

UNIT-1 : Introduction to components of computer system

Lec-1

Today's Target

✓ Block Diagram of Computer.

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

Introduction to Components of a Computer System:

Memory, Processor, I/O Devices, Storage, Operating System, Concept of Assembler, Compiler, Interpreter, Loader and Linker.

Idea of Algorithm: Representation of Algorithm, Flowchart, Pseudo Code with Examples, From Algorithms to Programs, Source Code.

Programming Basics: Structure of C Program, Writing and Executing the First C Program, Syntax and Logical Errors in Compilation, Object and Executable Code. Components of C Language. Standard I/O in C, Fundamental Data types, Variables and Memory Locations, Storage Classes.

Computer:

- A computer is an electronic device that takes input from the user, processes it under the control of set of instructions and generates output. It has the ability to store, retrieve, and process data.
- A computer is made up of multiple parts and components that facilitate user functionality. A computer has two primary categories:

Hardware:

- Physical structure that houses a computer's processor, memory, storage, communication ports and peripheral devices

Software:

- Includes operating system (OS) and software applications

QUESTIONS

What is digital computer? Also explain block diagram of digital computer in detail.

OR

**Discuss/Illustrate the major components of a digital computer with suitable block diagram.
Also discuss the functions of these components.**

(2022-23,21-22,2019-20,2017-18,2014-15, 2013-14, 2010-11,2009-10,2008-09)

7 Marks

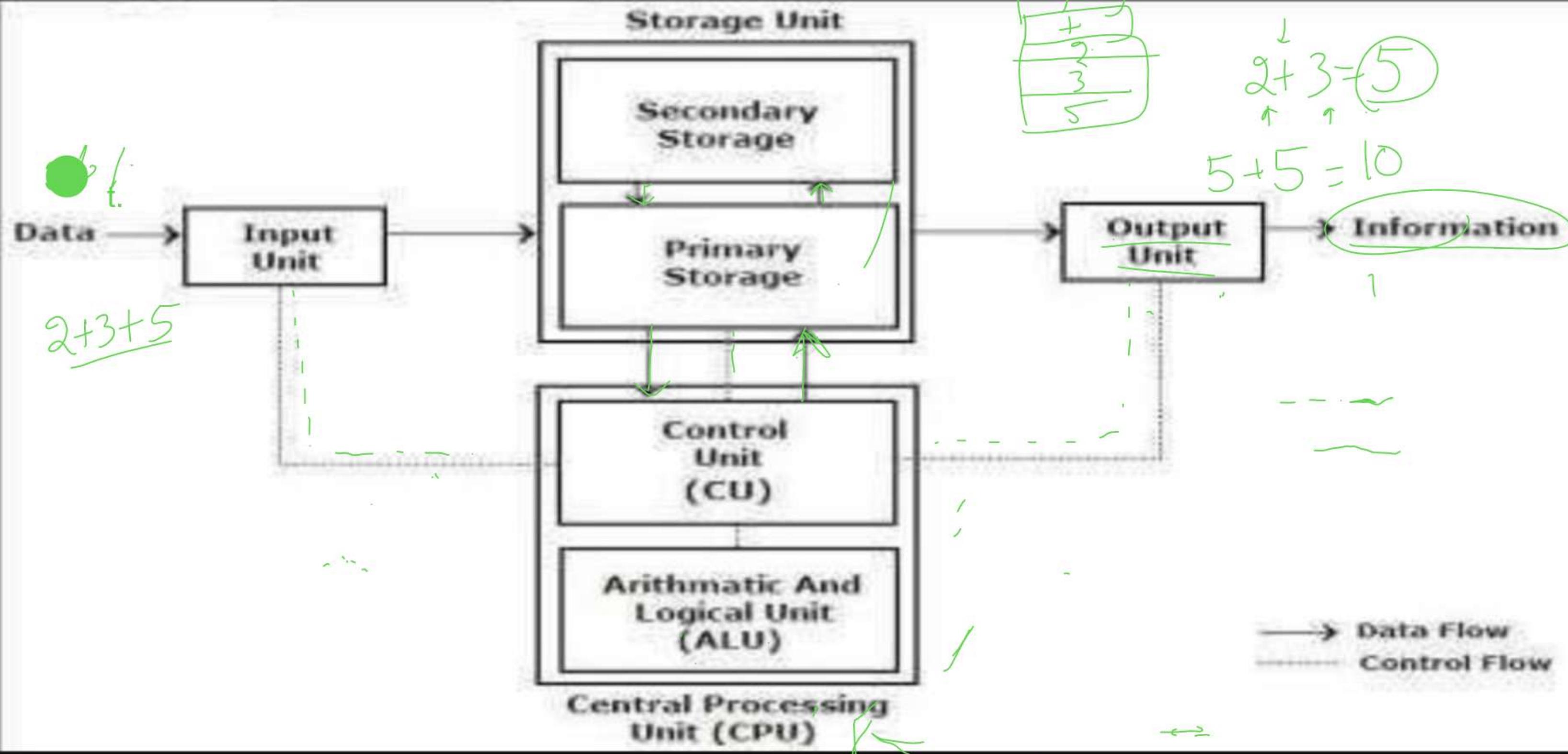
What is digital computer?

Digital Computer: Digital computers are the computer systems which uses the binary number system, which has two digits: 0 and 1 and performs many computational tasks.

- ❑ **Inputting:** It is the process of entering data & instructions to the computer system.
- ❑ **Storing:** The data & instructions are stored for either initial or additional processing.
- ❑ **Processing:** It requires performing arithmetic or logical operation on the saved data to convert it into useful information.
- ❑ **Outputting:** It is the process of producing the output data to the end user.
- ❑ **Controlling:** The above operations have to be directed in a particular sequence to be completed.

$$2 + 3 = 5$$

BLOCK DAIGRAM OF COMPUTER



Block diagram of digital computer in detail

Input Unit:

- We need to first enter the data & instruction in the computer system, before any computation begins.
- This task is done by the input devices. (Eg: keyboard, mouse, scanner, digital camera etc).

Storage Unit:

- The data & instruction that are entered have to be stored in the computer. Similarly, the end results & the intermediate results also have to be stored somewhere before being passed to the output unit.
- The storage unit provides solution to all these issues. This storage unit is designed to save the initial data, the intermediate result & the final result.

$$\begin{array}{r} 9+3+5 \\ 2+5+5 \\ \hline 10 \end{array}$$

$$\begin{array}{r} 2 \\ 3 \\ 5 \\ + \\ 6 \\ \hline 10 \end{array}$$

Primary storage

- It is also called main memory of computer.
- It is computer memory that a processor or computer accesses first or directly.
- It allows a processor to access running execution applications and are temporarily stored in a specific memory location.
- It is of two types RAM and ROM.

$$2 \times 3 + 5$$

Secondary Storage:

- The secondary storage, also called as the auxiliary storage, can retain information even when the system is off.
- It is basically used for holding the program instructions & data on which the computer is not working on currently, but needs to process them later.

Block diagram of digital computer in detail

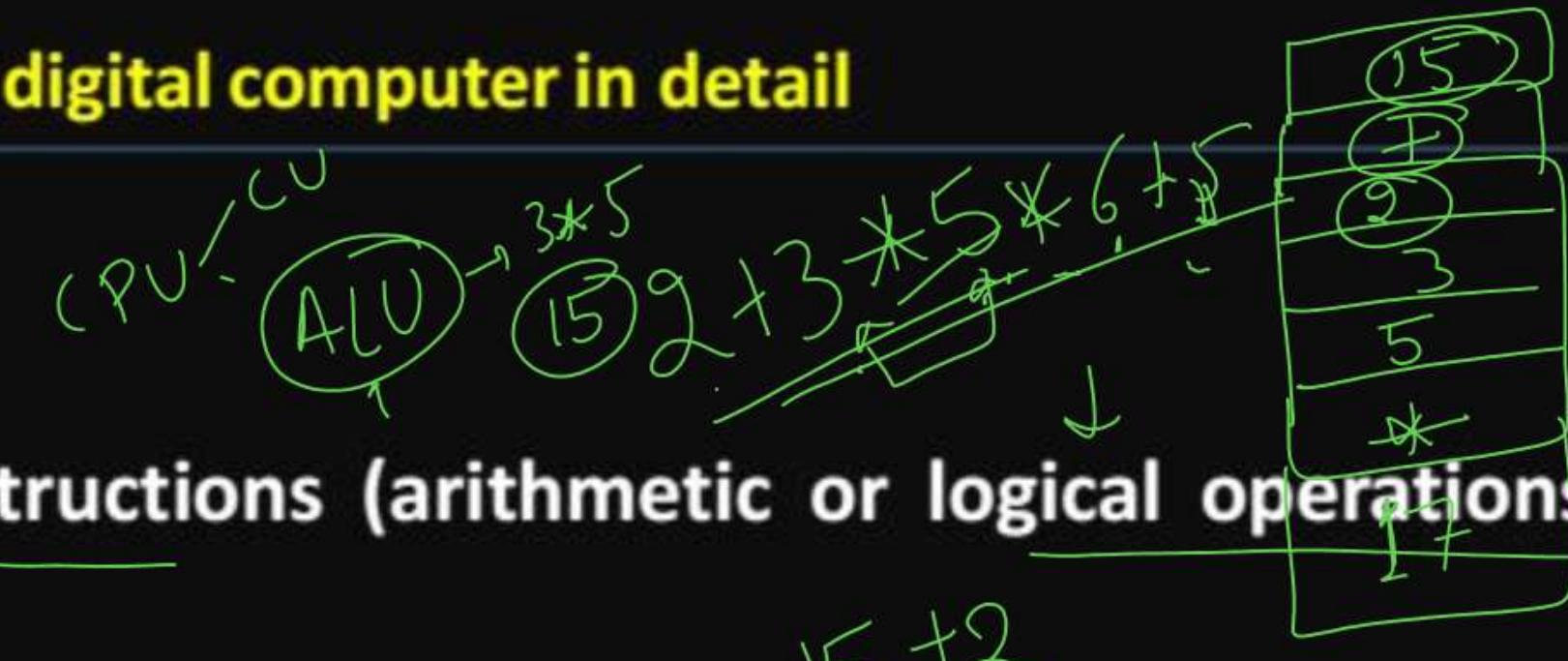
Central Processing Unit:

- Central Processing Unit (CPU) has two major components: **ALU** (Arithmetic Logic Unit) and **CU** (Control Unit).
- The CPU is the **brain of the computer**.
- In a computer system, all the major calculations & comparisons are made inside the CPU.

Block diagram of digital computer in detail

Arithmetic Logic Unit:

- The actual execution of the instructions (arithmetic or logical operations) takes place over here.
- The ALU performs simple addition, subtraction, multiplication, division, and logic operations, such as OR and AND.
- Intermediate results that are generated in ALU are temporarily transferred back to the primary storage, until needed later. Hence, data may move from the primary storage to ALU & back again to storage, many times, before the processing is done.



$$15 + 2 \\ 17$$

Block diagram of digital computer in detail

Control Unit:

- This unit controls the operations of all parts of the computer but does not carry out any actual data processing.
- It manages and coordinates all the units of the system. It also communicates with Input/output devices for transfer of data or results from the storage units.



Output Unit:

- Output unit accepts the results produced by the computer in coded form.
- It converts these coded results to human readable form.
- Finally, it displays the converted results to the outside world with the help of output devices (E.g. monitors, printers, projectors etc).

Download **Gateway Classes** Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : Introduction to components of computer system

Lec-2

Today's Target

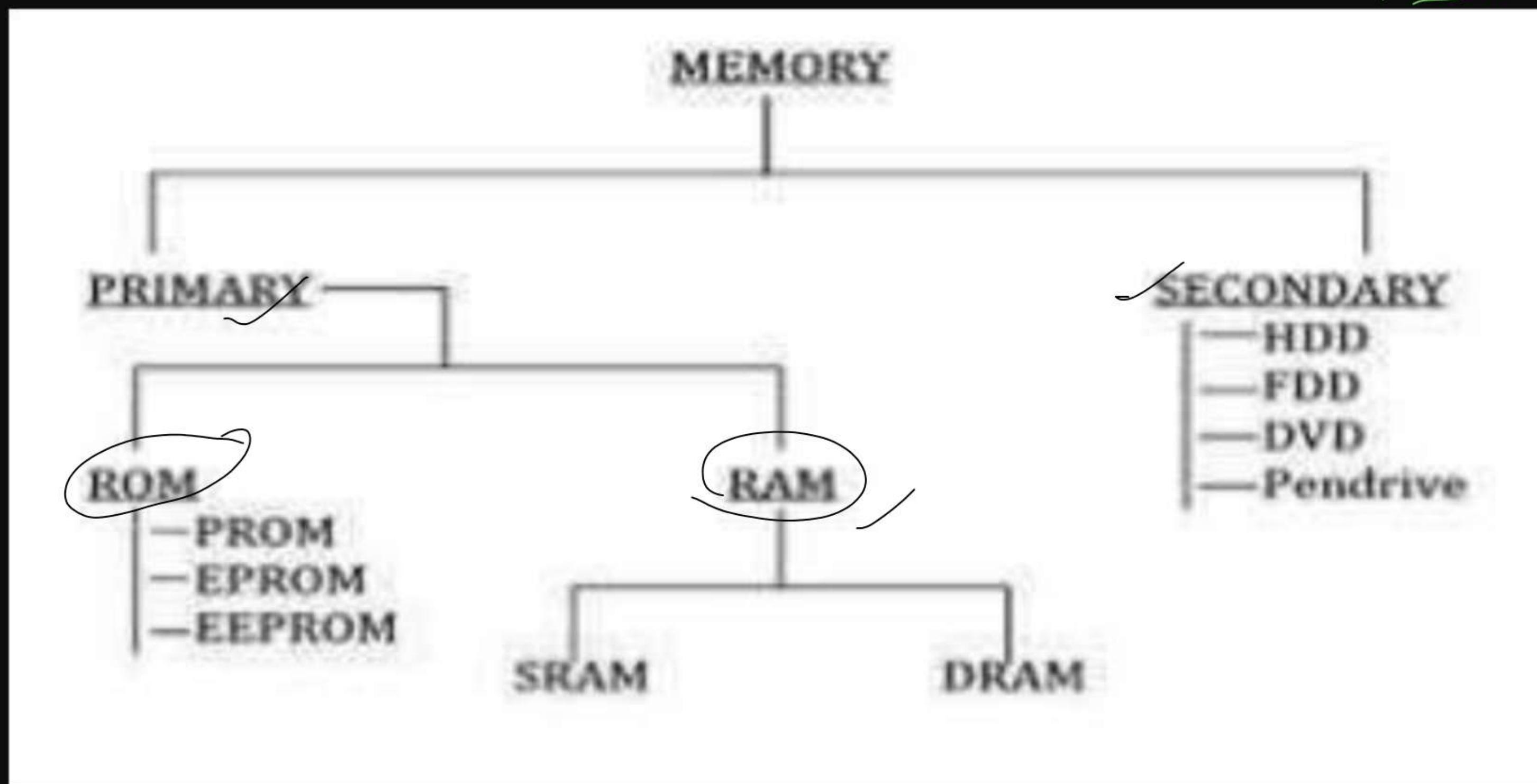
- ✓ Types of memory in a computer system.
- ✓ Hierarchical structure of computer system.

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

Ques) Explain different types of memories in a computer system. (2014-15, 2017-18)

7 Marks



Ques) Differentiate between PRIMARY and SECONDARY MEMORY.

PRIMARY MEMORY(MAIN / INTERNAL MEMORY)

Primary memory is temporary

Primary memory is directly accessible by Processor/CPU

Nature of Parts of Primary memory varies RAM- volatile in nature. ROM- Non-volatile

Primary memory devices are more expensive than secondary storage devices

Accessing data from primary memory is faster

Primary memories are made of semiconductors.

Examples: RAM, ROM, Cache memory, PROM, EPROM, Registers etc

SECONDARY MEMORY(AUXILIARY/ EXTERNAL MEMORY)

Secondary memory is permanent

Secondary memory is not directly accessible by CPU

It's always Non-volatile in nature

Secondary memory devices are less expensive when compare to primary memory devices

Accessing data from secondary memory is slower

Secondary memories are made of magnetic and optical material

Examples: Hard Disk, Floppy Disk, Magnetic Tapes etc

Ques) Differentiate between RAM and ROM

RAM (RANDOM ACCESS MEMORY) (read/write memory)

RAM is volatile in nature as it automatically erased when computer is shutdown

RAM can be directly accessed by processor

Static RAM and Dynamic RAM

Costlier than ROM

The data stored in RAM is used by the Central Processing Unit (CPU) to process current instructions

RAM has a higher capacity when compared to ROM

The speed of RAM is higher when compared to ROM

ROM (READ ONLY MEMORY)

ROM is non-volatile in nature

ROM can't be directly accessed by the processor

PROM, EPROM, EEPROM

Cheaper than RAM

the data stored in ROM is used to bootstrap the computer.

ROM has a lower capacity compared to RAM

The speed of Read-only Memory is slower when compared to RAM

Ques) Differentiate between RAM and ROM

Static RAM

- Static RAM also known as SRAM, retain stored information as long as the power supply is ON.
- SRAM are of higher cost and consume more power .They have higher speed than Dynamic RAM.

Dynamic RAM

- Dynamic RAM also known as DRAM, its stored information in a very short time (a few milliseconds) even though the power supply is ON.
- The Dynamic RAM is cheaper and moderate speed and also they consume less power.

Ques) Differentiate between-

• PROM (Programmable Read Only Memory)-

\downarrow byte = 8 bits

EPROM (Erasable Programmable Read Only Memory)

010001

082

bit
& byte

EEPROM (Electrically Erasable Programmable Read Only Memory)

- PROM chip is programmable ROM.
- It is PROM chips to write data once and read many.
- Once chip has been programmed, the recorded information cannot be changed.
- PROM is also non-volatile memory.

① 01011

- EPROM chip can be programmed time and again by erasing the information stored earlier in it.
- Information stored in EPROM exposing the chip for some time ultraviolet light.

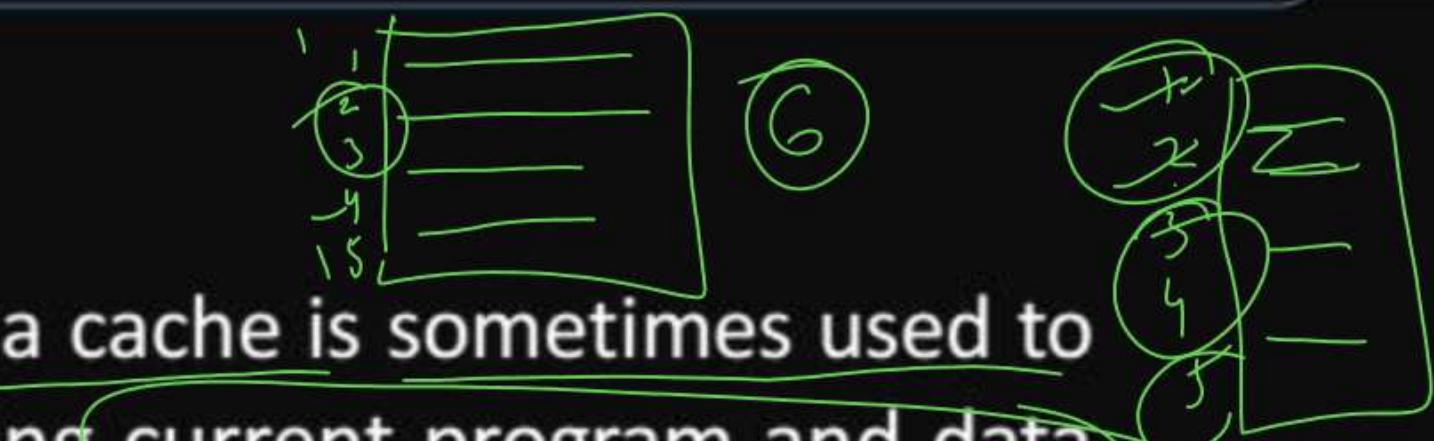


- The EEPROM is programmed and erased by special electrical waves in millisecond.
- A single byte of a data or the entire contents of device can be erased.



