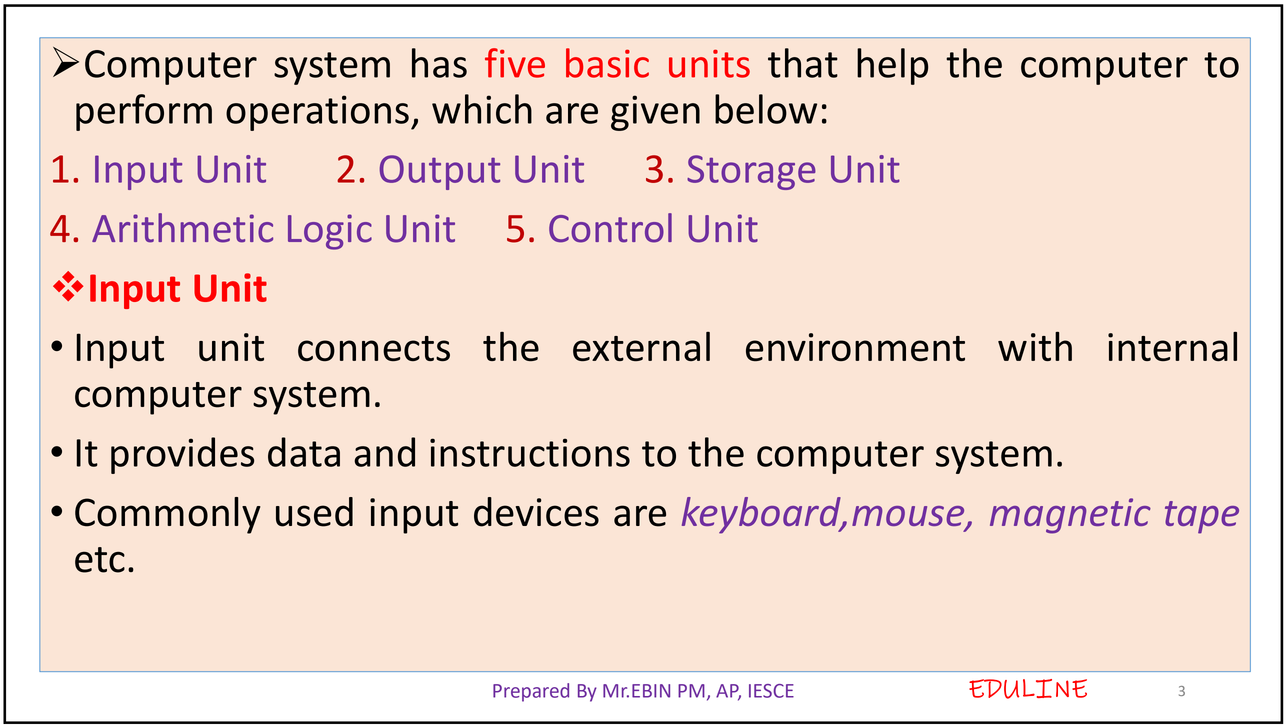
MODULE 1

BASICS OF COMPUTER HARDWARE & SOFTWARE

CO - Students will be able to identify hardware & software parts of a computer system.

BASICS OF COMPUTER ARCHITECTURE

- ➢Computer is an electronic machine that makes performing any task very easy.
- ➢In computer, the CPU executes each instruction provided to it, in a series of steps, this series of steps is called **Machine Cycle**, and is repeated for each instruction.
- ➢ One machine cycle involves
  - ***Fetching of instruction***
  - ***Decoding the instruction***
  - ***Operand fetching***
  - ***Executing the instruction***



➢Computer system has five basic units that help the computer to perform operations, which are given below:

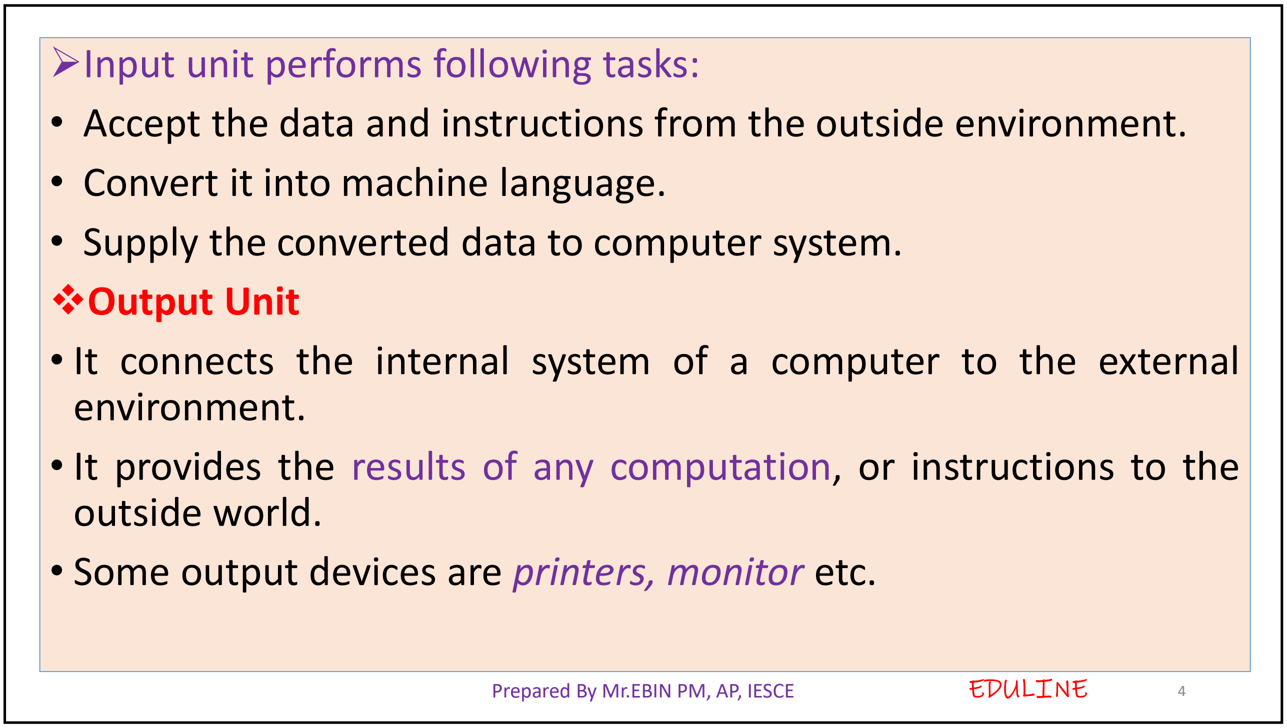
1. Input Unit 2. Output Unit 3. Storage Unit

