task. There are 3 parts of a user defined function.

① Function declaration → *ગણતરી કરવા માટે જાણ કરવા*

② function call → *અમારે જે કંપ્યુટર કોડ લખવો*

③ Function definition :- *ગણતરી કરવા માટે કોડ લખવો, જે કોડ લખવા માટે*





① Function declaration :- In this part we write prototype of a function.

Syntax :- Return type functionname (List of Arguments);

eg :-
 int add (int a, int b);
 float
 char
 void

Function call :- In this part we call a function to work.

Syntax: functionname (actual value);

Example:-
 add (5, 10);

② Function definition :- In this part we write the program that what a function will do.

Syntax:-
 Return type functionname (List of Arguments);
 {
 statements;
 ...
 }

Example :-
 int add (int a, int b);
 {
 int c;
 C = a + b;
 return (c);
 }



Q> W.A.P in C to add any given two numbers using function.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int add (int a, int b); /* Function declaration */
```

```
void main ( )
```

```
{
    int m, n, r;
```

```
    clrscr();
```

```
    printf("Enter any two no. to add");
```

```
    scanf("%d %d", &m, &n);
```

10
m

20
n

30
r

Enter any two no. to add =
10
20

```
    r = add(m, n); /* Function calling */
```

```
    printf("Add = %d", r);
    getch();
```

```
}
int add (int a, int b) /* Function definition */
{
    int c;
    c = a + b;
    return (c);
}
```

[Function]

↓
Syntax

Declare

Call

Definition

[Recall]

① System defined
or
Library function

② User defined

Categories of user defined function

- ① function with Argument without return type
- ② function with Argument with return type
- ③ function without Argument with return type
- ④ function without Argument without return type

① Function with Argument without return type.

```

#include <stdio.h>
#include <conio.h>
void add (int a, int b);
void main ( )
{
    int m, n;
    clrscr();
    printf("Enter any two no. to add = ");
    scanf("%d %d", &m, &n);

```

10 20
m n

```

    add(m, n);
    getch();
}
void add (int a, int b)
{
    int c;
    c = a + b;
    printf("Add = %d", c);
}

```

30
c

Enter any two no = 10
20
Add = 30



[2] Function with Argument with return type

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int add (int a, int b);
```

```
void main ( )
```

```
{
```

```
int m, n, r;
```

```
clrscr();
```

```
printf("Enter any two no. to Add=");
```

```
scanf("%d %d", &m, &n);
```

10 20 30
m n r

```
r = add(m, n);
```

```
printf("Add = %d", r);
```

```
getch();
```

```
} int add (int a, int b)
```

```
{
```

```
int c;
```

```
c = a + b;
```

```
return(c);
```

```
}
```

Enter any two no. 10
20
Add = 30

30
c

[3] Function without argument with return type

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int add ( );
```

```
void main ( )
```

```
{
```

```
int r;
```

```
clrscr();
```

```
r = add ( );
```

```
printf("Add = %d", r);
```

```
getch();
```

```
}
```

30
r



```

int add ( )
{
    int a, b, c;
    printf("Enter any two no. to add=");
    scanf("%d %d", &a, &b);
    c = a + b;
    return(c);
}

```

Enter any two no. to
10
20
add = 30

TECHNICAL CLASSES

[u] Function without Argument without return type

```

#include <stdio.h>
#include <conio.h>
void add ( );
void main ( )
{
    clrscr ( );
    add ( );
    getch ( );
}

```

```

void add ( )
{
    int a, b, c;
    printf("Enter any two no to add=");
    scanf("%d %d", &a, &b);
    c = a + b;
    printf("Add = %d", c);
}

```

10 20 30
a b c
Enter any two no. to
10
Add = 20

Note 1:- main function is a user defined function

Note 2:- If we don't write return type (blank) of a function by default it will return integer

eg:-

```
int add ( );
```

```
return 0;
```

Recap [Types of Function]

[4 Category]

⊛ a function by default return an integer
eg:-