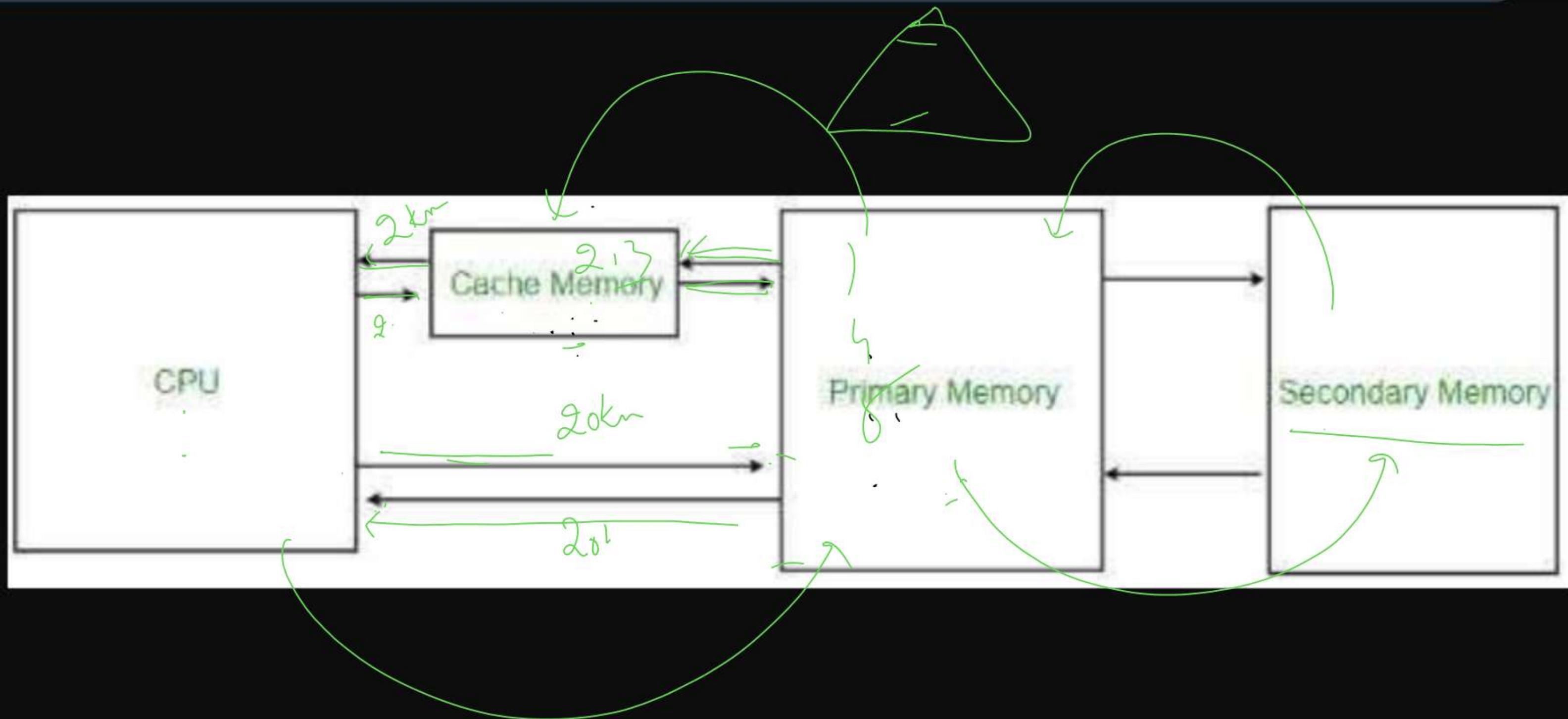
Cache Memory:

- A special very high-speed memory called a cache is sometimes used to increase the speed of processing by making current program and data available to the CPU at a rapid rate.
- The purpose of cache memory is to store program instructions and data that are used repeatedly in the operation of programs or information that the CPU is likely to need next.
- The computer processor can access this information quickly from the cache rather than having to get it from computer's main memory. In the Memory Hierarchy System, a cache memory is placed between CPU & main memory.



155

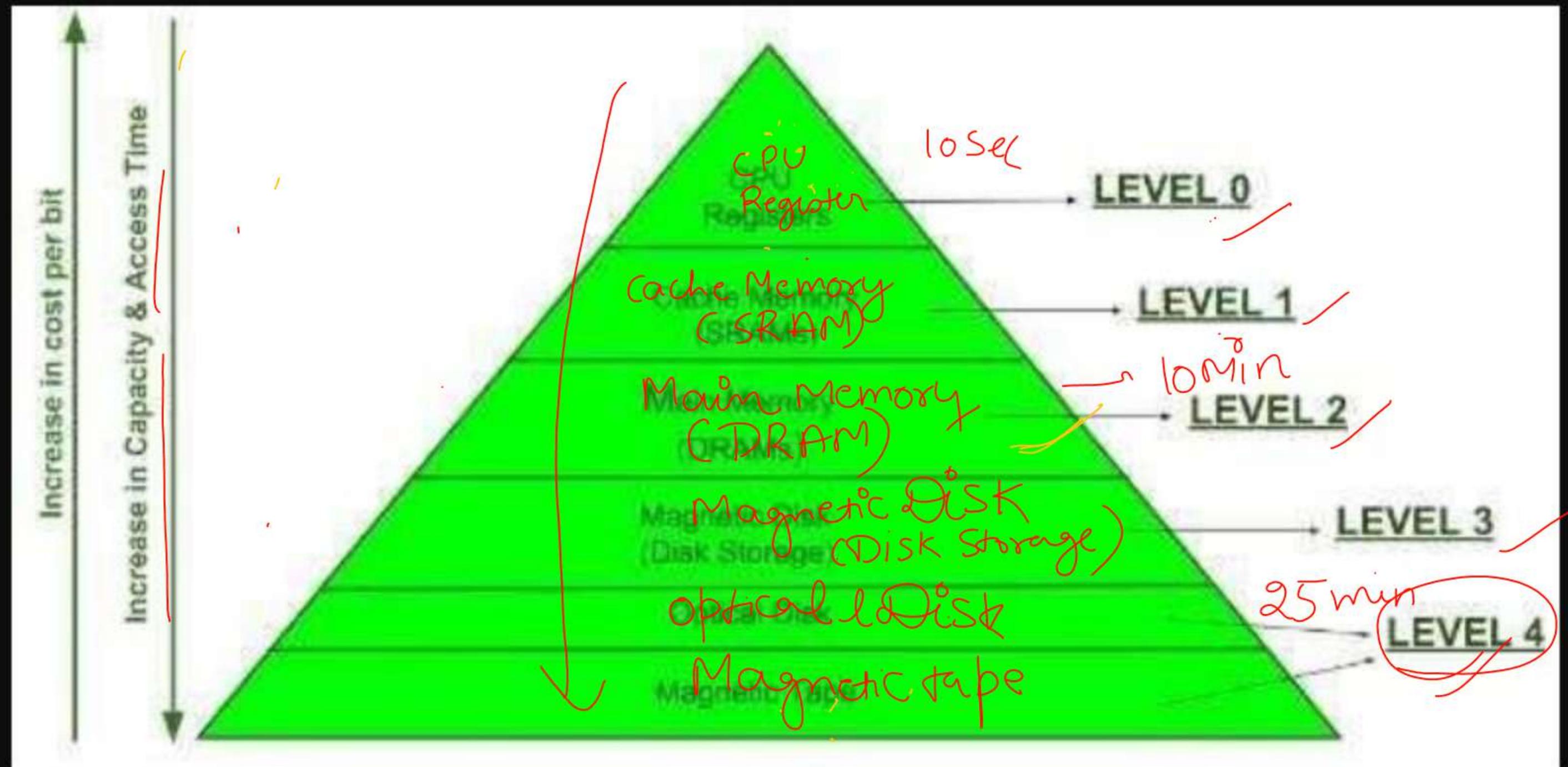
Ques) Write short note on cache memory.



Ques) Draw and explain the memory hierarchical structure of computer system.

(2022-23, 2018-19)

2 marks



Ques) Draw and explain the memory hierarchical structure of computer system.
(2018-19)

Memory Hierarchy Design is divided into 2 main types:

External Memory or Secondary Memory -

- Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.

Internal Memory or Primary Memory -

- Comprising of Main Memory, Cache Memory & CPU registers.
- This is directly accessible by the processor.

(34)

(Level 0, Level 1
Level 2)

**Ques) Draw and explain the memory hierarchical structure of computer system.
(2018-19)**

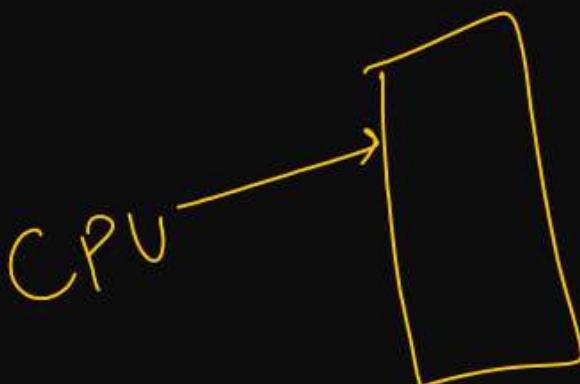
We can infer the following characteristics of Memory Hierarchy Design from above figure:

Capacity:

- It is the global volume of information the memory can store.
- As we move from top to bottom in the Hierarchy, the capacity increases.

Access Time:

- It is the time interval between the read/write request and the availability of the data.
- As we move from top to bottom in the Hierarchy, the access time increases.



Performance:

- Earlier when the computer system was designed without Memory Hierarchy design, the speed gap increases between the CPU registers and Main Memory due to large difference in access time.
- This results in lower performance of the system and thus, enhancement was required.
- This enhancement was made in the form of Memory Hierarchy Design because of which the performance of the system increases.
- One of the most significant ways to increase system performance is minimizing how far down the memory hierarchy one has to go to manipulate data.

Cost per bit: Internal Memory is costlier than External Memory.

Download **Gateway Classes** Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : Introduction to components of computer system

Lec-3

Today's Target

- ✓ High level language and low level language
- ✓ Translator and its types.
- ✓ Programming language and its type.

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

Ques) Differentiate between high level and low level programming languages. (2021-22)

High level language	low level language
It is programmer friendly language.	It is a machine friendly language.
It is easy to understand.	It is tough to understand.
It is simple to debug	It is complex to debug comparatively.
It is simple to maintain	It is complex to maintain comparatively.
It is portable.	It is non-portable.
It can run on any platform.	It is machine-dependent.
It needs compiler or interpreter for translation	It needs compiler or interpreter for translation.
High level language is less memory efficient.	Low level language is high memory efficient.

Ques) What do you mean by translator? Explain Assembler, compiler and interpreter.(2017-18)(2018-19)

Translators:

- A program written in high-level language is called as source code.
- To convert the source code into machine code, translators are needed.
- A translator takes a program written in source language as input and converts it into a program in target language as output.
Low level language is high memory efficient.
- It also detects and reports the error during translation.

Ques) What do you mean by translator? Explain Assembler, compiler and interpreter.(2017-18)(2018-19)

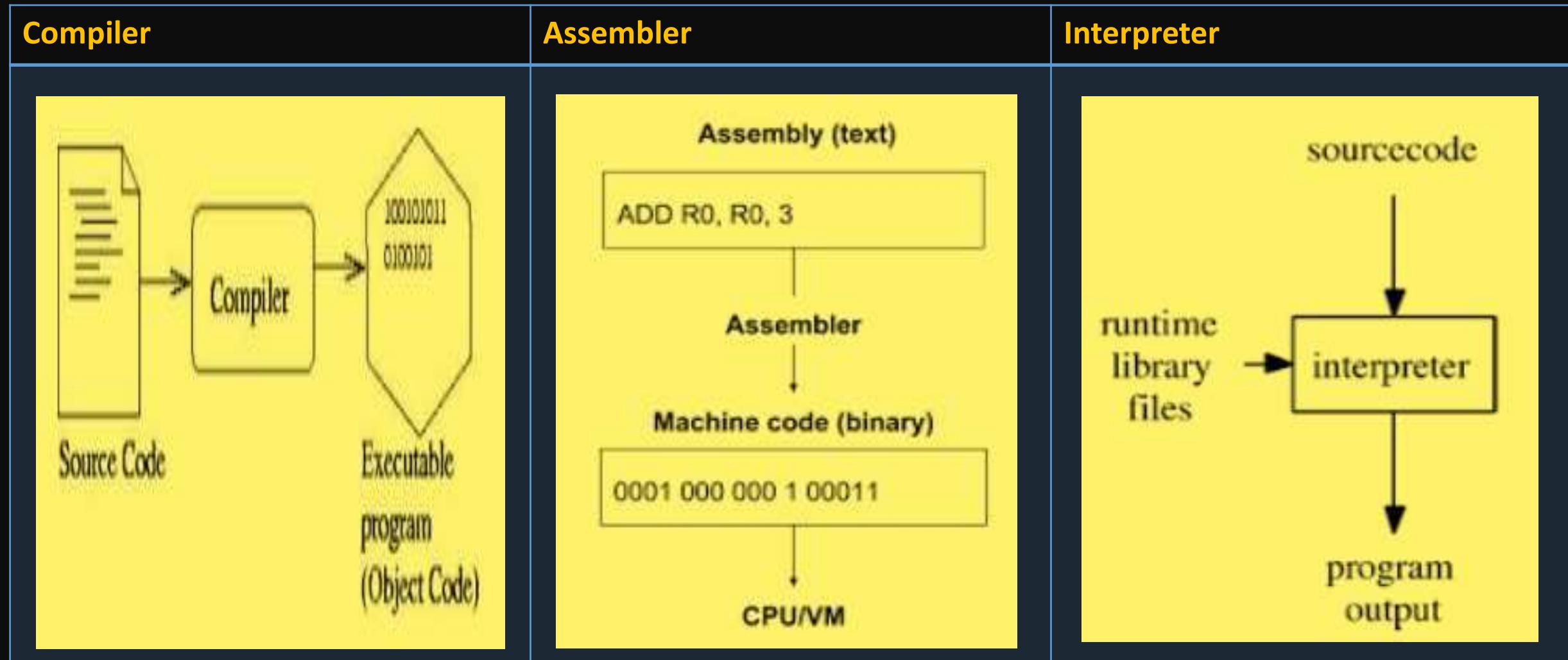
Roles of translator:

1. Translating the high-level language program input into an equivalent machine language program.
2. Providing diagnostic messages wherever the programmer violates specification of the high-level language program

Ques) What do you mean by translator? Explain Assembler, compiler and interpreter.(2017-18)(2018-19)

Compiler	Assembler	Interpreter
<ul style="list-style-type: none">Compiler is a translator which is used to convert programs in high-level language to low-level languageIt translates the entire program and also reports the errors in source program encountered during the translation.	<ul style="list-style-type: none">Assembler is a translator which is used to translate the assembly language code into machine language	<ul style="list-style-type: none">Interpreter is a translator which is used to convert programs in high-level language to low-level language.Interpreter translates line by line and reports the error once it encountered during the translation process.It gives better error diagnostics than a compiler

Ques) What do you mean by translator? Explain Assembler, compiler and interpreter.(2017-18)(2018-19)



Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Computer Program:

- A program is a set of instructions following the rules of the chosen language.

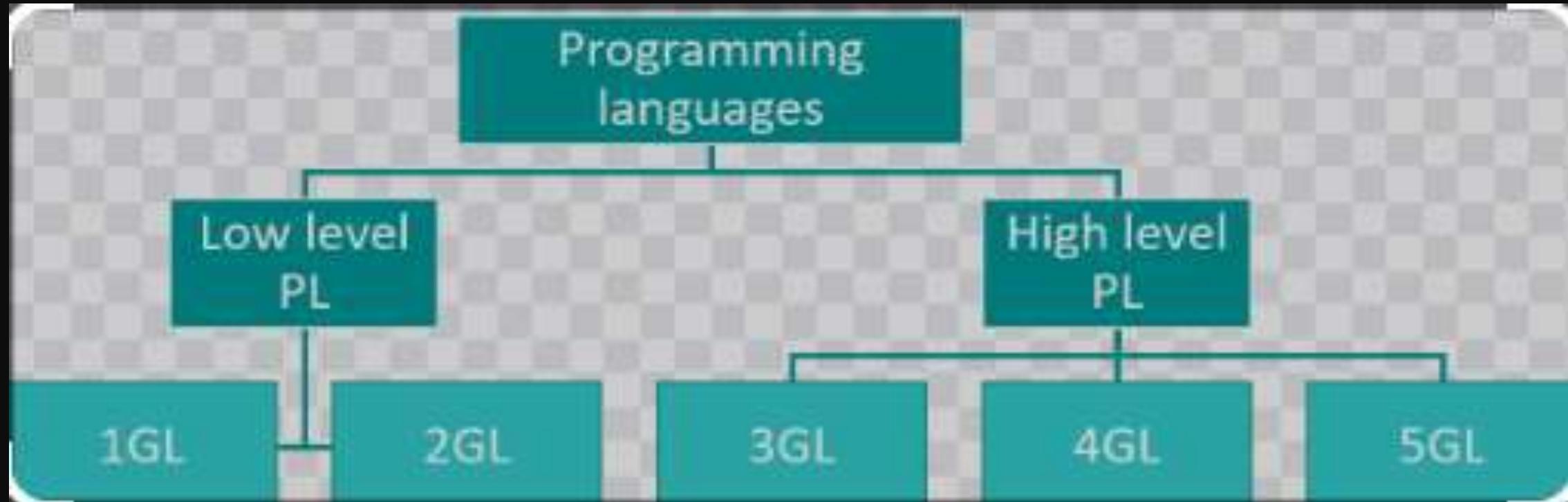
Programming Language:

- A vocabulary and set of grammatical rules (syntax) for instructing a computer to perform specific tasks.
- Programming languages can be used to create computer programs.
- The term programming language usually refers to high-level languages, such as C, C++, JAVA.

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Generations of Programming Language:

- So far Five Generations of programming languages have been defined.
- A brief introduction of each of the five generations is given below



Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

First Generation Programming Language (1GL):

- First generation of programming language refers to machine language.
- Machine language is lower level language is the combination of binary digits.
- These languages directly talk to hardware. While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers.
- Programs written in machine language are not portable.

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Machine Instruction	Machine Operation
0000 0000	Stop
0000 0001	Rotate bristles left
0000 0010	Rotate bristles right
0000 0100	Go back to start of program
0000 1000	Skip next instruction if switch is off

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Second Generation Programming Language (2GL):

- Second generation of languages is also low level language which is known as assembly language.
- A program written in assembly language consists of a series of instructions mnemonics (much like English) that correspond to a stream of executable instructions.

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

- Programs written in machine language are not portable. The problem is that the computer doesn't understand the assembly code, so we need a way to convert it to machine code using assembler.
- Example: Machine language: 10110000 01100001
Assembly language: mov a1, 91

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Third Generation Programming Language (3GL):

- Third Generation programming languages are High level Programming languages.
- They allow us to write computer code using instructions resembling everyday spoken language (for example: print, if, while) which are then translated into machine language to be executed.
- Programs written in a high level language need to be translated into machine language before they can be executed.

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Third Generation Programming Language (3GL):

- Some programming languages use a compiler to perform this translation and others use an interpreter.
- Programs written in high level languages are portable

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Objectives of high-level languages:

- To relieve the programmer of the tedious task of writing programs in machine language and assembly languages.
- To provide programs that can be used on more than one type of machine with very few changes.
- To allow the programmer more time to focus on understanding the user's needs and designing the software required meeting those needs. Examples of High-level Language: C, C++, JAVA, PHYTON

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Fourth Generation Programming Language (4GL):

General characteristics of 4GL are:

- a) Closer to human languages
- b) Portable
- c) Simple and requires less effort than 3GL

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Objectives of fourth generation languages:

1. Increasing the speed of developing programs and minimizing user effort to obtain information from computer.
2. Decreasing the skill level required of users so that they can concentrate on the application rather than coding, and thus solves their own problems without the help of a professional programmer.

Ques) What do you mean by programming languages? Explain different types of programming languages. (2020-21)

Fifth Generation Programming Language (5GL):

- Natural Languages represent the next step in the development of programming languages; the text of a natural language statement very closely resembles human speech.
- These languages are also designed to make the computer “smarter”.
- Languages used for writing programs for Artificial Intelligence and come under 5GL

**Ques) Differentiate between compiler and interpreter.(2017-18), (2018-19),
(2019-20),(2020-21)**

COMPILER	INTERPRETER
It takes an entire program at a time.	It takes a single line of code or instruction at a time.
It generates intermediate object code.	It does not produce any intermediate object code.
The compilation is done before execution.	Compilation and execution take place simultaneously.
Comparatively faster	Slower
Memory requirement is more due to the creation of object code.	It requires less memory as it does not create intermediate object code.
Display all errors after compilation all at the same time.	Displays error of each line one by one.
Difficult	Easier comparatively
C, C++, C#	PHP, Perl, Python, Ruby

Download **Gateway Classes** Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : Introduction to components of computer system

Lec-4

Today's Target

- ✓ What is System software and application software.
- ✓ What is Linker and Loader.

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)
2 years + experience

What is system software?