4. Arithmetic Logic Unit 5. Control Unit

❖**Input Unit**

- Input unit connects the external environment with internal computer system.
- It provides data and instructions to the computer system.
- Commonly used input devices are *keyboard,mouse, magnetic tape* etc.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 3



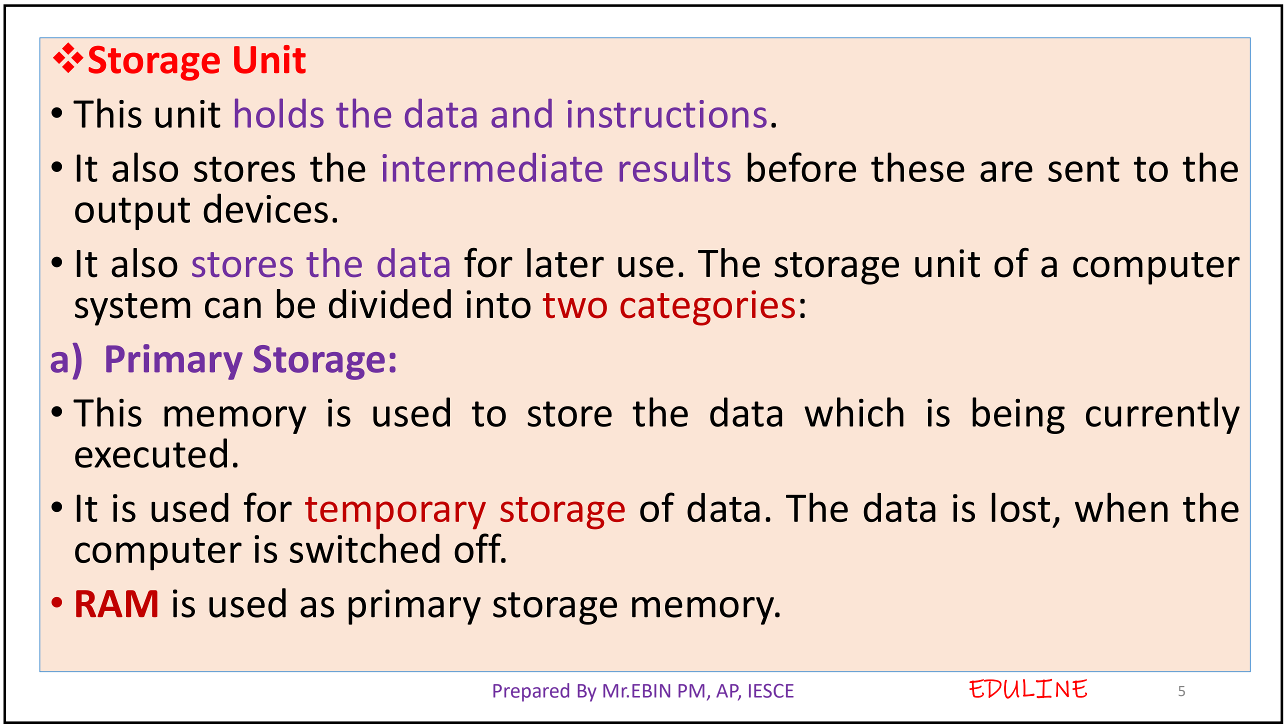
➢Input unit performs following tasks:

- Accept the data and instructions from the outside environment.
- Convert it into machine language.
- Supply the converted data to computer system.

❖**Output Unit**

- It connects the internal system of a computer to the external environment.
- It provides the results of any computation, or instructions to the outside world.
- Some output devices are *printers, monitor* etc.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 4