```
add ( );
```

[Recursive Function]

When a function calls itself it is known as recursion or recursive function.

Example :- In general

```
Guest function → int add ( )
Host function → void main ( )
{
    ...
    add ( ); ←
}
```

In recursive function

```
#
#
void main ( )
{
    printf ("Hello");
    main ( );
}
```

hello
...
Infinite time



Q. > W.A.P in C to Print factorial of any number using recursive function

```
#include <stdio.h>
#include <conio.h>
int fact (int n);
void main()
{
    int m, r;
    clrscr();
    printf("Enter any no to find factorial:");
    scanf("%d", &m);
    → r = fact(m);
    printf("Factorial = %d", r);
    getch();
}
```

5 120
m r

```
int fact(int n)
{
    if (n == 0 || n == 1)
    {
        return (1);
    }
    else
    {
        n = n * fact(n-1);
    }
    return (n);
}
```

Diagram illustrating the recursive calculation of factorial for n=5:

- 5 * 4 * 3 * 2 * 1
- 5 * 24
- 120
- 6
- 3 * 2
- 2 * 1
- 1 * 1

[Pointer]

Pointer is also a variable which is used to store address of any variable. It means a pointer doesn't store any value but address only. Pointer works on address so that the program in which pointer is used will be work faster than normal variable.

Declaration of Pointer variable

To declare a Pointer variable we use $*$ (Astrick) with data type & variable name.

Example:- `int *a;`  $*a \rightarrow$ Pointer variable

Initialization of Pointer variable

To initialize a Pointer variable we use $\&$ (address of operator)

Example:- `a = &b;`

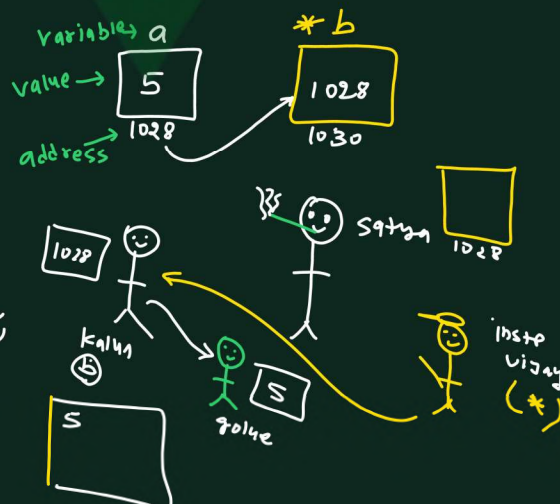
Accessing Value

To access value from variable we use $*$ (value at the address of operator) with `printf` function.

Example:-

`printf("%d", *a);`

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a = 5;
    int *b;
    b = &a;
    printf("%d", *b);
    getch();
}
```



Function call by value

⇒ `add(a, b);`

⇒ It uses simple variable.

function call by Reference

`add(&a, &b);`

It uses Pointer variables

TECHNICAL CLASSES

[Function call by reference]

W.A.P in C to SWAP any two variables value using Pointer.

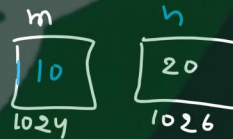
```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void swap(int *a, int *b);
```

```
void main()
```

```
{
    int m, n;
    clrscr();
```



```
printf("Enter value of m & n =");
```

```
scanf("%d %d", &m, &n);
```