- System Software is a set of programs that control and manage the operations of computer hardware.
- It also helps application programs to execute correctly.
- System Software are designed to control the operation and extend the processing functionalities of a computer system.
- System software makes the operation of a computer more fast, effective, and secure.
- Example: Operating system

Features of System Software:

- System Software is closer to the system
- Generally written in a low-level language
- The system software is difficult to design and understand
- Fast in speed
- Less interactive
- Smaller in size
- Hard to manipulate

What is an Application Software?

- Application Software is a program that does real work for the user.
- It is mostly created to perform a specific task for a user.
- Application Software acts as a mediator between the end-user and System Software.
- It is also known as an application package.

What is an Application Software?

- It is a user-specific and is designed to meet the requirements of the user.
- Example: **Word-processing software** **Spreadsheet software**
Database software **Graphics software** **Education software**
Entertainment software

Features of Application Software:

- Perform more specialized tasks like word processing, spreadsheets, email, photo editing, etc.
- It needs more storage space as it is bigger in size.
- Easy to design and more interactive for the user.
- Generally written in a high-level language.

Difference between system software and application software.

System Software

They are designed to manage the resources of the system, like memory and process management, security, etc.

It is written in a low-level language like a machine or assembly language.

The System Software starts running when the system is powered on and runs until the system is powered off.

The System Software is a general-purpose software

It is classified as a package program or customized program.

Application Software

They are designed to fulfill the requirements of the user for performing specific tasks.

A high-level language is used to write Application Software.

The Application Software starts when the user begins, and it ends when the user stops it.

Application Software is specific purpose software.

It is classified as time-sharing, resource sharing, client-server.

Difference between system software and application software.

Installed on the computer system at the time when the operating system is installed.

Capable of running independently.

Users never interact with system software as it functions in the background.

System software are independent of the application software

System software is crucial for the effective functioning of a system.

Installed as per user's requirements.

Can't run independently.

Users interact with application software while using specific applications.

Application software needs system software to run.

Application software is not extremely important for the functioning of the system.

- It is a computer program that takes one or more object files generated by a compiler and combines them into one, executable program. Computer programs are usually made up of multiple modules that span separate object files, each being a compiled computer program..

- Loader is the program of the operating system which loads the executable from the disk into the primary memory (RAM) for execution. It allocates the memory space to the executable module in main memory and then transfers control to the beginning instruction of the program.

Difference between linker and loader (2019-20),(2020-21)

S.No.	Linker	Loader
1	A linker is an important utility program that takes the object files, produced by the assembler and compiler, and other code to join them into a single executable file.	A loader is a vital component of an operating system that is accountable for loading programs and libraries.
2	It uses an input of object code produced by the assembler and compiler.	It uses an input of executable files produced by the linker.
3	The foremost purpose of a linker is to produce executable files.	The foremost purpose of a loader is to load executable files to memory.
4	Linker is used to join all the modules.	Loader is used to allocate the address to executable files.
5	It is accountable for managing objects in the program's space.	It is accountable for setting up references that are utilized in the program.

Download **Gateway Classes** Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : Introduction to components of computer system

Lec-5

Today's Target

- ✓ What is operating System.
- ✓ Types of operation system.

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)
2 years + experience

Ques) What do you mean by operating system? Explain different types of operating systems.

Operating System:

- It is system software that performs all the basic tasks like managing file, process, and memory.
- Operating system acts as manager of all the resources, i.e. resource manager.
- Operating system becomes an interface between user and machine.

Ques) What do you mean by operating system? Explain different types of operating systems.



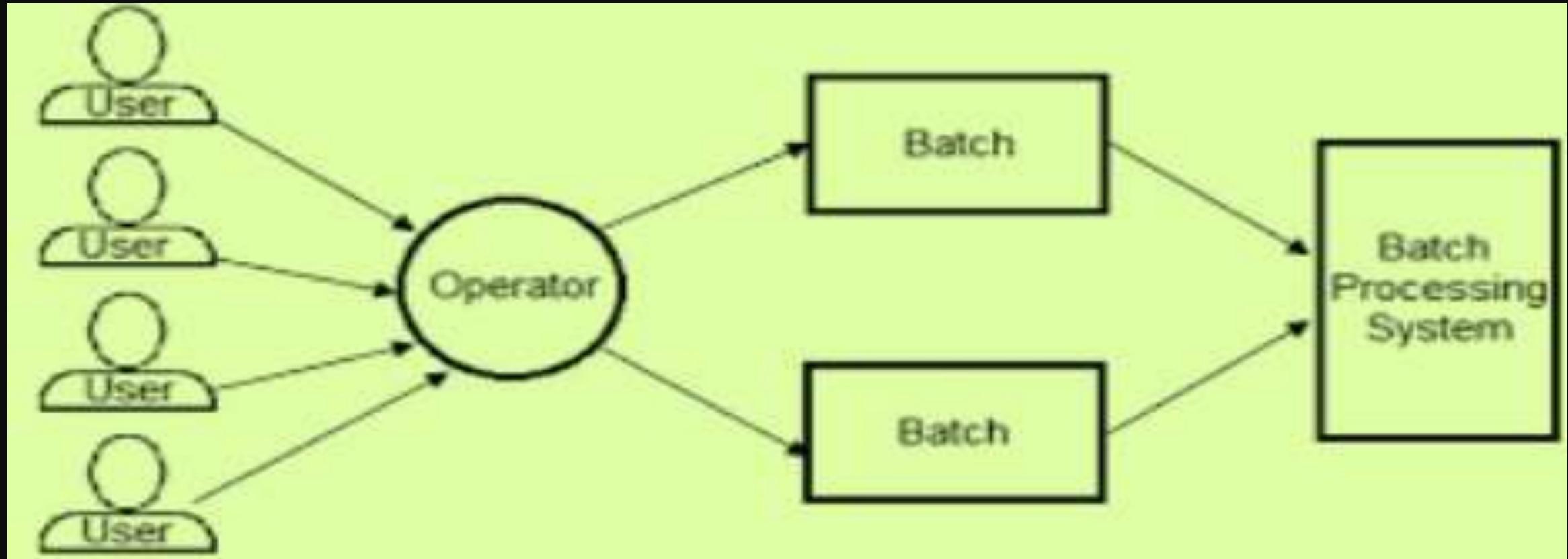
Ques) What do you mean by operating system? Explain different types of operating systems.

Types of Operating Systems:

Batch Operating System:

- This type of operating system does not interact with the computer directly.
- There is an operator which takes similar jobs having same requirement and group them into batches.
- It is the responsibility of operator to sort the jobs with similar needs.

Ques) What do you mean by operating system? Explain different types of operating systems.



Ques) What do you mean by operating system? Explain different types of operating systems.

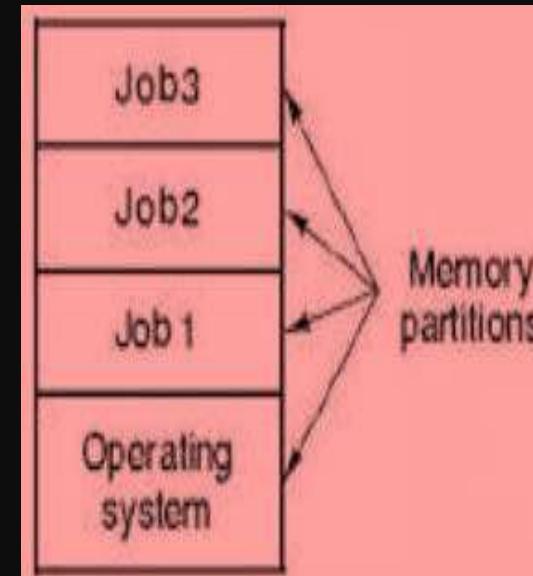
Multiprogramming OS:

- In a multiprogramming system there are one or more programs loaded in main memory which are ready to execute.
- Only one program at a time is able to get the CPU for executing its instructions while all the others are waiting their turn.
- The main idea of multiprogramming is to maximize the use of CPU time.

Ques) What do you mean by operating system? Explain different types of operating systems.

Multiprogramming OS:

- Suppose the currently running process is performing an I/O task; OS may interrupt that process and give the control to one of the other in-main-memory programs that are ready to execute. In this way, no CPU time is wasted by the system waiting for the I/O task to be completed.



Ques) What do you mean by operating system? Explain different types of operating systems.

Ques) What do you mean by operating system? Explain different types of operating systems.

Multitasking OS:

- Multitasking means working on multiple tasks simultaneously, such as using your computer while listening to music.
- Also, using a browser, search for something on the internet and create a word document that is your assignment.
- It appears that all of the tasks are taking place at the same time.
- It is not all of the tasks happening simultaneously; the processor moves between them at such a fast pace that we believe they are happening simultaneously.

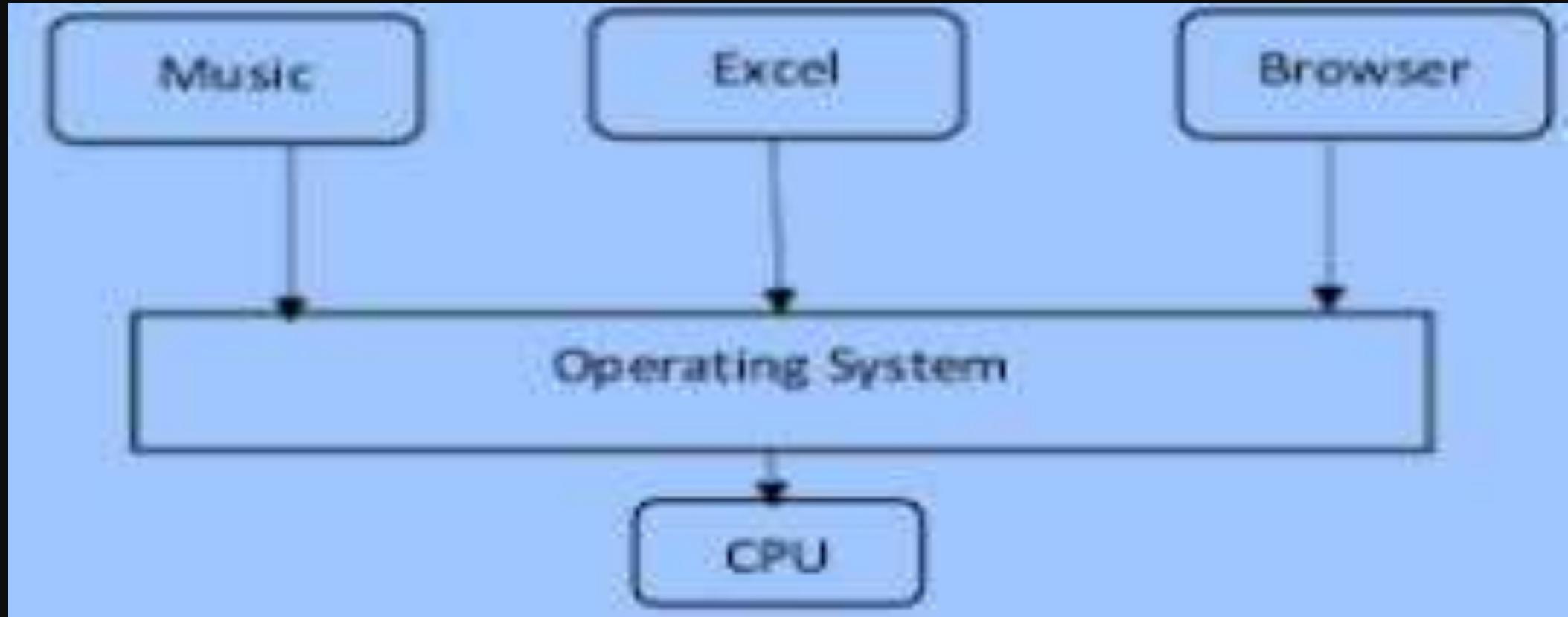
Ques) What do you mean by operating system? Explain different types of operating systems.

- Multitasking is similar to multiprogramming in that the CPU is assigned to a process for a specified period of time, i.e., '**Time quantum or time slice**', after which the CPU 'Context switches' to another process.
- It runs various programs at the same time.
- The PC requires a huge memory to execute multitasking (**RAM or ROM**).
- Its primary goal is to improve the timing of the CPU's response.
- Users can engage with the system during multitasking, for example, by typing a letter while the printing process is running.

Ques) What do you mean by operating system? Explain different types of operating systems.

- Multitasking is a highly complicated system. It is based on the time slice principle, which assigns a fixed amount of time to each activity to be completed.
- It is especially useful when a program requires a high level of parallelism.
- It provides a set amount of time for each program to run.

Ques) What do you mean by operating system? Explain different types of operating systems.

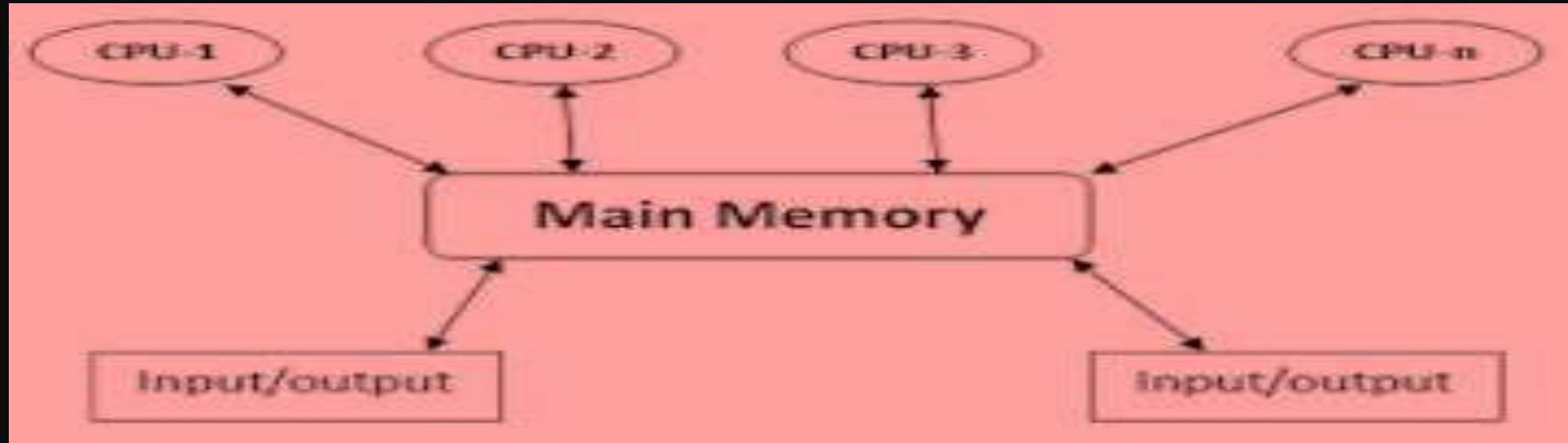


Ques) What do you mean by operating system? Explain different types of operating systems.

Multiprocessing OS:

- Multiprocessing refers to the hardware (i.e., the CPU units) rather than the software (i.e., running processes).
- If the underlying hardware provides more than one processor then that is multiprocessing.
- Several variations on the basic scheme exist, e.g., multiple cores on one die or multiple dies in one package or multiple packages in one system.
- A system can be both multiprogrammed by having multiple programs running at the same time and multiprocessing by having more than one physical processor.

Ques) What do you mean by operating system? Explain different types of operating systems.



Ques) What do you mean by operating system? Explain different types of operating systems.

Multithreading OS:

- Multithreading is an execution model that allows a single process to have multiple code segments (i.e., threads) run concurrently within the “context” of that process.
- You can think of threads as child processes that share the parent process resources but execute independently.
- Multiple threads of a single process can share the CPU in a single CPU system or (purely) run in parallel in a multiprocessing system

Ques) What do you mean by operating system? Explain different types of operating systems.

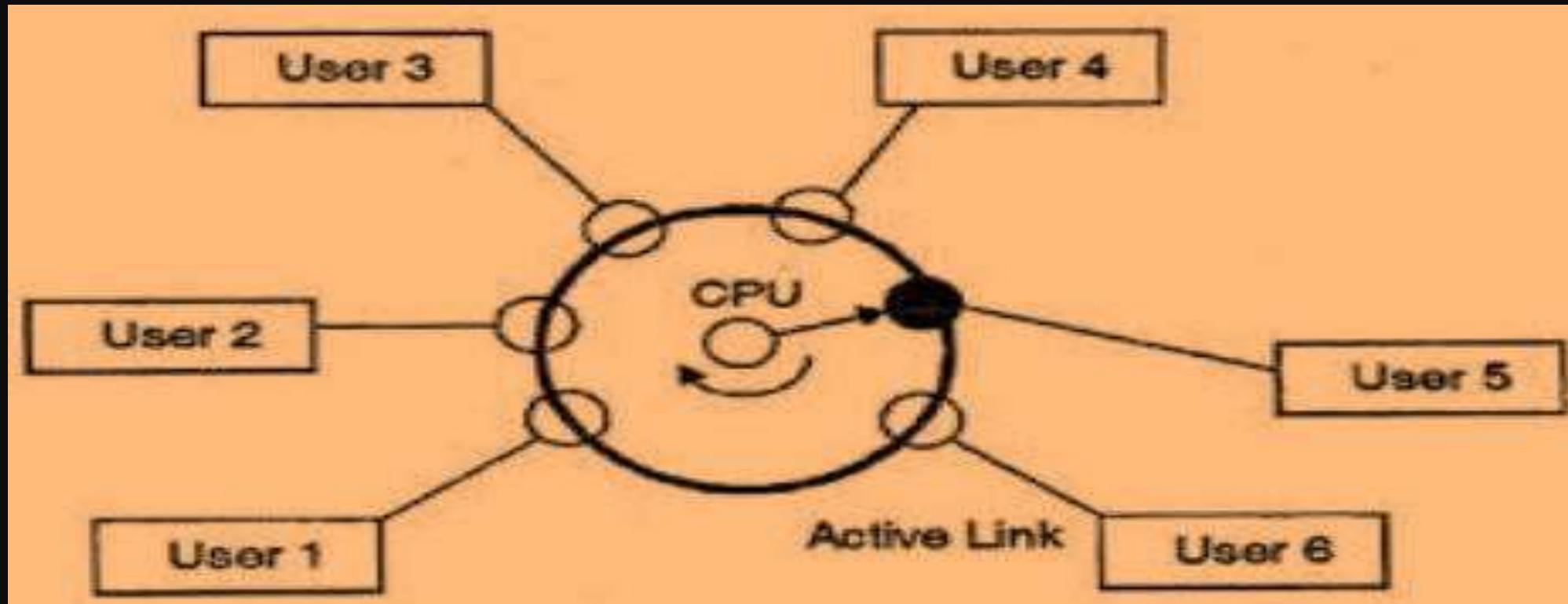


Ques) What do you mean by operating system? Explain different types of operating systems.

Time-Sharing OS:

- Each task has given some time to execute, so that all the tasks work smoothly.
- Each user gets time of CPU as they use single system. These systems are also known as Multitasking Systems.
- The task can be from single user or from different users also. The time that each task gets to execute is called quantum.
- After this time interval is over OS switches over to next task.
- Example of Time-Sharing OS is UNIX

Ques) What do you mean by operating system? Explain different types of operating systems.



Ques) What do you mean by operating system? Explain different types of operating systems.

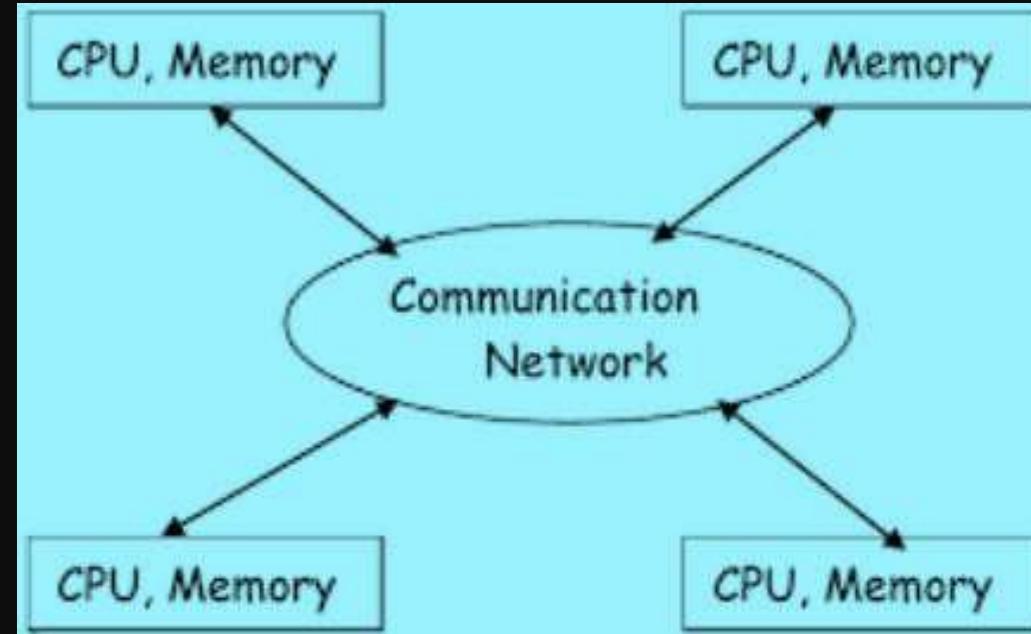
Distributed OS:

- Various autonomous interconnected computers communicate each other using a shared communication network.
- Independent systems possess their own memory unit and CPU.
- These are referred as loosely coupled systems or distributed systems.
- These systems processors differ in sizes and functions.