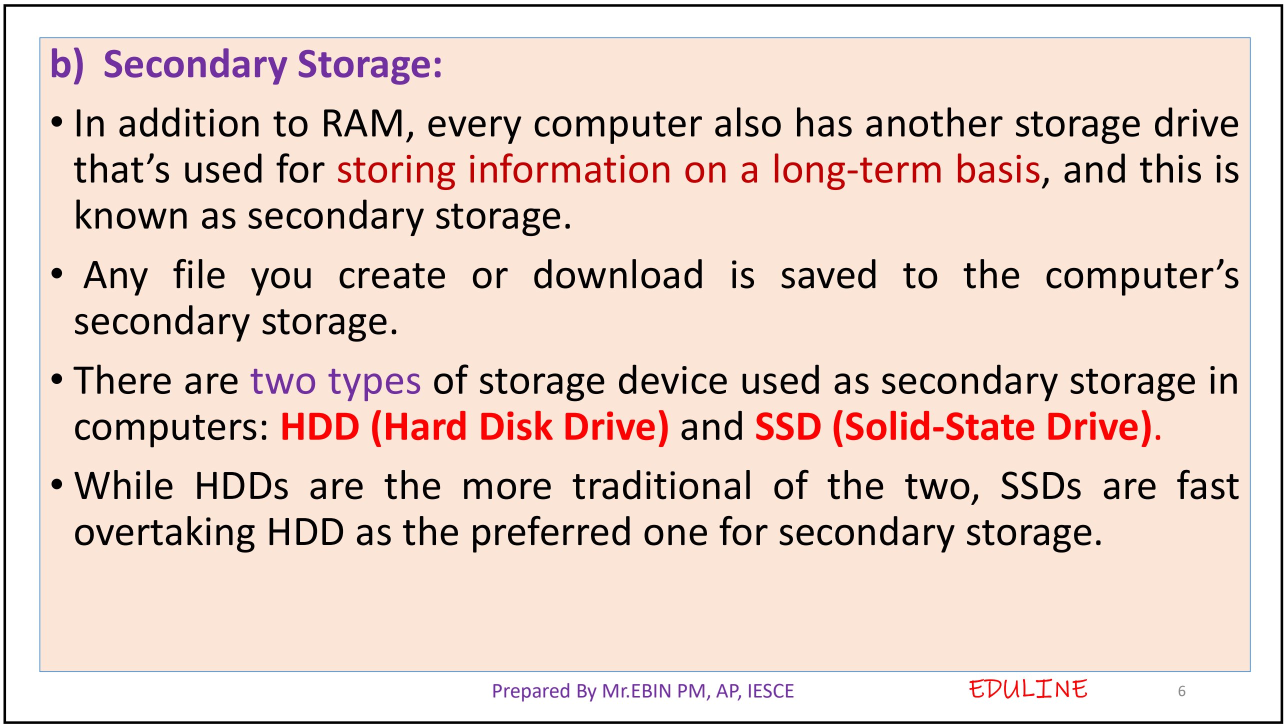


❖Storage Unit

- This unit holds the data and instructions.
- It also stores the intermediate results before these are sent to the output devices.
- It also stores the data for later use. The storage unit of a computer system can be divided into two categories:

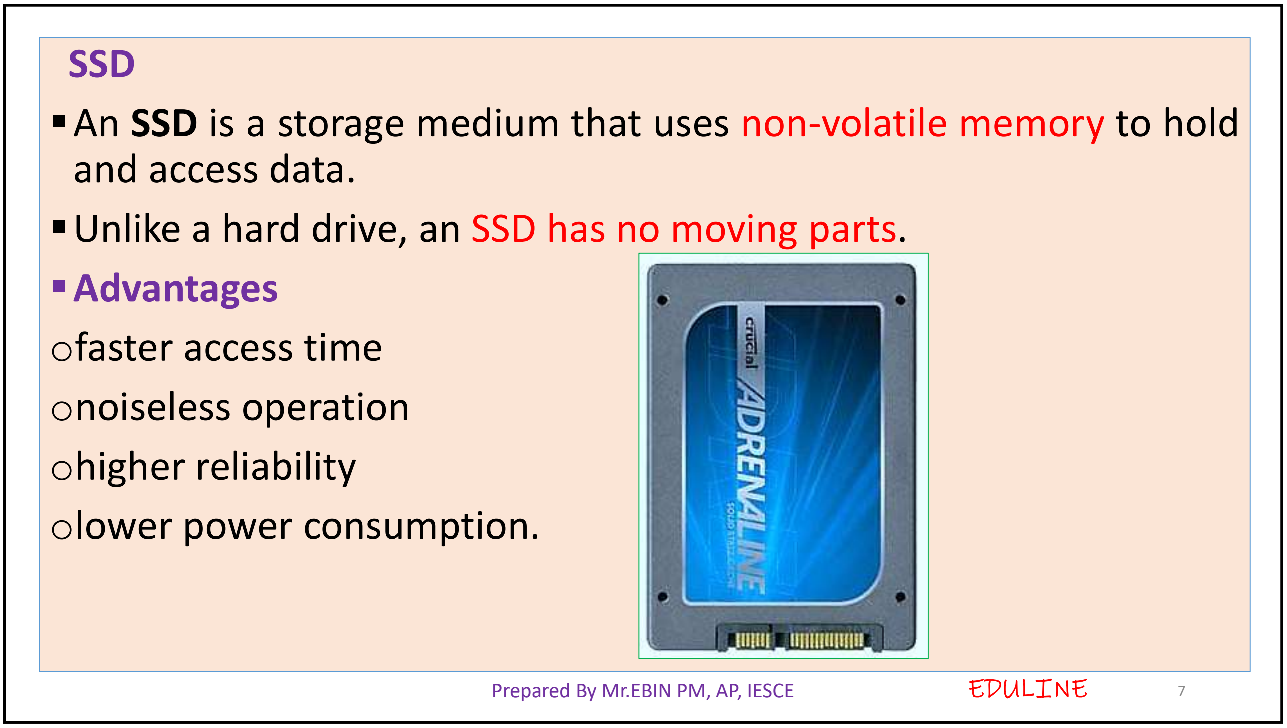
a) Primary Storage:

- This memory is used to store the data which is being currently executed.
- It is used for temporary storage of data. The data is lost, when the computer is switched off.
- **RAM** is used as primary storage memory.



b) Secondary Storage:

- In addition to RAM, every computer also has another storage drive that’s used for storing information on a long-term basis, and this is known as secondary storage.
- Any file you create or download is saved to the computer’s secondary storage.
- There are two types of storage device used as secondary storage in computers: **HDD (Hard Disk Drive)** and **SSD (Solid-State Drive)**.
- While HDDs are the more traditional of the two, SSDs are fast overtaking HDD as the preferred one for secondary storage.