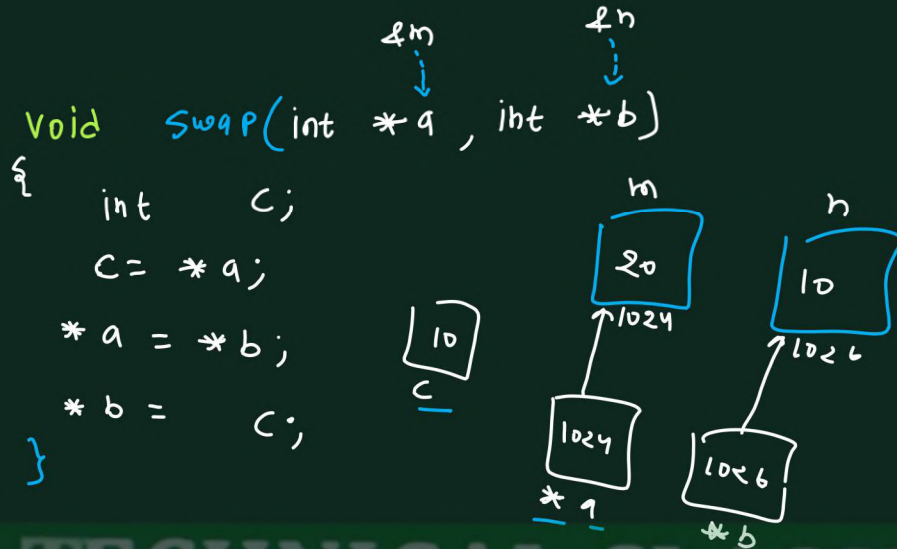
```
printf("before swapping m = %d and n = %d", m, n);
```

```
    swap(&m, &n);
```

```
printf("After swapping m = %d and n = %d", m, n);
```

```
getch();
```

```
}
```



TECHNICAL CLASSES

Recap :-
Function, Pointer, Call by value &
 " " Reference

[Break Statement]

Break Statement is used to stop the execution of a Loop or it is used to exit from a Switch, Case Statement.

Example :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    clrscr();
    for(i=1; i<=10; i++)
    {
        if(i>5)
        {
            break;
        }
        printf("%d\n", i);
        getch();
    }
}
```

1
2
3
4
5

[Continue Statement]

Continue Statement is used in a Loop. When it is executed by compiler it send back the Control to upper part by ignoring statements written below it.



Example:-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    clrscr();
    for(i=1; i<=10; i++)
    {
        if(i >= 5 && i <= 8) ✓
        {
            continue;
        }
        printf("%d\n", i);
    }
    getch();
}
```

1
2
3
4
5
6
7
8
9
10

[goto Statement]

goto Statement is a **unconditional jumping** Statement. Which transfers control from one part of Program to another part without using any condition. It works with a Label Statement. Using of goto Statement in a Program is a bad manner.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    goto abc; /* goto statement */
    printf("welcome");
    printf("Hello");
    abc: /* Label Statement */
    printf("good evening");
    getch();
}
```

Enumerated data Type

It is a user defined data type that consists of integer constant, each assign a name.

Example :- `enum { Sun, Mon, ... };`
 0 1

TECHNICAL CLASSES

```
#include <stdio.h>
#include <conio.h>
enum grade
{
    Low, medium, high
};
void main()
{
    enum grade a;
    clrscr();
    a = high;
    printf("%d", a);
    getch();
}
```

→ 2

Comment Line in C

Comments are written in a program for developer so that they may get information about program.

Comment line is ignored by compiler. It means any thing writing in comment line will be not displayed on output screen.



There are two types of Comment Line in C

[1] Single Line Comment Line :- (//)

It is used for One Line Comment

Example :-
`goto abc; // it is used for Jumping`

TECHNICAL CLASSES

[2] Multiline Comment Line (/* */)

Any thing written between /* and */ is ignored by compiler.

Eg.-
/*
.....
.....
..... */

Since 2011



Technical Classes

तकनीकी शिक्षा के लिए No.1 संस्थान

Online -- Download
Technical Classes
App from PlayStore



Offline & Online

- Polytechnic + B.Tech Semester
- For ME, CE, EE, ECE, CSE
- BCECE LEET Entrance Exam
- Polytechnic Entrance Exam
- SSC, RRB & All J.E. Exams

Contact – 93347 89450/ 91555 63777



Offline -- Technical Classes, Rajiv Nagar, Patna-24