Ques) What do you mean by operating system? Explain different types of operating systems.

- The major benefit of working with these types of operating system is that it is always possible that one user can access the files or software which are not actually present on his system but on some other system connected within this network i.e., remote access is enabled within the devices connected in that network.
- Example of Distributed Operating System are- LOCUS

Ques) What do you mean by operating system? Explain different types of operating systems.



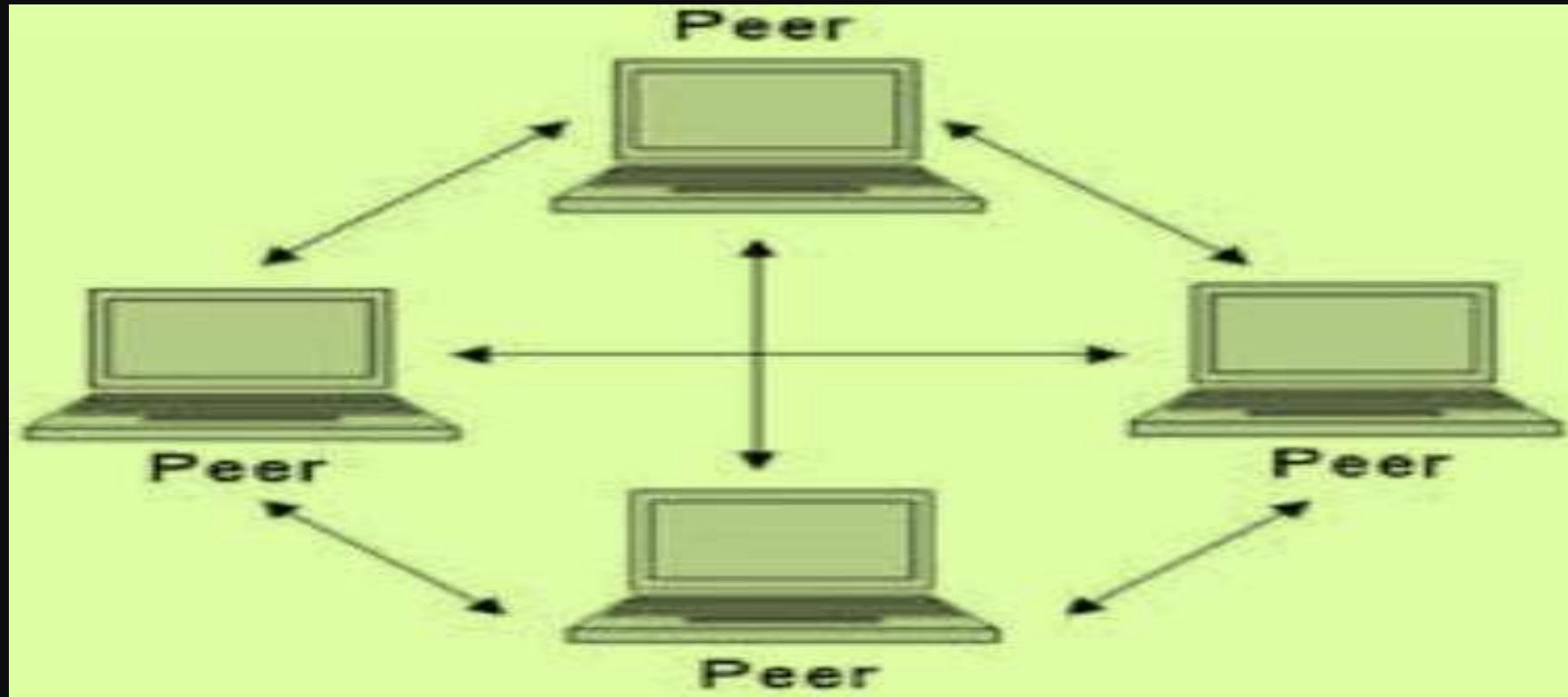
Ques) What do you mean by operating system? Explain different types of operating systems.

Network OS:

- These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions.
- These types of operating systems allows shared access of files, printers, security, applications, and other networking functions over a small private network.
- Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux etc.

Ques) What do you mean by operating system? Explain different types of operating systems.

:



Ques) What do you mean by operating system? Explain different types of operating systems.

Real-Time OS:

- These types of OS serve the real-time systems.
- The time interval required to process and respond to inputs is very small.
- This time interval is called response time.
- Real-time systems are used when there are times requirements are very strict like missile systems, air traffic control systems, robots etc.

Ques) What do you mean by operating system? Explain different types of operating systems.

Types of Real-Time Operating System which are as follows:

Hard Real-Time Systems:

- These OS are meant for the applications where time constraints are very strict and even the shortest possible delay is not acceptable.
- These systems are built for saving life like automatic parachutes or air bags which are required to be readily available in case of any accident. Virtual memory is almost never found in these systems.

Ques) What do you mean by operating system? Explain different types of operating systems.

Soft Real-Time Systems:

- These OS are for applications where time-constraint is less strict.
- Examples of Real-Time Operating Systems are: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc

Download **Gateway Classes** Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : Introduction to components of computer system

Lec-6

Today's Target

- ✓ Functions of operating system.

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

Ques1 What are the functions of operating system? Explain (2022-23, 2014-15, 2013-14).

(7marks)

35

Ques
(1)

Ques2 What is an operating system and what is its purpose? Name four popular operating system. (2014-15).

(7marks)

Ques3 What is an operating system? Give name of any four operating systems.

Write any five important features of any one type of operating system. (2011-12).

Functions of an Operating System (Resource Manager):

- An operating system has variety of functions to perform.

Some of the prominent functions of an operating system can be broadly outlined as:

- Processor Management:
 - This deals with management of the Central Processing Unit (CPU).
 - The operating system takes care of the allotment of CPU time to different processes.

- When a process finishes its CPU processing after executing for the allotted time period, this is called scheduling.
- There are various type of scheduling techniques that are used by the operating systems:

1. First Come First Serve (FCFS):

- Process which comes first in the ready queue is scheduled first.

2. Shortest Job First (SJF):

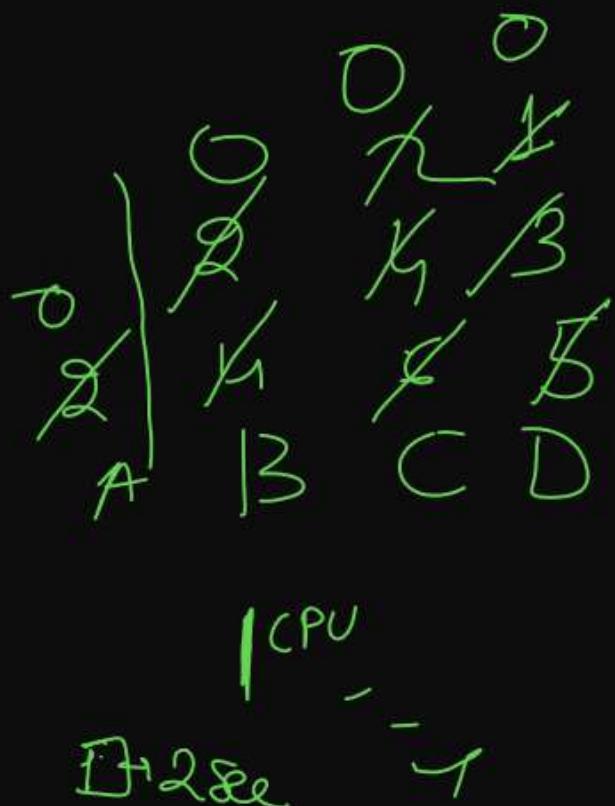
- Process which need the shortest CPU time is scheduled first

3. Round Robin Scheduling:

- Each process is assigned a fixed CPU execution time in cyclic way

Priority Based scheduling :

- In this scheduling, processes are scheduled according to their priorities.



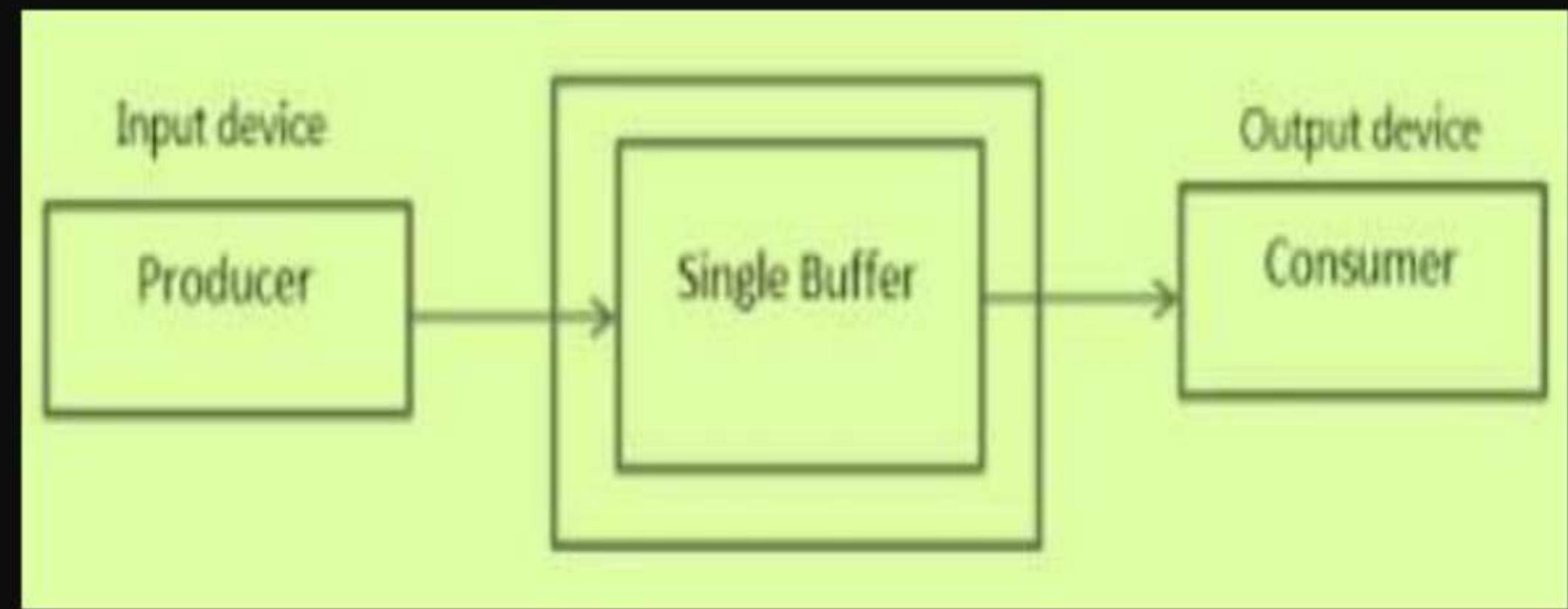
- Device (I/O) Management:
 - The Operating System communicates with hardware and the attached devices and maintains a balance between them and the CPU.
 - This is all the more important because the CPU processing speed is much higher than that of I/O devices.
 - In order to optimize the CPU time, the operating system employs two methods

Ques) What are the functions of operating system.

1. Buffering:

- The buffer is an area in the main memory used to store or hold the data temporarily.
- In other words, buffer temporarily stores data transmitted from one place to another, either between two devices or an application.
- The act of storing data temporarily in the buffer is called buffering.

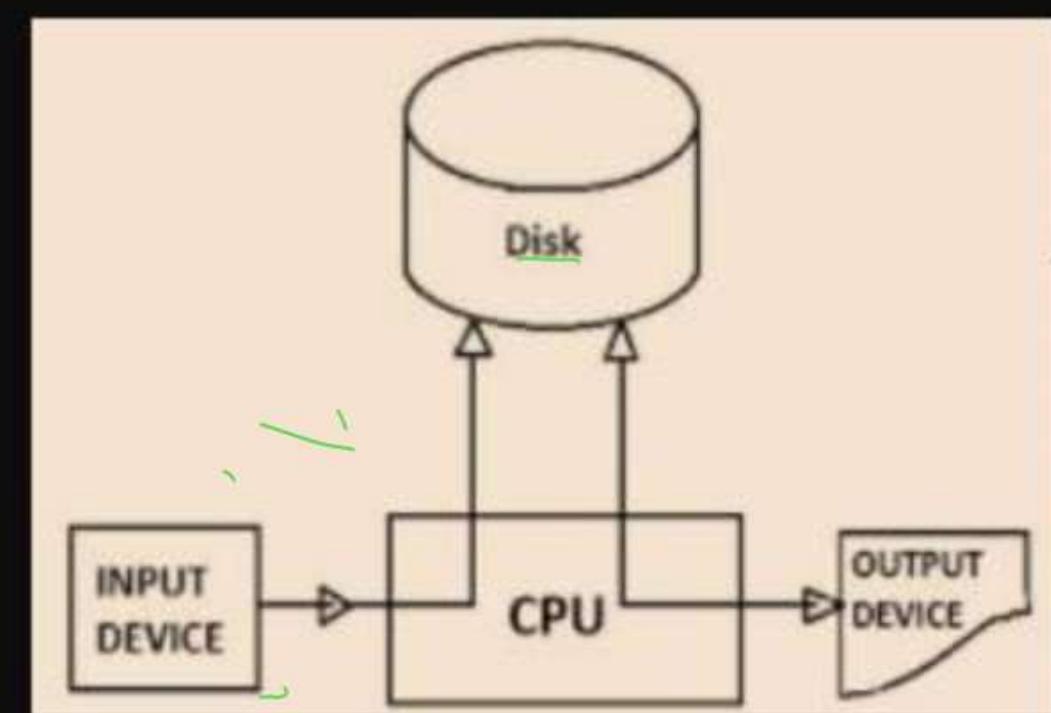
Ques) What are the functions of operating system.



2. Spooling:

- This is a device management technique used for processing of different tasks on the same input/output device.
- When there are various users on a network sharing the same resource then it can be a possibility that more than one user might give it a command at the same point of time.

- So, the operating system temporarily stores the data of every user on the hard disk of the computer to which the resource is attached.
- The individual user need not wait for the execution process to be completed.
- Instead the operating system sends the data from the hard disk to the resource one by one.
- Example: printer



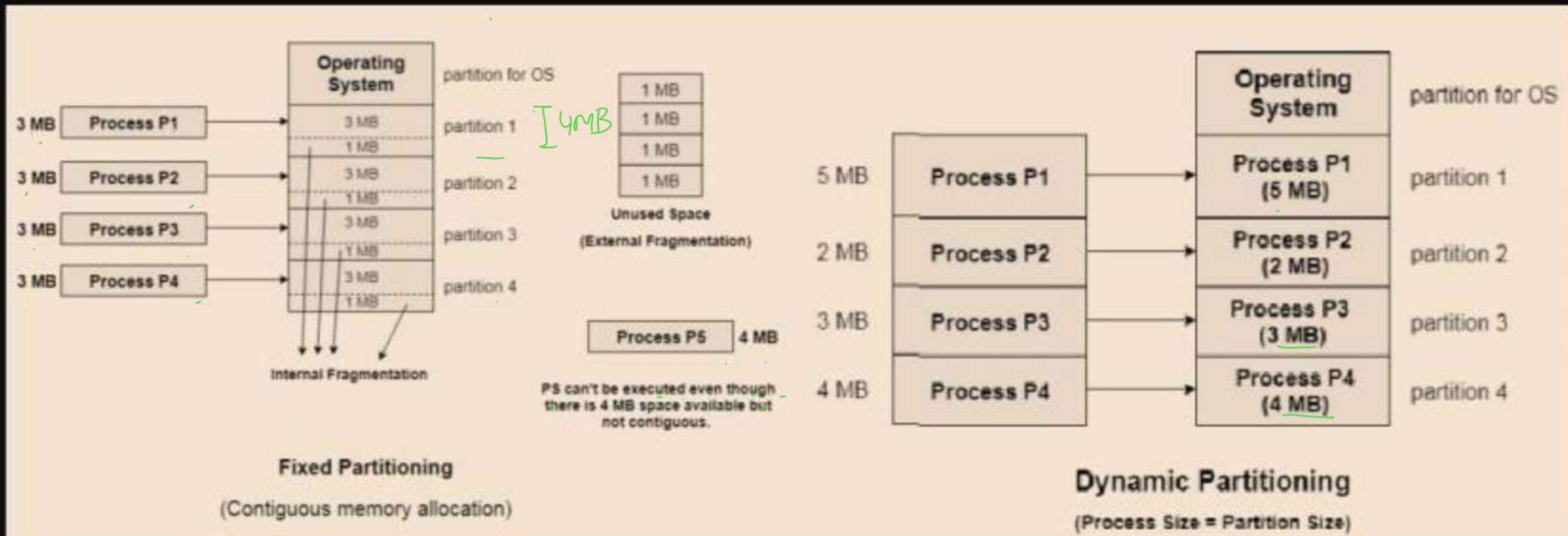
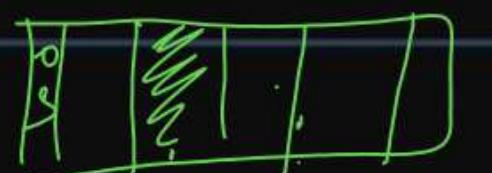
- Memory management:

- In a computer, both the CPU and the I/O devices interact with the memory.
- When a program needs to be executed it is loaded onto the main memory till the execution is completed.
- Thereafter that memory space is freed and is available for other programs.
- The common memory management techniques used by the operating system are Partitioning and Virtual Memory.

1. Partitioning:

- The total memory is divided into various partitions of same size or different sizes.
- This helps to accommodate number of programs in the memory.
- The partition can be fixed i.e. remains same for all the programs in the memory or variable i.e. memory is allocated when a program is loaded on to the memory.
- The later approach causes less wastage of memory but in due course of time, it may become fragmented.

Ques) What are the functions of operating system.



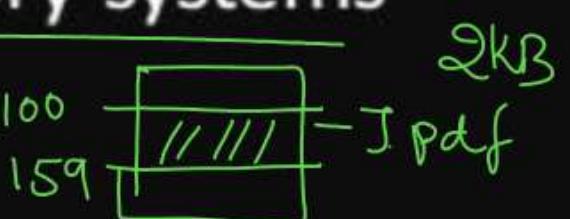
2. Virtual Memory: ✓

- This is a technique used by the operating systems which allow the user can load the programs which are larger than the main memory of the computer.
- In this technique the program is executed even if the complete program cannot be loaded inside the main memory leading to efficient memory utilization.



- File Management:

- The operating System manages the files, folders and directory systems on a computer.
- Any data on a computer is stored in the form of files and the operating system keeps information about all of them using File Allocation Table (FAT).
- The FAT stores general information about files like filename, type (text or binary), size, starting address and access mode (sequential/indexed sequential/direct/relative).



- The file manager of the operating system helps to create, edit, copy, allocate memory to the files and also updates the FAT.
- The operating system also takes care that files are opened with proper access rights to read or edit them

Download **Gateway Classes** Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : Idea of Algorithm

Lec-7

Today's Target

- ✓ What is Algorithm and its characteristics.
- ✓ What is flowchart.
- ✓ Advantages and Disadvantages of flowchart.
- ✓ Example of algorithm and flowchart
- ✓ What is pseudo code.

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

Ques) What do you mean by algorithm? Explain all the characteristics of algorithms.
(2018-19)

Algorithm:

- An algorithm is a description of a procedure which terminates with a result.
- Algorithm is a step by-step method of solving a problem.
- Purpose of algorithms is to implement re-usability in software applications.
- Once we have designed the overview of solution to any problem in
algorithmic form it can be coded in any language like C, C++, Java etc.

Ques) What do you mean by algorithm? Explain all the characteristics of algorithms.
(2018-19)

Characteristics of algorithm are as follows:

Finiteness:

➤ An algorithm terminates after a finite numbers of steps.

Definiteness:

➤ Each step in algorithm is unambiguous means that the action specified by the step cannot be interpreted (explain the meaning of) in multiple ways & can be performed without any confusion.