SSD

- An **SSD** is a storage medium that uses non-volatile memory to hold and access data.
- Unlike a hard drive, an SSD has no moving parts.
- **Advantages**

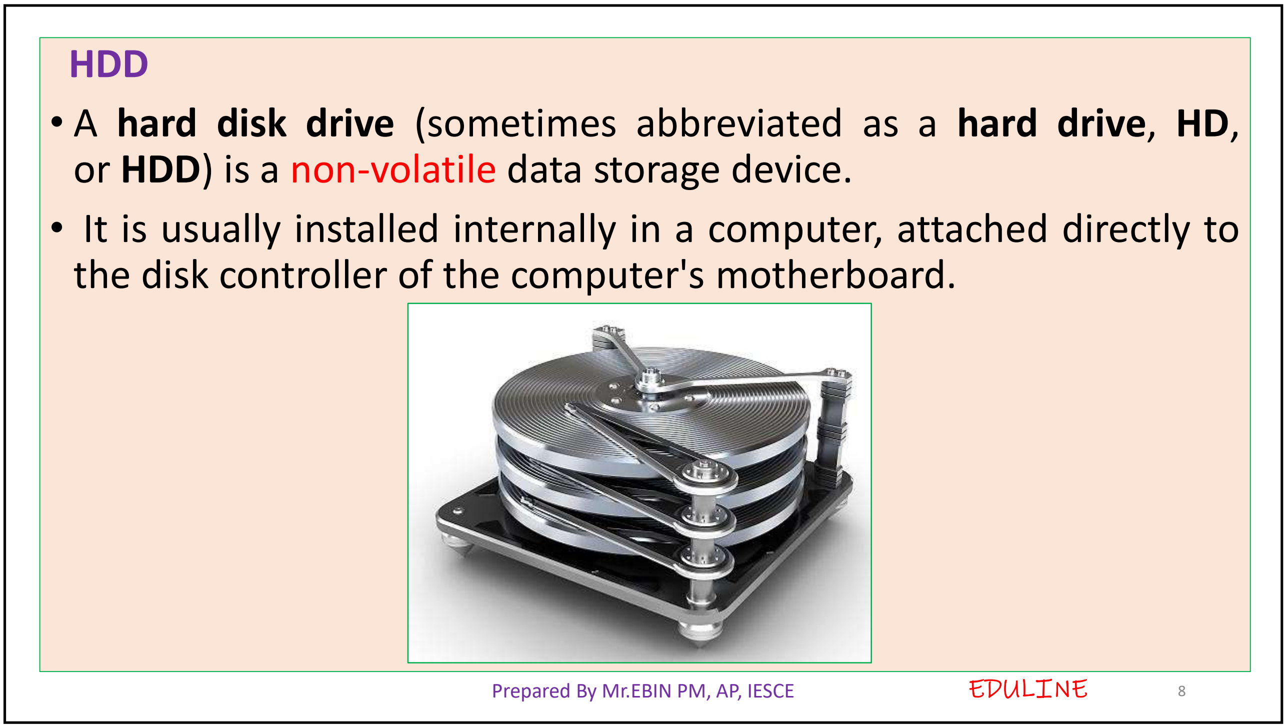
○faster access time

○noiseless operation

○higher reliability

○lower power consumption.

crucial
ADRENALINE



HDD

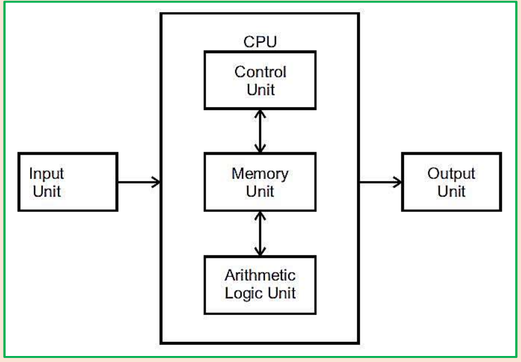
- A **hard disk drive** (sometimes abbreviated as a **hard drive**, **HD**, or **HDD**) is a non-volatile data storage device.
- It is usually installed internally in a computer, attached directly to the disk controller of the computer's motherboard.

❖Arithmetic Logical Unit (ALU)

- All the calculations are performed in ALU of the computer system.
- The ALU can perform basic operations such as addition, subtraction, division, multiplication etc.
- Whenever calculations are required, the control unit transfers the data from storage unit to ALU.
- When the operations are done, the result is transferred back to the storage unit.

❖Control Unit (CU)

- It controls all other units of the computer.
- It controls the flow of data and instructions to and from the storage unit to ALU.
- Thus it is also known as **central nervous system** of the computer.



CPU- CENTRAL PROCESSING UNIT

- A processor is an integrated electronic circuit that performs the calculations that run a computer.
- A processor performs arithmetical, logical, input/output (I/O) and other basic instructions that are passed from an operating system (OS).
- Most other processes are dependent on the operations of a processor.
- The terms processor, CPU and microprocessor are commonly linked

- A processor (CPU)is the logic circuitry that responds to and processes the basic instructions that drive a computer.
- The CPU is seen as the main and most crucial integrated circuitry (IC) chip in a computer, as it is responsible for interpreting most of computers commands.
- CPUs will perform most basic arithmetic, logic and I/O operations, as well as allocate commands for other chips and components running in a computer.

- The term *processor* is used interchangeably with the term central processing unit (CPU),although strictly speaking, the CPU is not the only processor in a computer.
- The **GPU(graphics processing unit)** is the most notable example, but the hard drive and other devices within a computer also perform some processing independently.
- Nevertheless, the term processor is generally understood to mean the CPU.

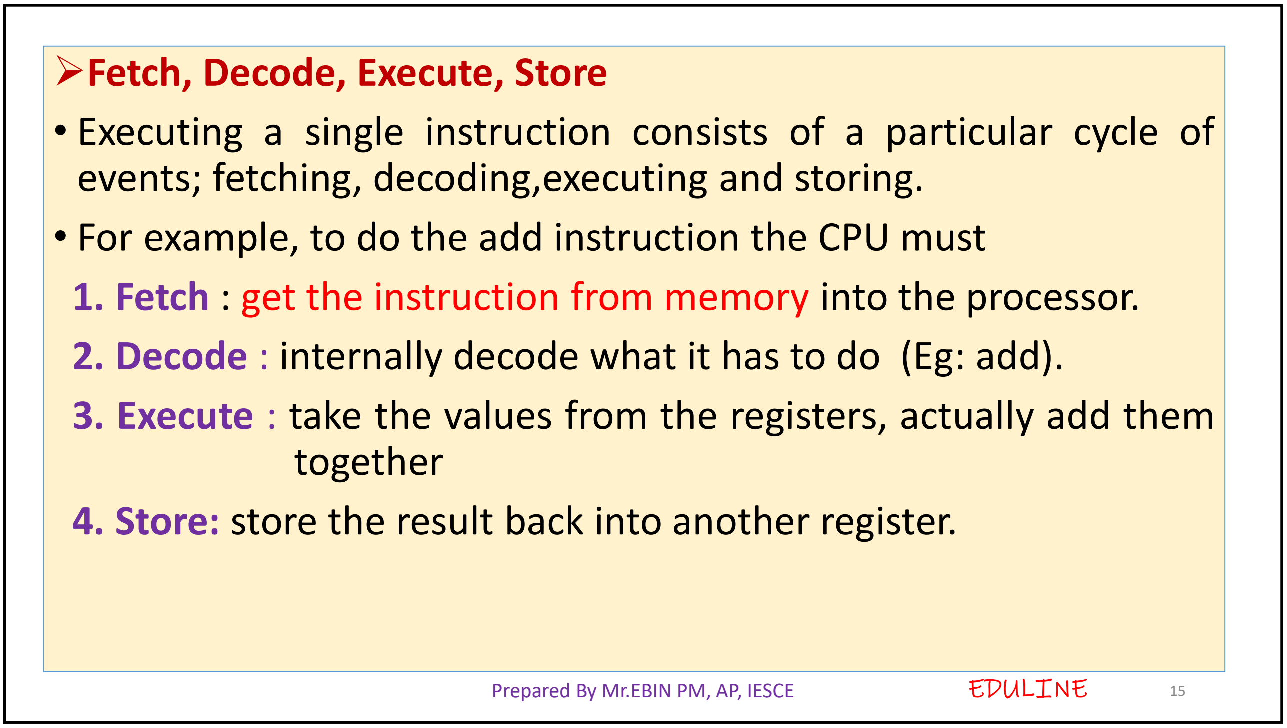
(GPU)

- Processors can be found in PCs, smartphones, tablets and other computers.
- The two main competitors in the processor market are **Intel** and **AMD**.
- CPU is the brain of computer system.
- It performs following tasks:

  It performs all operations.

  It takes all decisions.

  It controls all the units of computer.

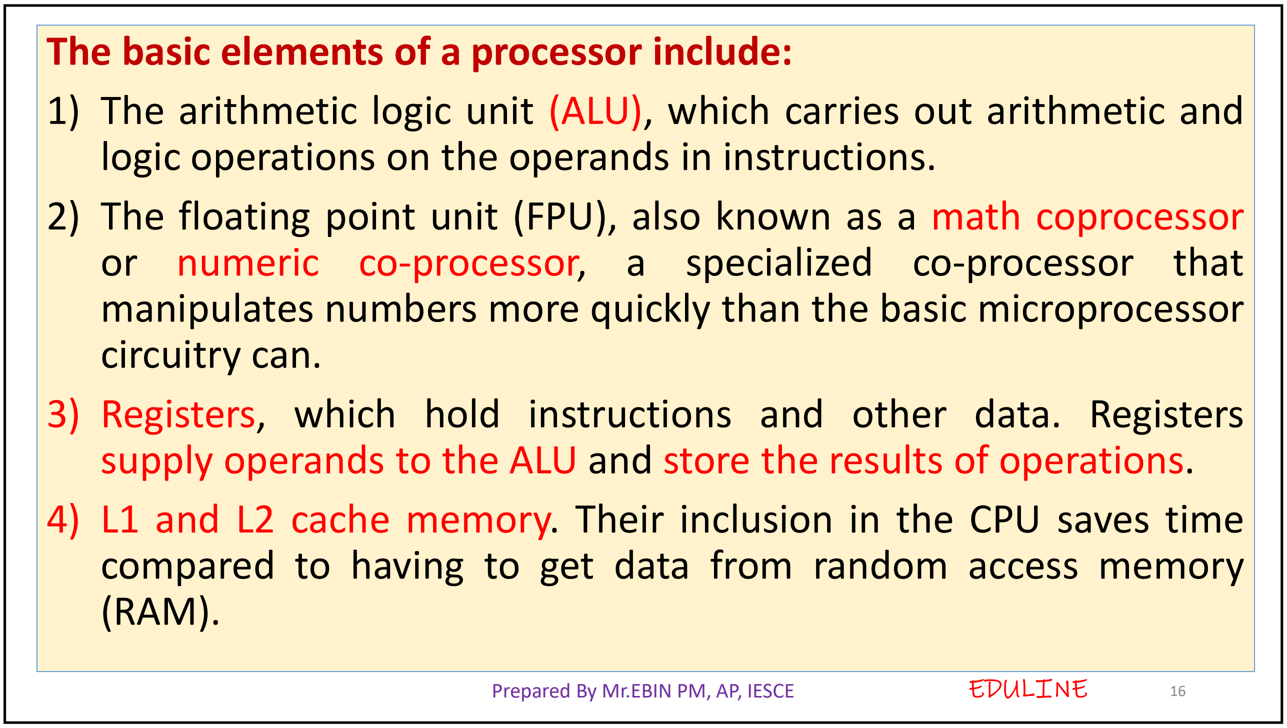


➢Fetch, Decode, Execute, Store

- Executing a single instruction consists of a particular cycle of events; fetching, decoding,executing and storing.
- For example, to do the add instruction the CPU must