Ques) What do you mean by algorithm? Explain all the characteristics of algorithms.
(2018-19)

Input:

- An algorithm accepts zero or more inputs.

ax2

Output:

- An algorithm should produce at least one output.

$$\Delta \text{area} = a * a
2 * 2 = 4$$

Effectiveness:

- It consists of basic instructions that are realizable.
- This means that the instructions can be performed by using the given inputs in a finite amount of time.

Ques) What do you mean by algorithm? Explain all the characteristics of algorithms.
(2018-19)

$$\alpha = 6$$

Example 1:

Algorithm to compute the perimeter of a rectangle:

1. Start ✓
2. Enter length, L 
3. Enter width, B ✓
4. Compute Perimeter = $2*L + 2*B$
5. Display Perimeter of a rectangle
6. End

$$l = 2$$

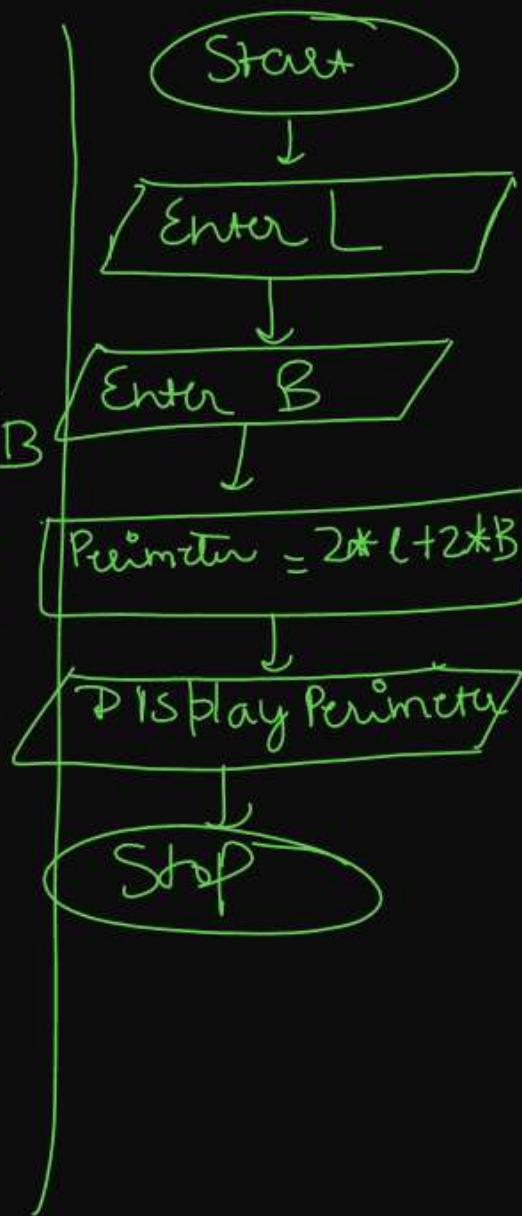
$$B = 1$$

$$\text{Perimeter} = 2*2 + 2*1$$

$$\text{Perimeter} = 6$$

$$2*(l+B)$$

$$2*l + 2*B$$



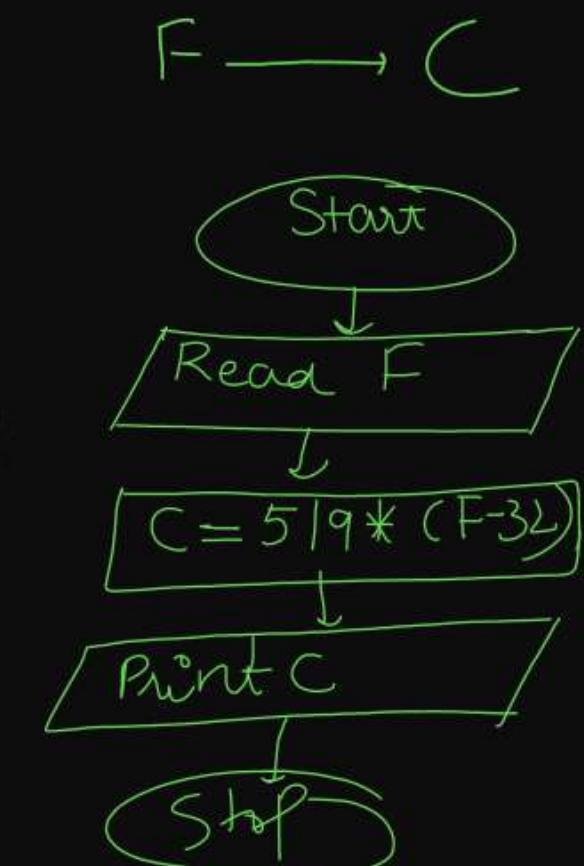
At

Ques) What do you mean by algorithm? Explain all the characteristics of algorithms.
(2018-19)

Example 2:

Algorithm to convert temperature Fahrenheit to Celsius.

1. Start Step
2. Read Temperature in Fahrenheit F
3. $C \leftarrow 5/9 * (F-32)$
4. Print Temperature in Celsius: C
5. End

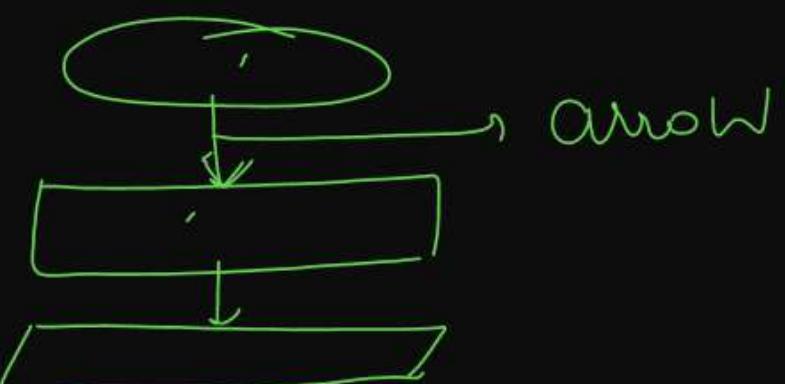


Input | Enter | Read
Start
End | Stop



Flowchart:

- A flowchart is a graphical or pictorial representation of an algorithm.
- Each step in the process is represented by a different symbol and contains a short description of the process step.
- The flow chart symbols are linked together with arrows showing the process flow direction.
- Some common flowchart symbols are listed below.

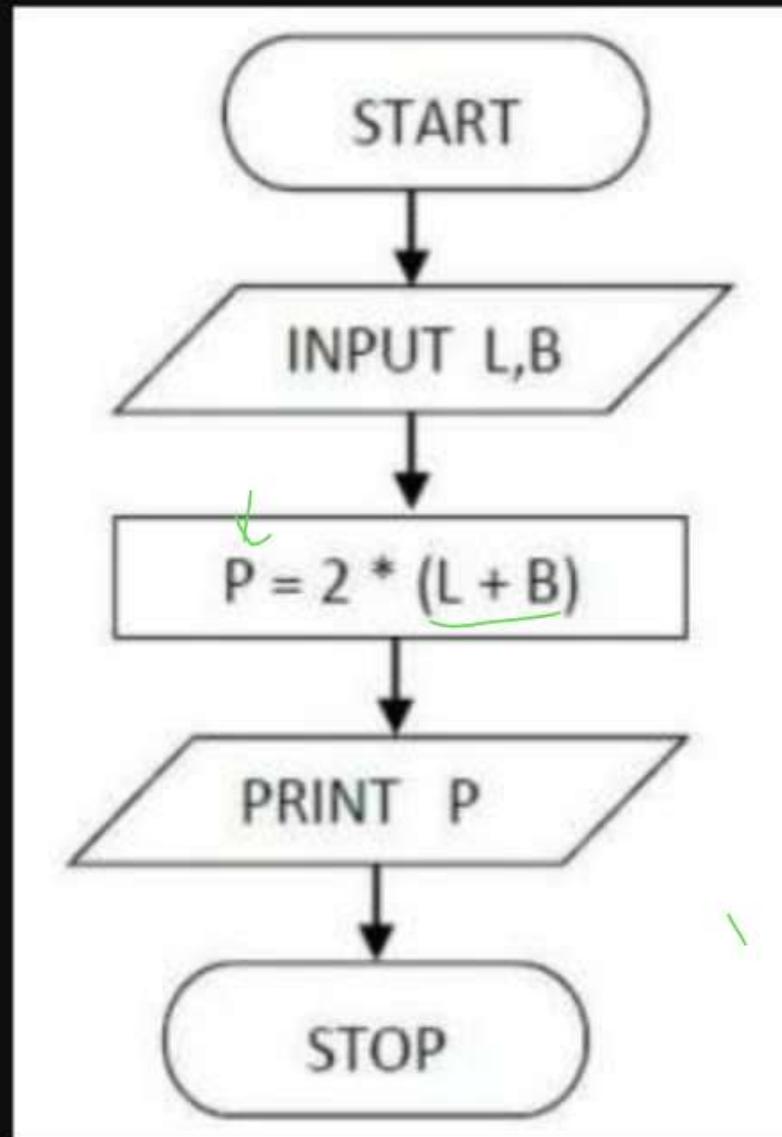


Ques) Explain flowchart in detail.(2021-22)

Symbol	Name	Function
	Start/end (Stop)	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

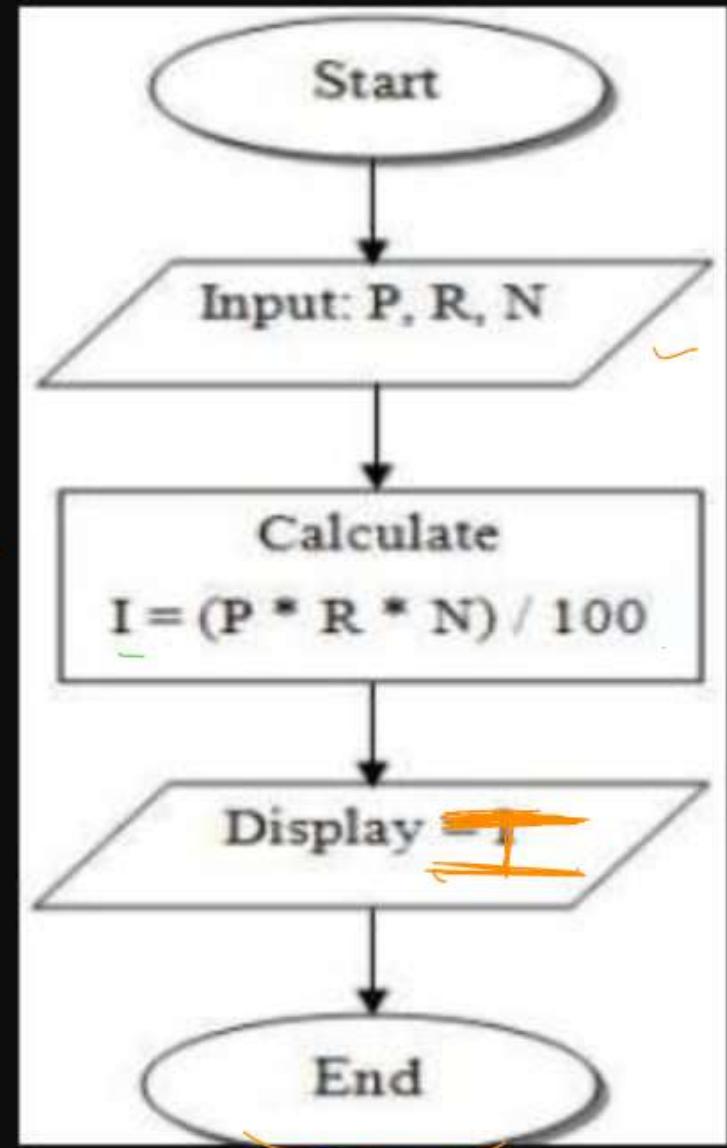
Ques) Explain flowchart in detail.(2021-22)

Flowchart to find the perimeter of the rectangle.



1. Start
 2. Input length (L) and width (B)
 3. Compute Perimeter $P = 2 * (L + B)$
 4. Print/Display P
 5. Stop

Flowchart of Simple Interest.



1. Start
 2. Input P, R, N
 3. Calculate Simple Interest
 $I = (P * R * N) / 100$
 4. Display I
 5. Stop | End

Advantages of Flowchart:

Better explanation of program:

- Flowchart is a pictorial description of any program.
- The program created by it can easily be understood.

Effective analysis:

Through the flowcharts, the program is divided into smaller parts, making effective analysis of the program.

Ques) What are the advantages and disadvantages of flowchart.

Designing of program:

Using a flowchart, your program can be designed correctly

Systematic debugging:

- When any program is created, it is always tried to make it right, but still the mistake remains somewhere.
- These mistakes can only be known when we start the execution of the program on the computer.

Ques) What are the advantages and disadvantages of flowchart.

- The mistake of this type of program, is called Bugs and the method of removing this mistake, is called Debugging.
- A flow chart is very helpful in detecting, locating and removing the mistake in a program

Ques) What are the advantages and disadvantages of flowchart.

Disadvantages of Flowchart:

Time consuming:

- Creating a flow chart takes more time.
- The flow chart is made especially for big complex problems.
- Which take more time and very hard work?

Difficult to make changes:

- If there is any modification in the flow chart the entire flowchart has to be recreated.

No standard:

- Program flowcharts, although easy to understand, but cannot be written as a standard, like others language.
- Nor with the help of flowcharts, it can be converted into the language of the program.
- Each programmer resolves the problem in its own way and creates a flow chart accordingly.

More examples of
flowchart and
algorithm.

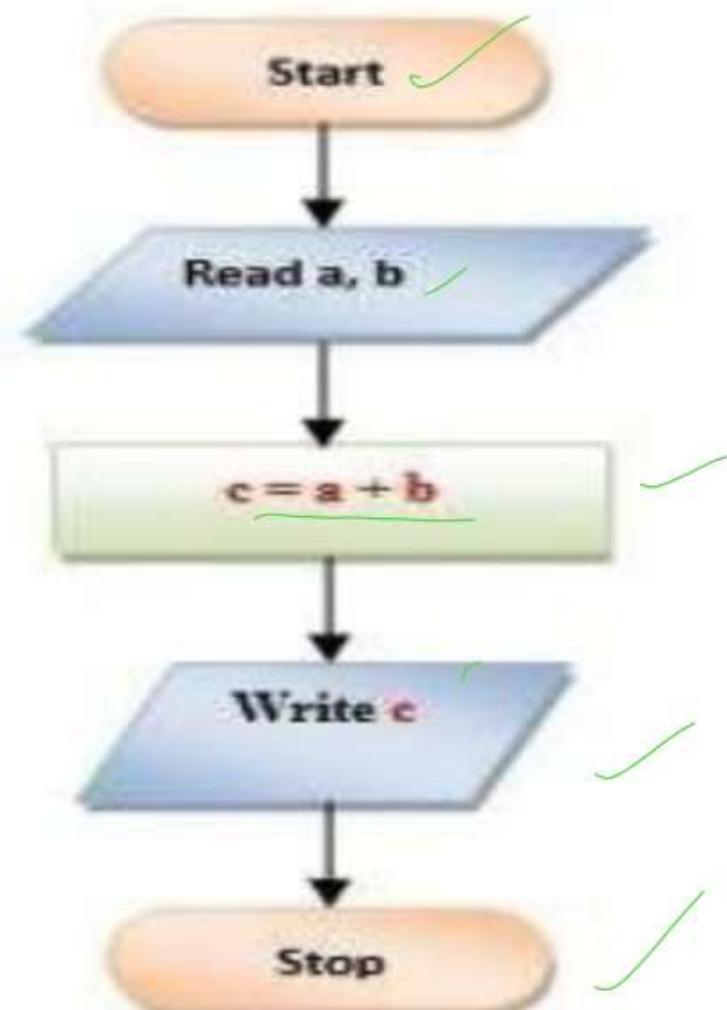


To find sum of two numbers

Algorithm

1. Start ✓
2. Read a, b ✓
3. $c = a + b$
4. Print or display c
5. Stop ✓

Flowchart



Example 1

$$\begin{aligned}a &= 2 \\b &= 3 \\c &= 2 + 3 \\c &= 5\end{aligned}$$

Input / Read
Display / Write / Print

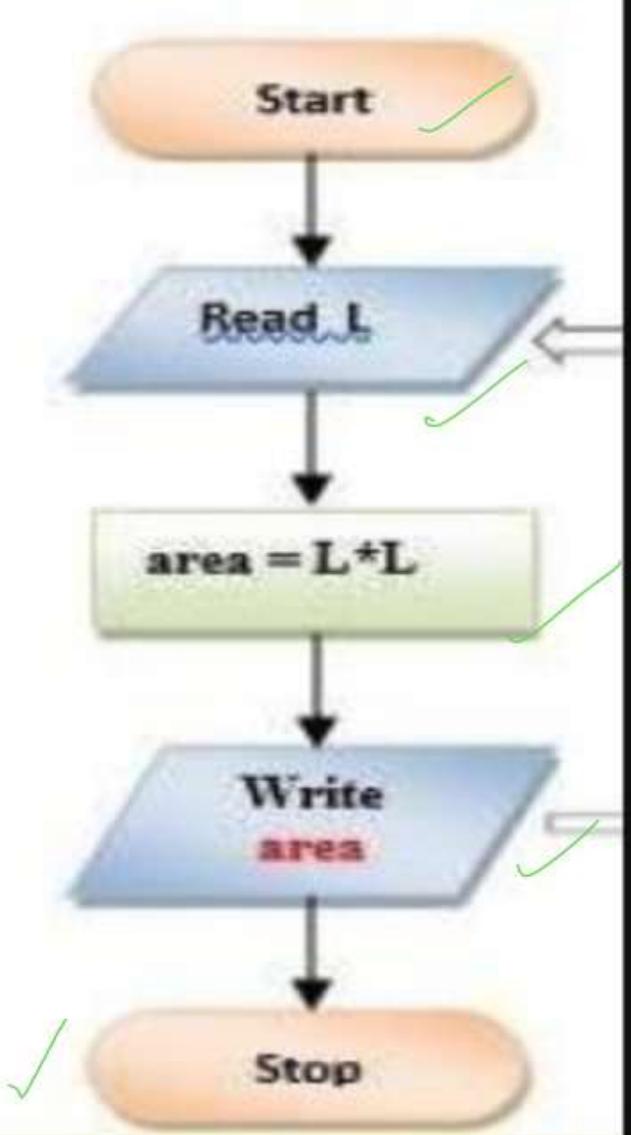
Finding Area of the square

Example 2

Algorithm

1. Start ✓
2. Read length, L ✓
3. $\text{area} = L * L$
4. Print or display area
5. Stop ✓

Flowchart



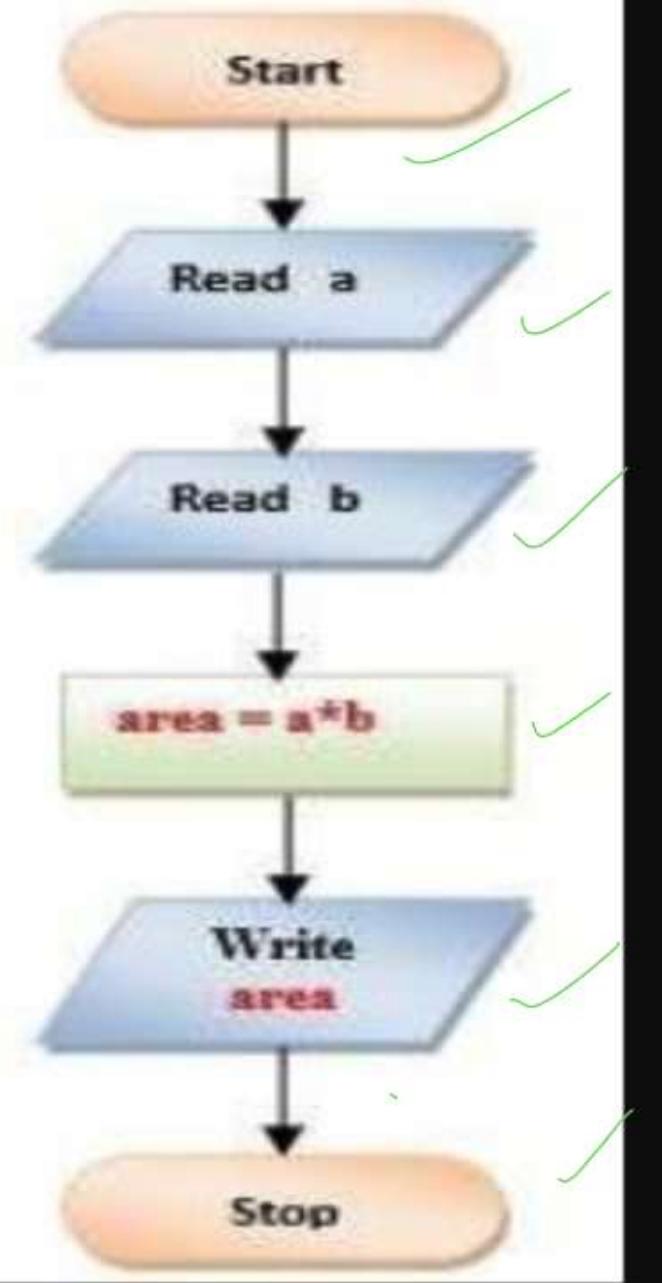
$$\begin{aligned} L &= 2 \\ \text{area} &= L * L \\ &2 * 2 = 4 \\ \boxed{\text{area} = 4} \end{aligned}$$

Finding Area of the rectangle

Algorithm

1. Start ✓
2. Read side length, a ✓
3. Read side length b ✓
4. $\text{area} = a * b$ ✓
5. Print or display area
6. Stop

Flowchart



Example 3

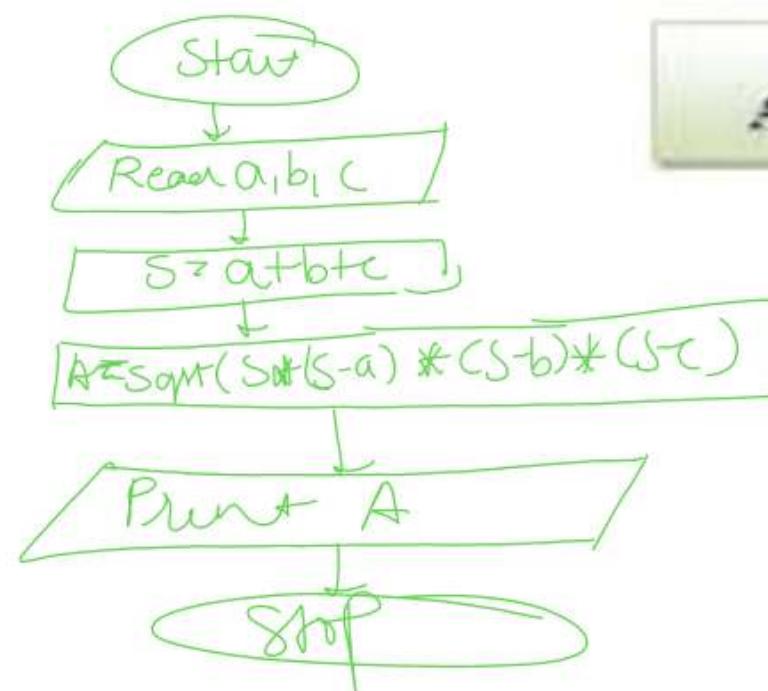
$\begin{array}{c} 3 \\ \downarrow \\ 2 \end{array}$ $\begin{array}{c} 3 \\ \downarrow \\ 2 \end{array}$
 $\text{Area} = a * b = 6$

$\begin{array}{c} a \\ \longrightarrow \\ \text{---} \\ \text{---} \\ b \end{array}$
 $\text{Area} = 6$

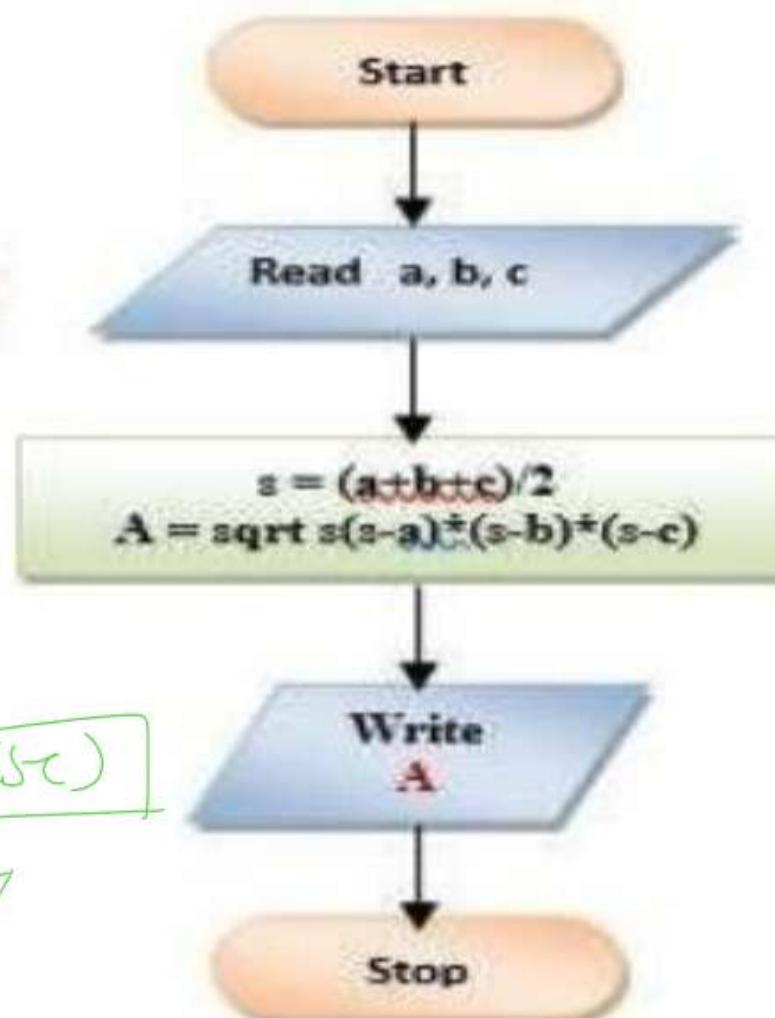
Area of a triangle where three sides are given

Algorithm

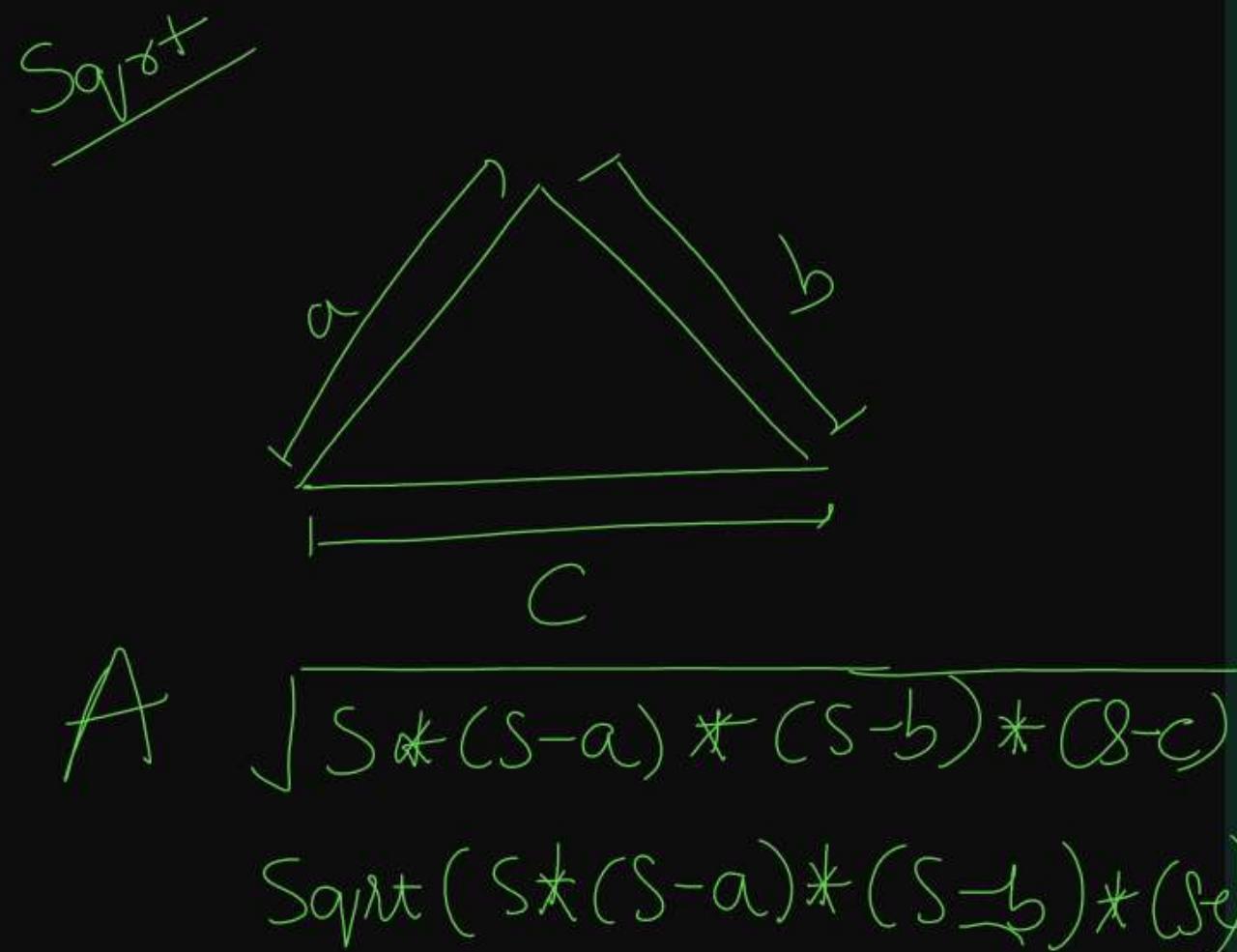
1. Start
2. Read a, b, c
3. $s = (a+b+c)/2$
4. $A = \sqrt{s * (s-a) * (s-b) * (s-c)}$
5. Print or display A
6. Stop



Flowchart



Example 4

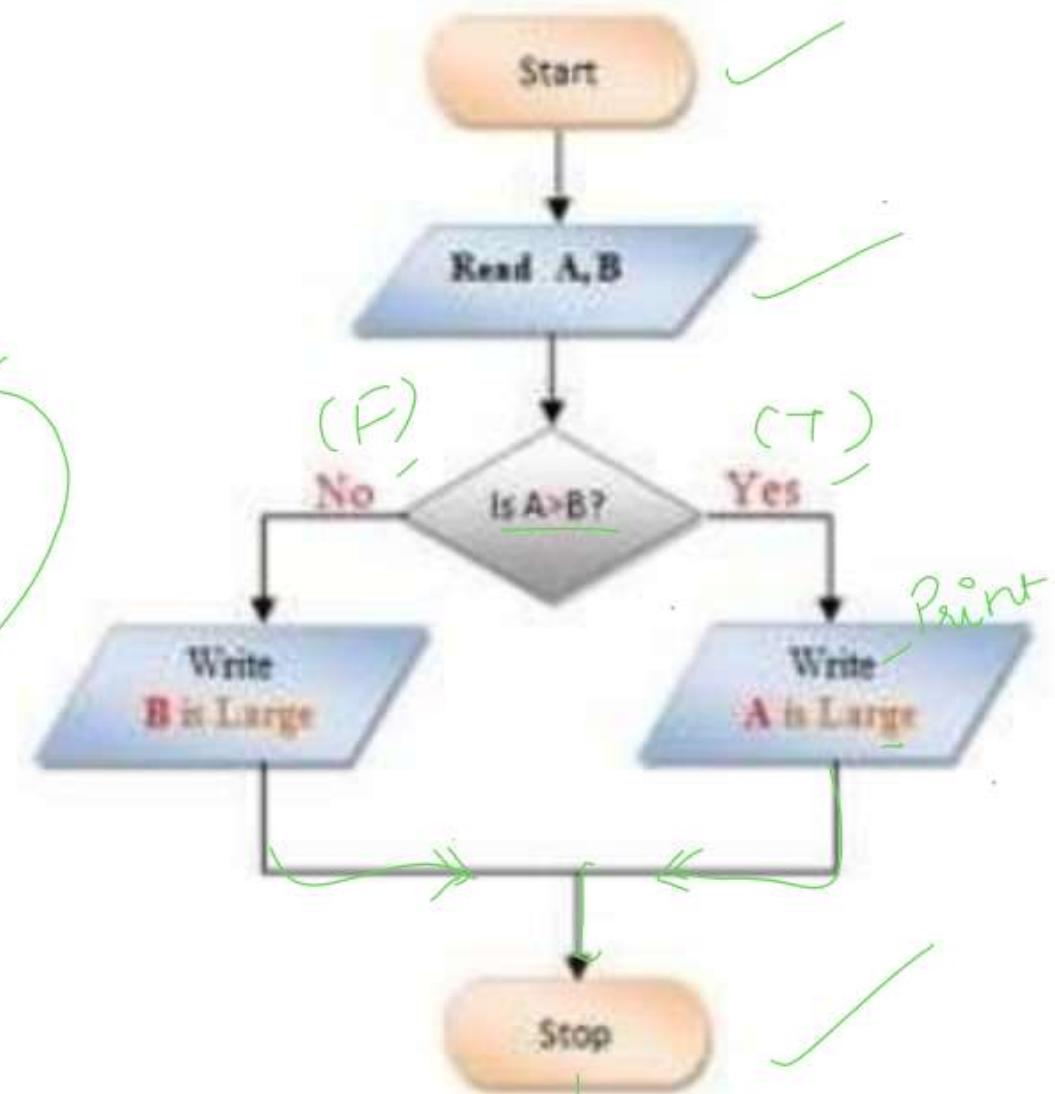


Greatest of two numbers

Algorithm

1. Start ✓
2. Read A,B ✓
3. If $A > B$ then
 - Print A is large ✓
 - else
 - Print B is large ✓
4. Stop

Flowchart



Example 5

A 2
B 3

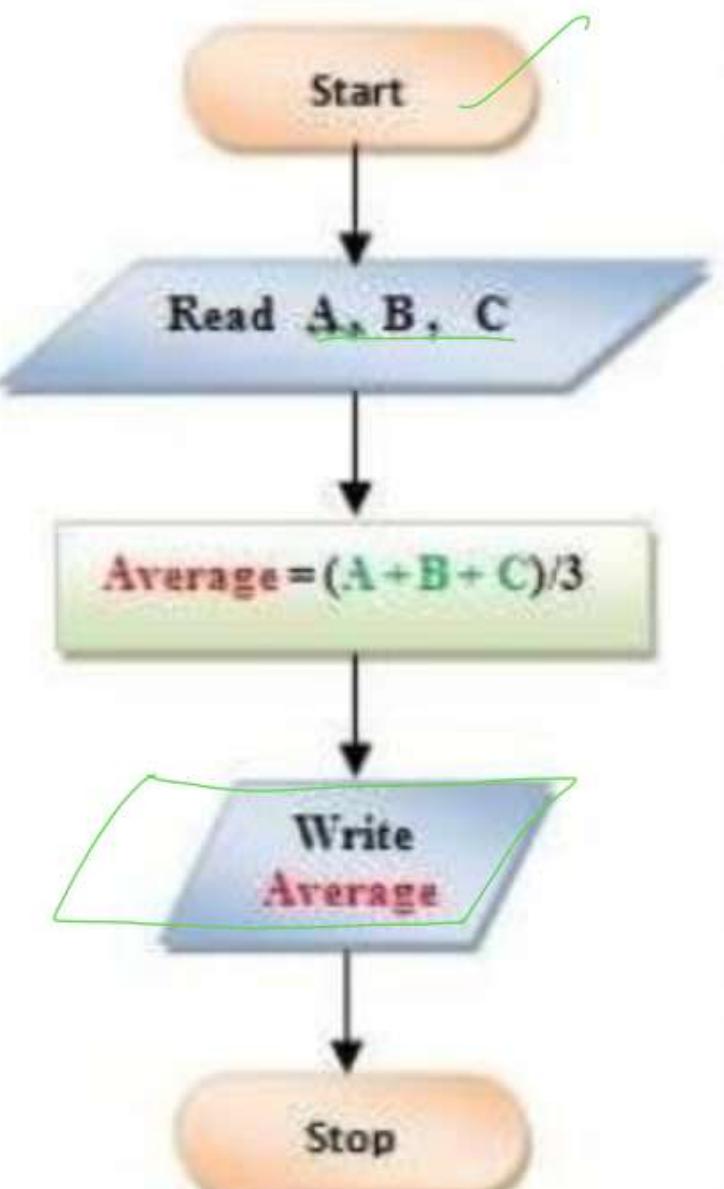
Is $A > B$?

Calculating the average for 3 numbers

Algorithm

1. Start
2. Read 3 numbers A, B, C
3. Calculate the average by the equation:
$$\text{Average} = (A + B + C) / 3$$
4. Print average
5. Stop

Flowchart



Example 6

A B C
2 2 3

$$\text{Avg} = (2 + 2 + 3) / 3$$

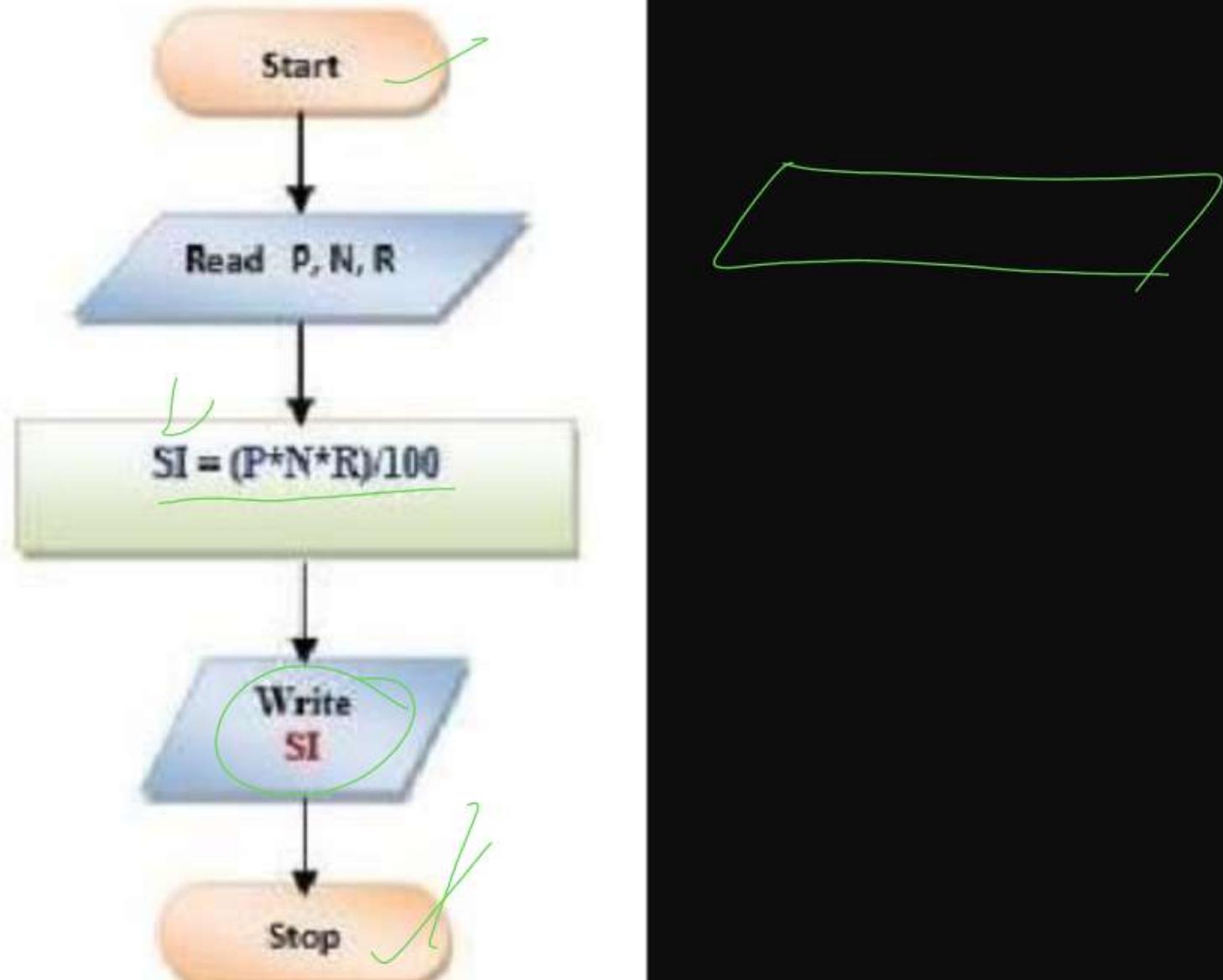
Calculate simple interest using the expression (SI=PNR/100)

Example 7

Algorithm

1. Start
2. Read P, N, R
3. $SI = (PNR)/100$
4. Print SI
5. Stop

Flowchart



Ques) What do you mean by pseudo code.(2021-22)

- Pseudo code is a simple way of writing programming code in English.
- Pseudo code is not actual programming language.
- It uses short phrases to write code for programs before you actually create it in a specific language.