1. **Fetch** : get the instruction from memory into the processor.
2. **Decode** : internally decode what it has to do (Eg: add).
3. **Execute** : take the values from the registers, actually add them together
4. **Store:** store the result back into another register.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 15



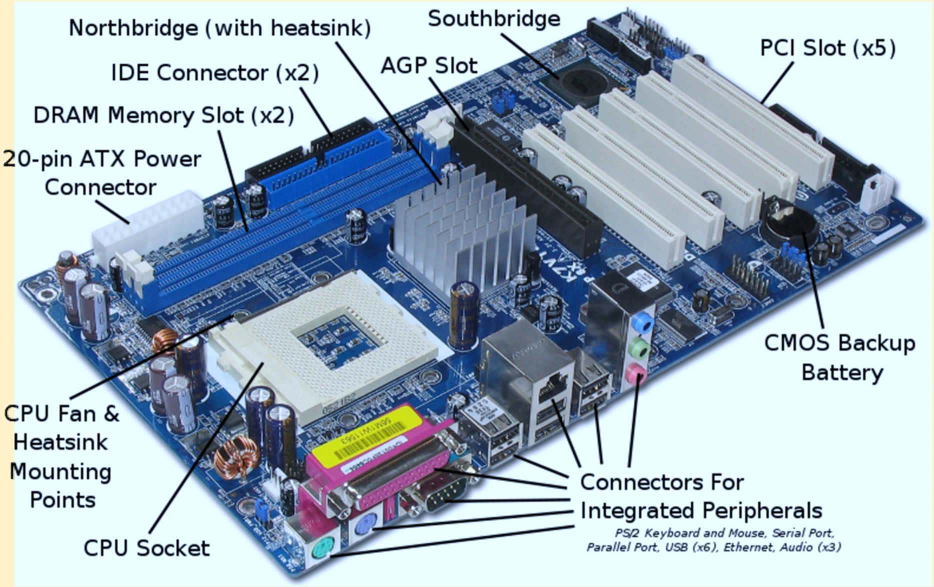
The basic elements of a processor include:

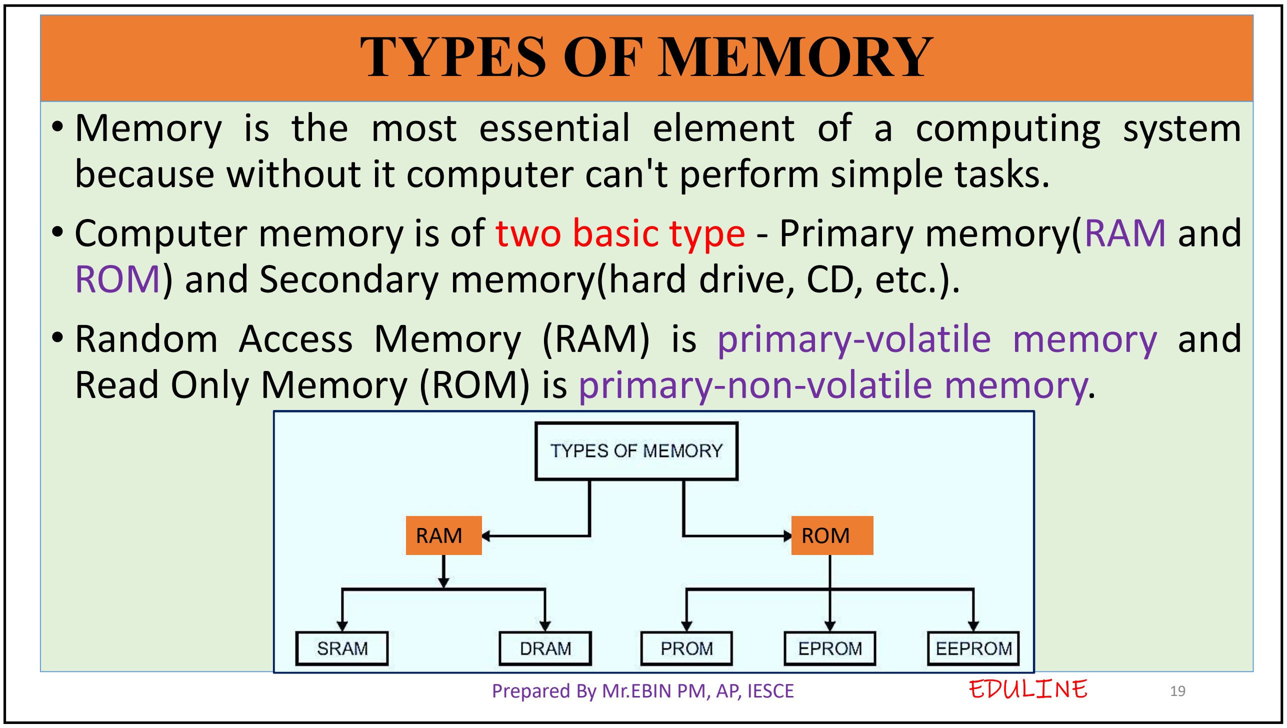
1) The arithmetic logic unit (ALU), which carries out arithmetic and logic operations on the operands in instructions.
2) The floating point unit (FPU), also known as a math coprocessor or numeric co-processor, a specialized co-processor that manipulates numbers more quickly than the basic microprocessor circuitry can.
3) Registers, which hold instructions and other data. Registers supply operands to the ALU and store the results of operations.
4) L1 and L2 cache memory. Their inclusion in the CPU saves time compared to having to get data from random access memory (RAM).

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 16

- Most processors today are multi-core, which means that the IC contains two or more processors for enhanced performance, reduced power consumption and more efficient simultaneous processing of multiple tasks
- Most computers may have up to two-four cores; however, this number can increase up to12 cores.
- for example, If a CPU can only process a single set of instructions at one time, then it is considered as a single-core processor.
- If a CPU can process two sets of instructions at a time it is called a dual-core processor
- four sets would be considered a quad-core processor.

❖MOTHERBOARD


Northbridge (with heatsink)
Southbridge
PCI Slot (x5)
IDE Connector (x2)
AGP Slot
DRAM Memory Slot (x2)
20-pin ATX Power Connector
CMOS Backup Battery
CPU Fan & Heatsink Mounting Points
Connectors For Integrated Peripherals
PS/2 Keyboard and Mouse, Serial Port, Parallel Port, USB (x6), Ethernet, Audio (x3)
CPU Socket

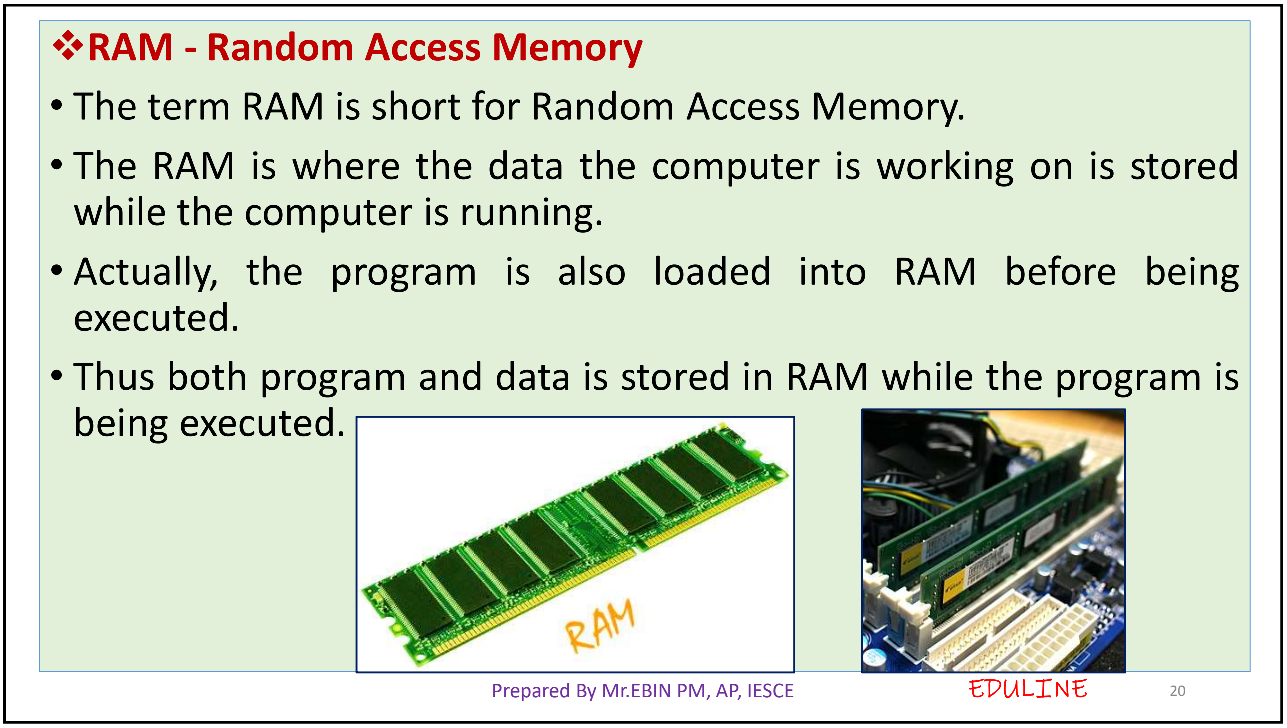


TYPES OF MEMORY

- Memory is the most essential element of a computing system because without it computer can't perform simple tasks.
- Computer memory is of two basic type - Primary memory(RAM and ROM) and Secondary memory(hard drive, CD, etc.).
- Random Access Memory (RAM) is primary-volatile memory and Read Only Memory (ROM) is primary-non-volatile memory.