- Once you know what the program is about and how it will function, then you can use pseudo code to create statements to achieve the required results for your program.
- Algorithms are written in Pseudo code.

1. main()
2. {
3. int L, B, peri;
4. scan (L and B);
5. peri=2*(L+B)
6. print peri;
7. }

```
#include <stdio.h>
void main()
{
    int l, B, Peri,
    Scanf("%d%d", &l, &B),
    Peri=2*(l+B),
    Trunf("%d", Peri),
}
```

Download **Gateway Classes** Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : BASIC OF C PROGRAMMING

Lec-8

Today's Target

- ✓ What is c?
- ✓ Why c?
- ✓ Feature of C
- ✓ Basic structure of C
- ✓ First hello world program

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

- C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972.
- It is a very popular language, despite being old.
- It was developed to write the UNIX operating system.

- It is one of the most popular programming language in the world
- If you know C, you will have no problem learning other popular programming languages such as Java, Python, C++, C#, etc, as the syntax is similar
- C is very fast, compared to other programming languages, like Java and Python
- C is very versatile; it can be used in both applications and technologies

Ques) Features of C language ?

1. Middle-Level Language: As it is a middle-level language so it has the combined form of both capabilities of assembly language and features of the high-level language.

2. Portability: C language is portable as programs that are written in C language can run and compile on any system with either none or small changes.

3. Rich set of built-in Operators: It is a diversified language with a rich set of built-in operators which are used in writing complex or simplified C programs

4. Procedural Language: In a procedural language like C step by step predefined instructions are carried out

$a + b$
↓
operator

STRUCTURE OF C PROGRAM

(22-23)

```
/* Documentation section */  
/* Link section */  
/* Definition section */  
/* Global declaration section */  
main()  
{  
    /* Declaration part */  
    /* Executable part (statements) */  
}  
/* Sub-program section */
```

/* Program to
Calculate the
area of Square */

/* Comments */

// Hi

/* This is program */

for calculate */

➤ The documentation section: is used for displaying any information about the program like the purpose of the program etc, and this section should be enclosed within comment lines. The statements in the documentation section are ignored by the compiler.

➤ The link section: consists of the inclusion of header files.

➤ The definition section: consists of macro definitions, defining constants etc.

➤ Global declaration: Anything declared in the global declaration section is accessible throughout the program, i.e. accessible to all the functions in the program.

```
//Hello /*  
() */
```

()

- **main ()**: function is mandatory for any program and it includes two parts, the declaration part and the executable part.
- **Sub-program section**: is optional and used when we require including user defined functions in the program.

$$\begin{aligned} \textcircled{Q} &= 2 \\ \text{Area} &- Q \times a \\ \text{Area} &- Pr \end{aligned}$$

First c program of hello world

```
#include <stdio.h>
void main()
{
    printf("Hello World");
}
```

Online Compiler

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("Hello World");
    getch();
}
```

Turbo C++

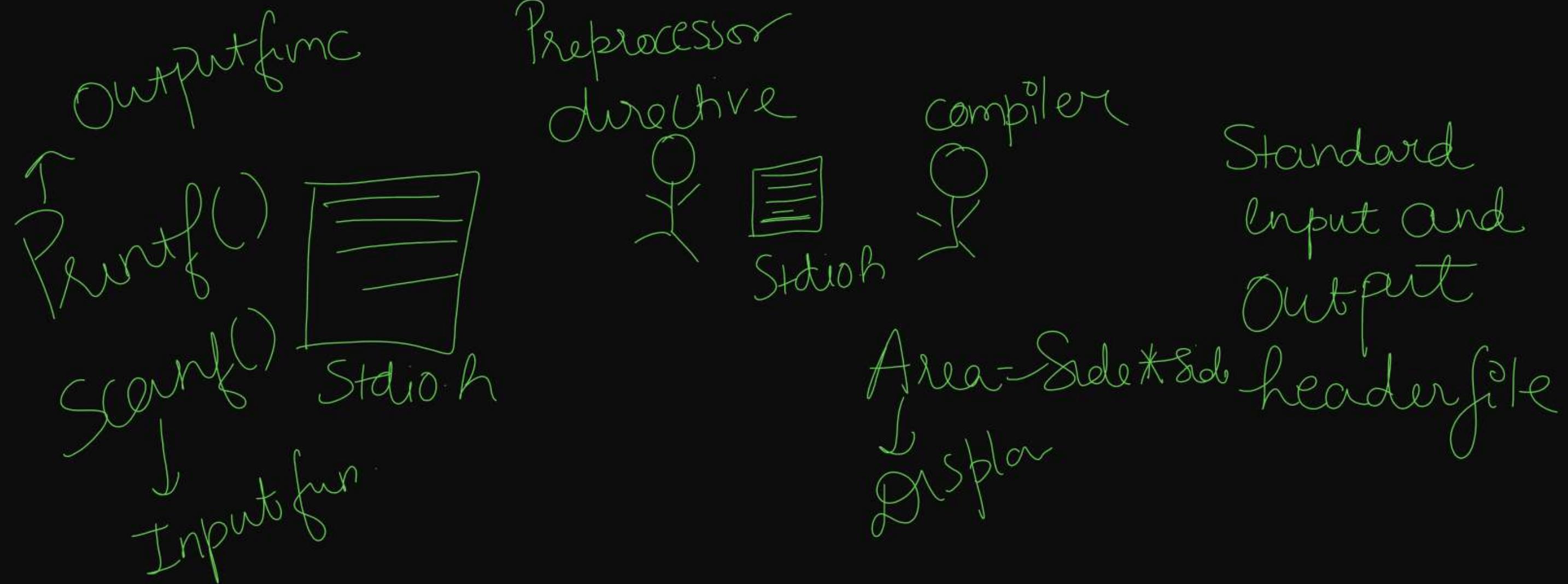


```
#include<stdio.h>
```

#include is a pre-processor directive. It is not really part of our program, but instead it is an instruction to the compiler to make it do something. It tells the C preprocessor to include the contents of a file (in this case system file called stdio.h).

#include<stdio.h>

↑ hash
↑ Preprocessor directive
↑ header file



void

➤ This means that this means nothing. In this case, it is referring to the function whose name follows.

➤ Void tells to compiler that a given entity has no meaning, and produces no error.

main()

➤ In this particular example, the only function in the program is called main.

➤ C regards the name main as a special case and will run this function first i.e. the program execution starts from main.

{ } (Brace)

- As the name implies, braces come in packs of two - for every open brace there must be a matching close one.
- Braces allow us to group pieces of program together, often called a block.

; (semicolon)

- It tells the compiler where a given statement ends.
- If the compiler does not find one of these characters where it expects to see one, then it will produce an error.

Printf()

➤ This function is used to display text, constant or value of variable on screen in specified format.

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

#include is a pre-processor directive. It is not really part of our program, but instead it is an instruction to the compiler to make it do something.

conio.h:It stands for console input-output.

clrscr() This function is used to clear the previous output

getch() :pauses the Output Console until a key is pressed


```
1 #include<stdio.h>
2 void main()
3 {
4
5     printf("Hello World\n");
6     printf("bye");
7
8
9 }
10
11 |
```

```
▼ ▶ ⌂
Hello World
bye

...Program finished with exit code 3
Press ENTER to exit console. □
```

Download **Gateway Classes** Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : BASIC OF C PROGRAMMING

Lec-9

Today's Target

- ✓ Error
- ✓ Component of C language

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

Types of Errors (2 Marks)

Syntax error: The errors which arises due to violation of any rule or regulation of C language
during the development of program those errors are known as Syntax error. These errors are
these errors are identified by a compiler during compilation. Example: missing of semicolon.

Printf("Hello");
Printf("Hi");
Printf(Hi);

Runtime error: The error that occurs during execution of program those errors are termed as runtime error.

Example: incorrect header file name.

```
Print("Hello");
#ifndef include<stdio.h>
#ifndef include<Stalib.h>
```

Logical error: The errors which are introduced due to usage of wrong expression formula for logic in program these errors are called as logical error for example using $2 * 3.14$ as area of circle.

$$\pi r^2$$

~~$3.14 * r * r$~~

$$\text{Area} = \text{Side} * \text{Side}$$

Square

$$\text{Area} = \text{Side} * \text{Side} * \text{Side}$$

Input data error: The error which occurs during the data entry process because of entering wrong input values after the execution of program.

2,3,4,1,-6,-1
2.3,4.5
A B C

The four main components of C language are

- 1) The Character Set.
- 2) Tokens (2 Marks)
- 3) Variables
- 4) Data Types (7 Marks)

1) The Character Set: Character set is a set of valid characters that a language can recognize. A character represents any letter, digit or any other sign.



Types	Character Set
Lower case	a-z
Upper case	A-Z
Digits	0-9
Special Character	!@#\$%^&* +,-,/
White space	Tab or new lines or space

(2Marks)

Tokens: The smallest individual unit in a program is known as a token .C has five tokens:

- i. Keywords
- ii. Identifiers
- iii. Constants
- iv. Strings
- v. Operators
- vi. Special Symbols

i) Keywords: Keywords are reserved word in C. They have predefined meaning cannot changed. All keywords must be

written in lowercase. Eg:- auto , long ,char ,short etc . There are total 32 keywords in C.

ii) Identifiers: - Identifiers refer to the names of variable, functions and arrays. These are user-defined names. An identifier in C can be made up of letters, digits and underscore.

Identifiers may start with either alphabets or underscore. Two rules

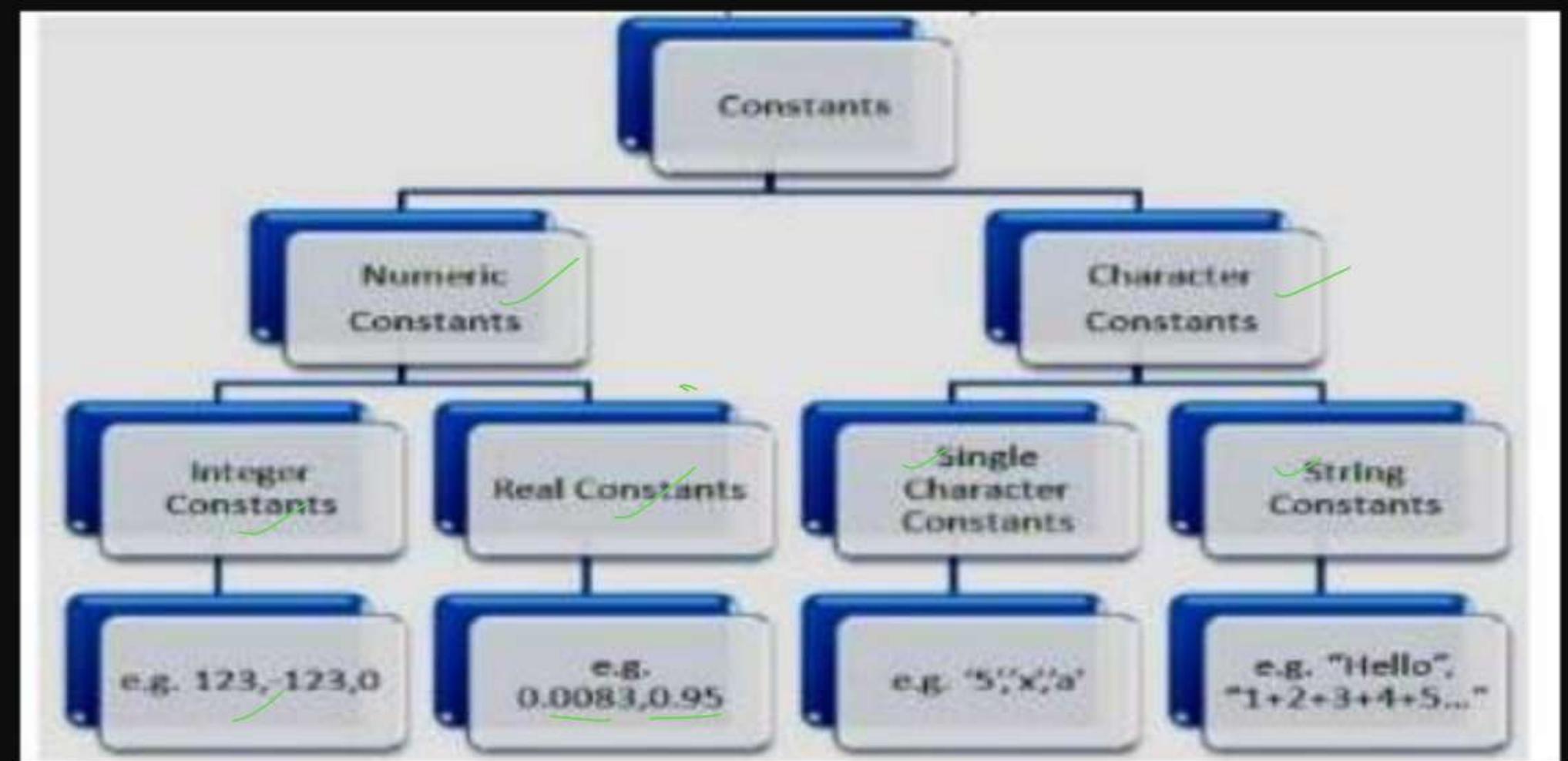
must be kept in mind when naming identifiers.

① $a=6X$

$a=6$ $a=5$
 $a1=6$
 $a=7$
 $A=9$

$b=6$

iii) Constants: - Constants in C refers to fixed values that do not change during the execution of a program. C supports several types of constants.



‘\$’ ‘\a’ ‘\A\B\c’
quotes

"Hello"

“1+2+3+4”
+ + + +

(3 units)

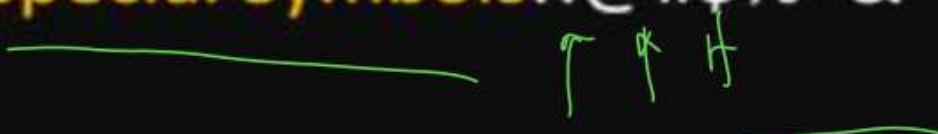
iv) String:- It may consist of any combination of digits, letters, escaped sequences and spaces enclosed in double quotes. Every string constant ends up with a NULL character which is automatically assigned (before the closing double quotation mark) by the compiler.

v) Operators:-

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. C language is rich in built-in operators and provides the following types of operators:

+ * / - a + b

)

- Arithmetic Operators (+, -, *, /, %)
 - Relational Operators (<, >)
 - Logical Operators (&, ||)
 - Bitwise Operators (<<, >>, & |)
 - Assignment Operators (=)
 - Increment and decrement operators (++, --)
 - Conditional operators (?)
- vi) Special Symbols: ! @ # \$ % ^ & *
- 

C Language Components

Variables:

A variable is an object or element that may take on any value or a specified type. Variable are nothing but identifiers, which are used to identify variables programming elements.

RULES FOR A VARIABLE NAME

- Names can contain letters, digits and underscores
- Names must begin with a letter or an underscore (_)
- Names are case sensitive (myVar and myvar are different variables)
- Names cannot contain whitespaces or special characters like !, #, %, etc.
- Reserved words cannot be used as names.

hy
he

$a = 6$
 $a = 7.5$
 $a = 'A'$

$\overline{=}$ Assignment
operator

A-Z 0-9
a-z _
A-Z - a9
a-z q x

Space
are
not
allowed
A BX
A\$B X
↑ Special
Character

ab valid ✓
_ab valid ✓
A30 valid
Invalid
1var ✗
Var1 valid
a b invalid

—L ✓

aL
a₁b ✗
white Space

Data types indicate the types of data a variable can have. A data types usually define a set of values, which can be stored in the variable along with the operations that may be performed on those values. C includes two types of data.

Primary or Primitive data type: These are fundamental data types in C.

Int , char, float, double

Derived Data Types: Derived data types are constructed from the simple data types and other derived data types . Derived data include arrays, functions, pointers, structures, unions and enumerations

Download Gateway Classes Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : BASIC OF C PROGRAMMING

Lec-10

Today's Target

- ✓ Format specifier
- ✓ Syntax of printf() and scanf()
- ✓ Basic program

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

FORMAT SPECIFIERS

- Format specifiers in C are used to take inputs and print the output of a type.
- The symbol we use in every format specifier is %.
- Format specifiers tell the compiler about the type of data that must be given or input and the type of data that must be printed on the screen.

int	%d
char	%C
float	%f
double	%lf

- Syntax of printf()

`Printf("format specifier", variablename);`

Syntax of scanf()

```
Scanf("format specifier", &variablename);  
          ↑  
          address
```

```
Printf("M=%d\nZ=%d",M,Z);
```

Printf("Hello world");
Variable print
Int a=20;
Printf("%d", a); → 20
Int c=10, z=20;
Printf("%d %d", c, z);
→ Int m=10, z1=7;

Download Gateway Classes Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : BASIC OF C PROGRAMMING

Lec-11**Today's Target**

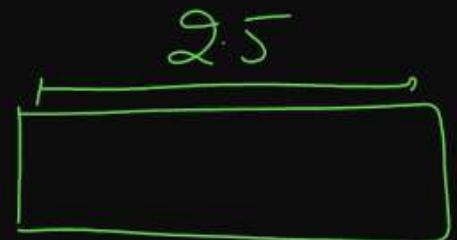
- ✓ Basic program of-
- Area of rectangle and square ✓
- Perimeter of rectangle and square ✓
- Swapping of two number using third variable ✓
- Swapping of two number without using third variable ✓
- Area and perimeter of circle ✓
- Simple interest
- Fahrenheit to Celsius

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

Write a program to print area of square

Area of Square = Side * Side



Scarf()
Printf()

Area, Side
↓
%f


```
1 #include<stdio.h>
2 void main()
3 {
4     float side,area;
5     printf("enter the area of square\n");
6     scanf("%f",&side);
7     area=side*side;
8     printf("the area of square=%f",area);
9 }
10
```

VS
GDB



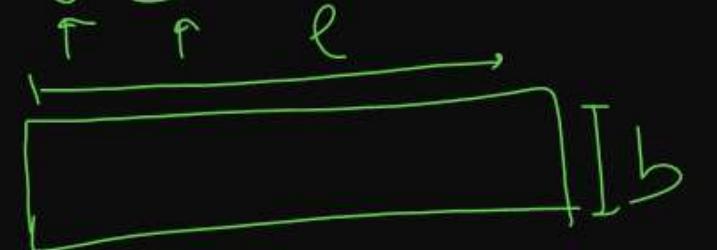
input

```
enter the area of square
5.6
the area of square=31.359999
```

```
...Program finished with exit code 0
Press ENTER to exit console.[]
```

Write a program to print area of rectangle

Area = $l * b$



Scans()

$\%f$

Printf()

2.0
2.1
6.5

main.c

```
1 #include<stdio.h>
2 void main()
3 {
4     float l,b,area;
5     printf("enter the length and breadth of rectangle\n");
6     scanf("%f %f",&l,&b);
7     area=l*b;
8     printf("area of rectangle=%f",area);
9 }
10
11
```

input

```
enter the length and breadth of rectangle
3.4
5.6
area of rectangle=19.040001
...Program finished with exit code 0
Press ENTER to exit console.□
```

Write a program to print perimeter of rectangle

$$\text{Perimeter} = 2 * (l + b)$$

Scanf() → %f %f
Printf() = %f

Write a program to print perimeter of rectangle

main.c

```
1 #include<stdio.h>
2 void main()
3 {
4     float l,b,peri;
5     printf("enter the length and breadth of rectangle\n");
6     scanf("%f %f",&l,&b);
7     peri=2*(l+b);
8     printf("the perimeter of rectangle=%f",peri);
9 }
10
11
12
```

```
enter the length and breadth of rectangle
4.6
4.7
the perimeter of rectangle=18.599998
...Program finished with exit code 0
Press ENTER to exit console.
```

input

Write a program to print perimeter of square

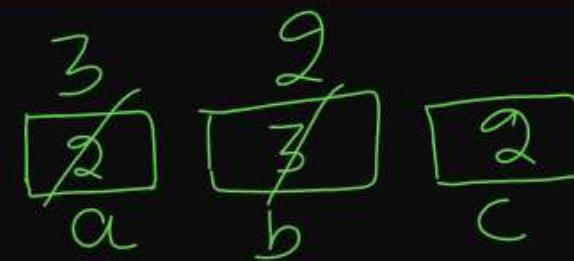
```
Peri = 4*Side    float
      Scanf()    %f
      Printf()   %f
```


Write a program to print perimeter of square

```
1 #include<stdio.h>
2 void main()
3 {
4     float side, peri;
5     printf("enter the side of square\n");
6     scanf("%f", &side);
7     peri=4*side;
8     printf("the perimeter of square=%f",peri);
9 }
10
11
```

```
enter the side of square
6.7
the perimeter of square=26.799999
...Program finished with exit code 0
Press ENTER to exit console. □
```

Write a program of swapping of two number using third variable



Scarf() $\left. \begin{array}{l} c = a; \\ a = b; \\ b = c; \end{array} \right\} \text{logic}$
 %d %d

Printf() %d %d

main.c

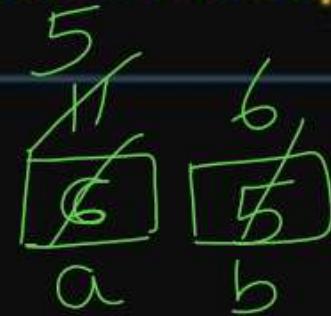
```
1 #include<stdio.h>
2 void main()
3 {
4     int a,b,c;
5     printf("enter the value of a and b\n");
6     scanf("%d %d",&a,&b);
7     printf("the value of a=%d and b=%d before swapping\n",a,b);
8     c=a;
9     a=b; { logic
10    b=c;
11    printf("the value of a=%d and b=%d after swapping",a,b);
12 }
13
14
15
```

```
enter the value of a and b
3
4
the value of a=3 and b=4 before swapping
the value of a=4 and b=3 after swapping
...Program finished with exit code 0
Press ENTER to exit console.[]
```

input

Activate Windows
Go to Settings to activate Windows.

Write a program of swapping of two number without using third variable



$$\begin{aligned} a = a + b, & \rightarrow 6 + 5 = 11 \\ b = a - b, & \rightarrow 11 - 5 = 6 \\ a = a - b, & \rightarrow 11 - 6 = 5 \end{aligned}$$

```
2 → int %d %d → Scanf()
printf("%d%d", a, b),
```



```
1 #include<stdio.h>
2 void main()
3 {
4     int a,b;
5     printf("enter the value of a and b\n");
6     scanf("%d %d",&a,&b);
7     printf("the value of a=%d and b=%d before swapping\n",a,b);
8     a=a+b;
9     b=a-b; { log
10    a=a-b;
11    printf("the value of a=%d and b=%d after swapping",a,b);
12 }
```

```
enter the value of a and b
2
3
the value of a=2 and b=3 before swapping
the value of a=3 and b=2 after swapping
...Program finished with exit code 0
Press ENTER to exit console.[]
```

Write a program to print area of circle

$$\text{Area} = \pi r^2 = 3.14 * r * r$$

↑
float
↑
 $\%f \leftarrow 1 \leftarrow \text{scanf}()$

Printf()
↑
 $\%f$

Variables
float $\leftarrow r, \text{Area}$

Write a program to print area of circle

main.c

```
1 #include<stdio.h>
2 void main()
3 {
4     float radius, area;
5     printf("enter the radius of circle\n");
6     scanf("%f",&radius);
7     area=3.14*radius*radius;
8     printf("the area of circle=%f",area);
9 }
```

input

```
enter the radius of circle
```

```
5.1
the area of circle=81.671394
```

```
...Program finished with exit code 0
Press ENTER to exit console. □
```

Write a program to print perimeter of circle

$$\text{Perimeter} = 2\pi r$$

$$2 * 3.14 * r$$

float Peri, r;

printf()

%d

float
Scanf()

%f

2.5

2.0

Write a program to print perimeter of circle

```
main.c
1 #include<stdio.h>
2 void main()
3 {
4     float radius, peri;
5     printf("enter the radius of circle\n");
6     scanf("%f", &radius);
7     peri=2*3.14*radius;
8     printf("the perimeter of circle=%f",peri);
9 }
10
11
```

enter the radius of circle
4.5
the perimeter of circle=28.260000
... Program finished with exit code 0
Press ENTER to exit console.

$$\begin{array}{l} 4.5 \\ \times 28.26 \\ \hline 28.26 \end{array}$$

$2 \times 3.14 \times 4.5$

Write a program to print simple interest

$SI = (P * R * T) / 100$

float P, R, T, SI;

scanf() → %f %f %f

Prentf()

↓

%f

Write a program to print simple interest

main.c

```
1 #include<stdio.h>
2 void main()
3 {
4     float p,r,t,si;
5     printf("enter principal, rate, time respectively\n");
6     scanf("%f %f %f",&p,&r,&t);
7     si=(p*r*t)/100;
8     printf("the simple interest=%f",si);
9 }
```

```
enter principal, rate, time respectively
300.45
4.6
2.5
the simple interest=34.551754
... Program finished with exit code 0
Press ENTER to exit console.□
```

input

Write a program to convert Fahrenheit to Celsius

$$C = (F - 32) * \frac{5}{9}$$

↓ ↓
float float
Printf() %f → Scanf()
%f

295
63.5

5/9

0.5555555555555555

9/5

0.1111111111111111

Write a program to convert Fahrenheit to Celsius

main.c

```
1 #include<stdio.h>
2 void main()
3 {
4     float f,c;
5     printf("enter the temperature in fahrenheit\n");
6     scanf("%f",&f);
7     c=(f-32)*(5.0/9.0);
8     printf("the temperature in celsius=%f",c);
9 }
```

```
enter the temperature in fahrenheit
212
the temperature in celsius=100.000000
...Program finished with exit code 0
Press ENTER to exit console.
```

Int a
2.6 - C
2
212.0

2 → 20
2.0
212.0

Download Gateway Classes Application
From Google Play store

Link in video Description

Thank You

UNIT-1 : BASIC OF C PROGRAMMING

Lec-12**Today's Target**

- Data types table ✓
- Escape sequence ✓
- C program compilation and execution ✓

By Pragya Rajvanshi

B.Tech M.tech (C.S.E.)

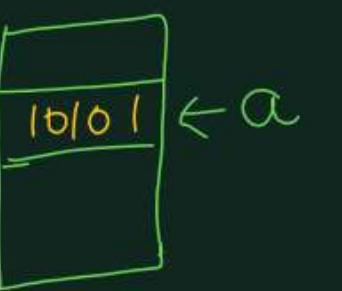
2.6

7 Marks

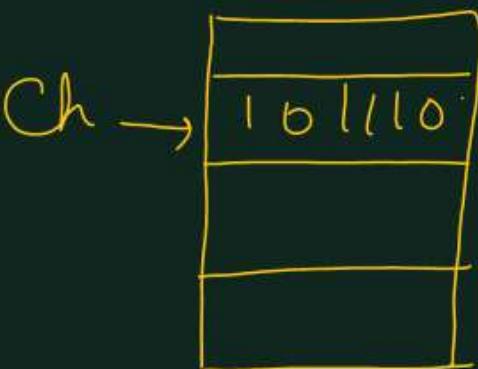
Data Type	Size in Bytes	Range	Format
signed char	1	-128 to 127 (2^7 to $2^7 - 1$)	%c
unsigned char	1	0 to 255 0 to $2^8 - 1$	%c
short signed int	2	-32768 to -32767 (-2^{15} to $2^{15} - 1$)	%d
short unsigned int	2	0 to 65535 0 to $2^{16} - 1$	%u
signed int	2	-32768 to -32767 - 2^{15} to $2^{15} - 1$	%d
unsigned int	2	0 to 65535 0 to $2^{16} - 1$	%u
long signed int	4	-2147483648 to +2147483647 - 2^{31} to $2^{31} - 1$	%ld
long unsigned int	4	0 to 4294967295 0 to $2^{32} - 1$	%lu
Float	4	3.4e-38 to 3.4e+38	%f
Double	8	1.7e-308 to 1.7e+308	%lf
long double	10	3.4e-4932 to 1.1e+4932	%Lf

int a=10;

—
2⁴ 2³ 2² 2¹ 2⁰
16 8 4 2 1
0 0 0 0 0 0
0 0 0 0 1 1
0 0 0 1 0 2
0 0 0 1 1 3
0 1 0 1 0 10
1 0 1 0 1 18



Char ch='A';



ASCII table

A - 65

Z - 90

a - 97

z - 122

Character $\xrightarrow[\text{from}]{\text{ASCII}}$ 65 \rightarrow Binary Value

Char ch='b', ~ 98

What is escape sequence character in c (2014-15,2017-18)

Esc. Seq.	Purpose	Esc. Seq.	Purpose
\n	New line	\t	Tab
\b	Backspace	\r	Carriage return
\f	Form feed	\a	Alert
\'	Single quote	\"	Double quote
\\\	Backslash		

2 Mark
7 Marks
Q
(1)
(2)
(3)

hello"hi
hello'hi

```
1 #include<stdio.h>
2 void main()
3 {
4     printf("hello");
5     printf("hi");
6 }
7
```

hellohi

...Program finished with exit code 2
Press ENTER to exit console.

```
1 #include<stdio.h>
2 void main()
3 {
4     printf("hello\n");
5     printf("hi");
6 }
7
```

hello
hi

...Program finished with exit code 2
Press ENTER to exit console.

```
1 #include<stdio.h>
2 void main()
3 {
4     printf("hello\t");
5     printf("hi");
6 }
7
```

```
▼ ↵ ⌂
hello hi
```

```
...Program finished with exit code 2
Press ENTER to exit console. □
```

```
1 #include<stdio.h>
2 void main()
3 {
4     printf("hello\"");
5     printf("hi");
6 }
7
```

hello"hi

...Program finished with exit code 2
Press ENTER to exit console. []

"\\ hello\"

"hello"

```
1 #include<stdio.h>
2 void main()
3 {
4     printf("hello\"");
5     printf("hi");
6 }
7
```

hello"hi

...Program finished with exit code 2
Press ENTER to exit console.

```
1 #include<stdio.h>
2 void main()
3 {
4     printf("surya");
5     printf("\n");
6     printf("surya\bdev");
7     printf("\n");
8     printf("surya\b\bdev");
9
10
11
12 }
```

```
surya
suryadev
surdev
```

```
... Program finished with exit code 10
Press ENTER to exit console.
```

Suryadev

Suryadev

Suryadev

Suryadev

Surdev

```
1 #include<stdio.h>
2 void main()
3 {
4 printf("surya\rdev");
5 }
6
```

devya

...Program finished with exit code 9
Press ENTER to exit console. █

devya
Surya
xyz (xyz\rw)

xyz. xyz

```
1 #include<stdio.h>
2 void main()
3 {
4 printf("surya\b");
5 }
6
```

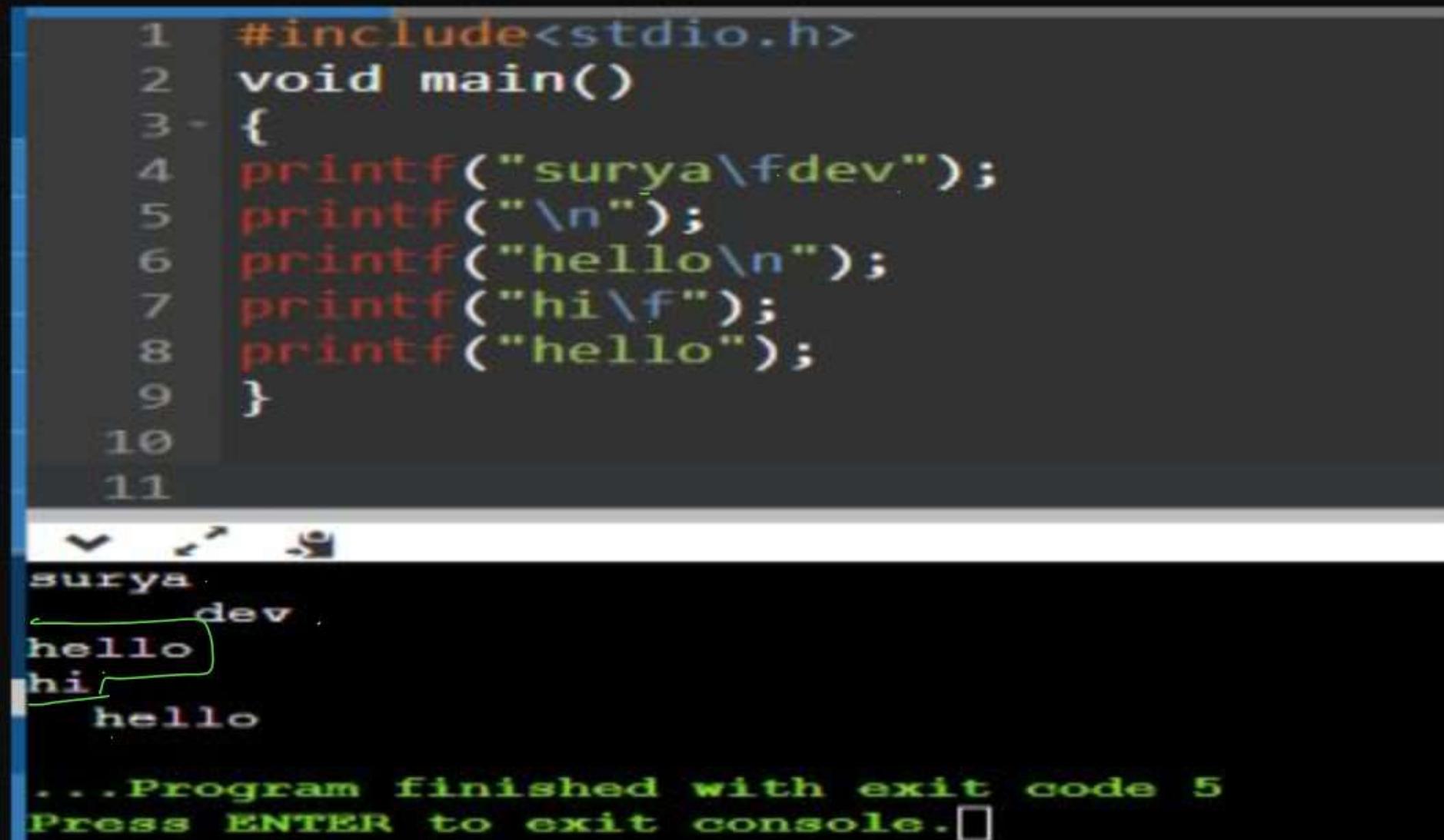


surya

...Program finished with exit code 6
Press ENTER to exit console. █

Write a program to print perimeter of circle

```
1 #include<stdio.h>
2 void main()
3 {
4 printf("surya\fdev");
5 printf("\n");
6 printf("hello\n");
7 printf("hi\f");
8 printf("hello");
9 }
10
11
```



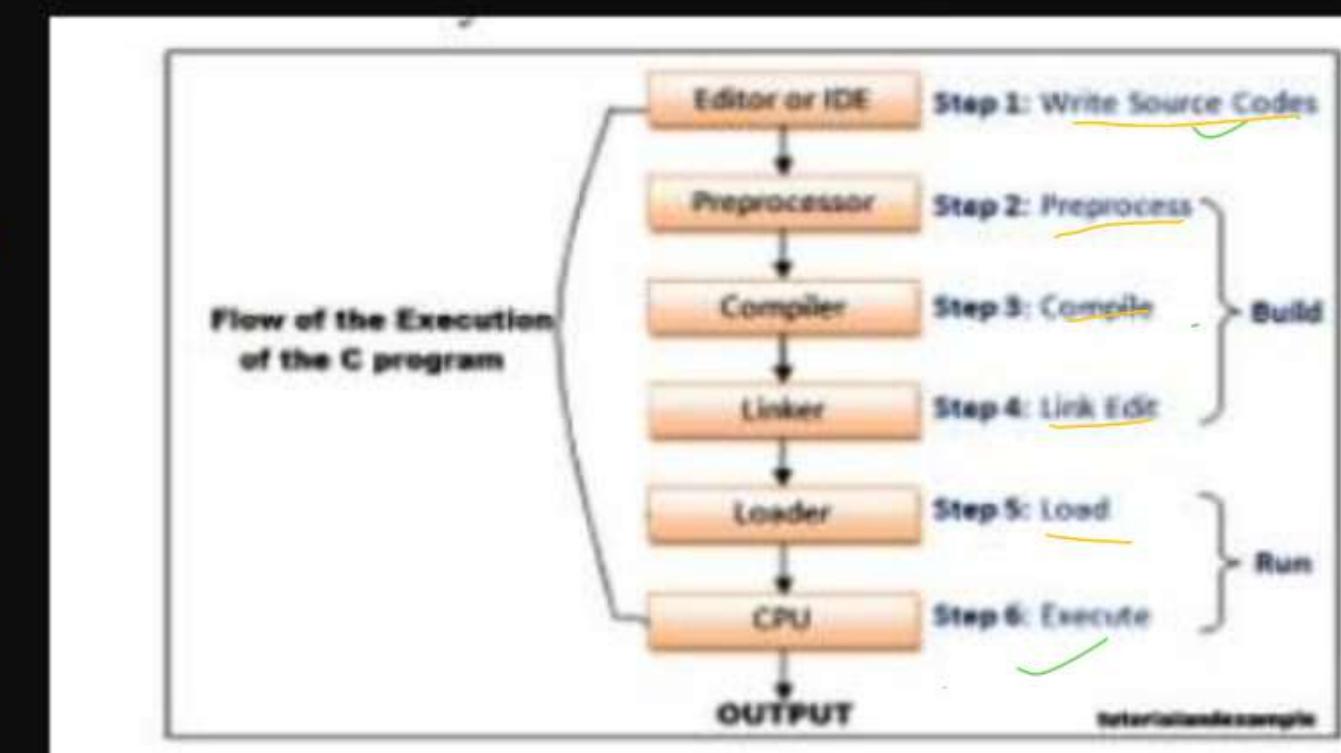
```
surya
  dev
hello
hi
hello

... Program finished with exit code 5
Press ENTER to exit console.□
```

Compilation and execution of program

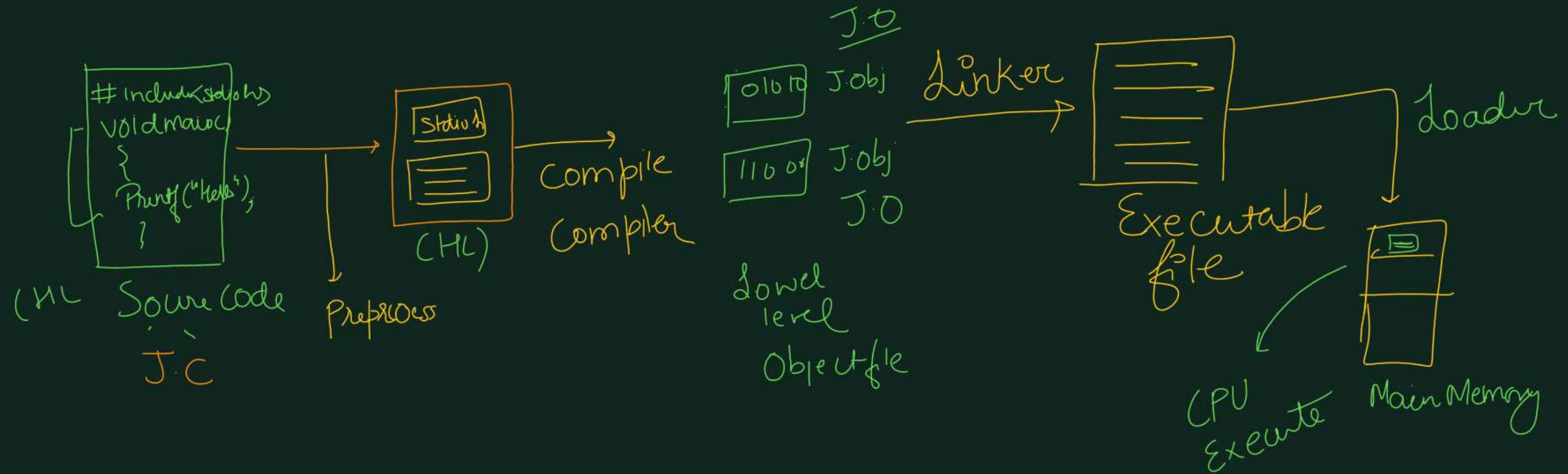
Source File- This file contains the source code of the program. The file extension of any c file is .c. The file contains C source code that defines the main function & maybe other functions.

Compiler
Hl → Lw
(Object Code)



.c
#include<Stdio.h>

.pdf
.pptx



Header File- A header file is a file with extension .h which contains the C function declarations and macro definitions and to be shared between several source files

Object File- An object file is a file containing object code, with an extension .o, meaning relocatable format machine code that is usually not directly executable. Object files are produced by an assembler, compiler, or other language translator, and used as input to the linker, which in turn typically generates an executable or library by combining parts of object files.

Executable File- The binary executable file is generated by the linker. The linker links the various object files to produce a binary file that can be directly executed.

Download Gateway Classes Application
From Google Play store

Link in video Description

Thank You