TYPES OF MEMORY
RAM
ROM
SRAM
DRAM
PROM
EPROM
EEPROM

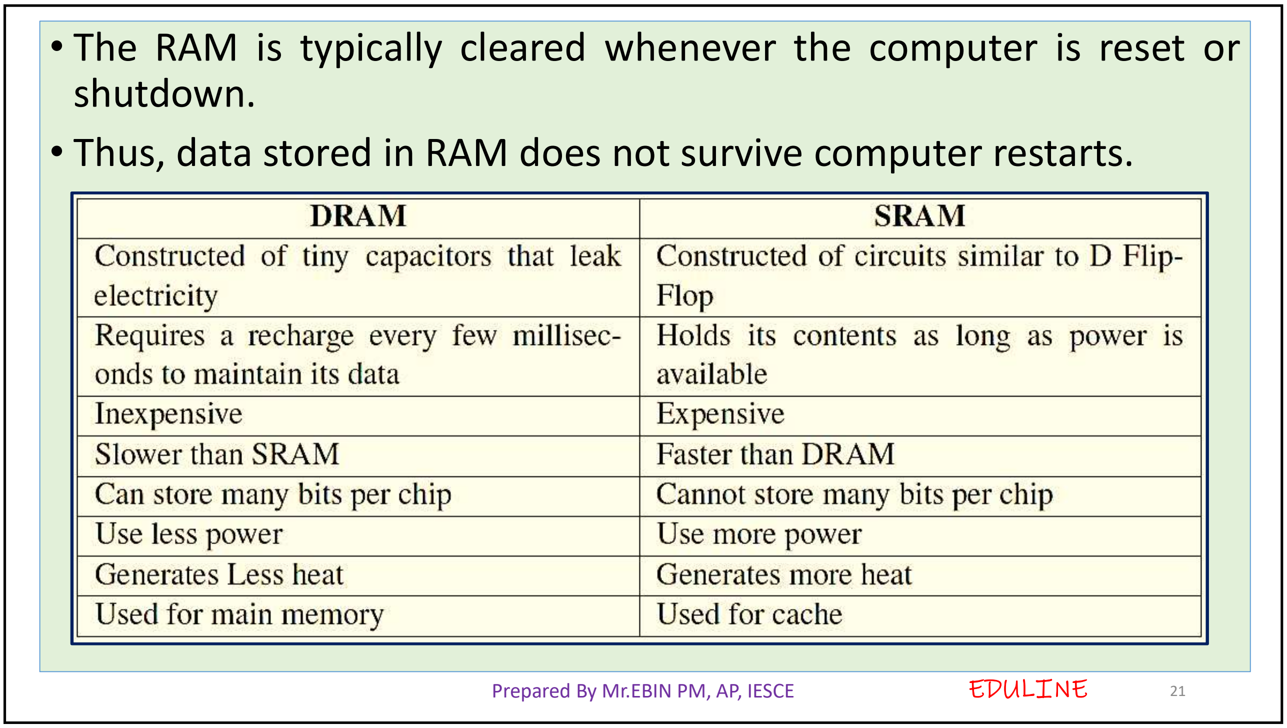
Prepared By Mr.EBIN PM, AP, IESCE EDULINE 19



❖RAM - Random Access Memory

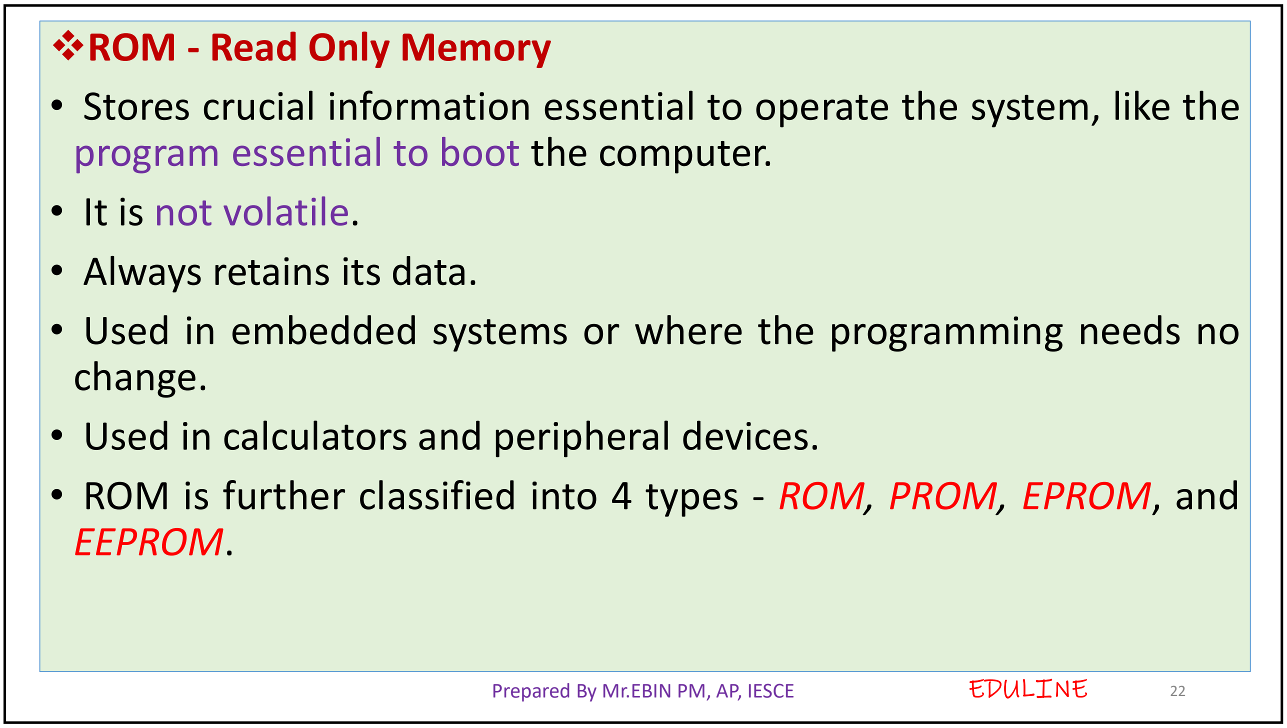
- The term RAM is short for Random Access Memory.
- The RAM is where the data the computer is working on is stored while the computer is running.
- Actually, the program is also loaded into RAM before being executed.
- Thus both program and data is stored in RAM while the program is being executed.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 20



- The RAM is typically cleared whenever the computer is reset or shutdown.
- Thus, data stored in RAM does not survive computer restarts.

DRAM	SRAM
Constructed of tiny capacitors that leak electricity	Constructed of circuits similar to D Flip-Flop
Requires a recharge every few milliseconds to maintain its data	Holds its contents as long as power is available
Inexpensive	Expensive
Slower than SRAM	Faster than DRAM
Can store many bits per chip	Cannot store many bits per chip
Use less power	Use more power
Generates Less heat	Generates more heat
Used for main memory	Used for cache



❖ROM - Read Only Memory

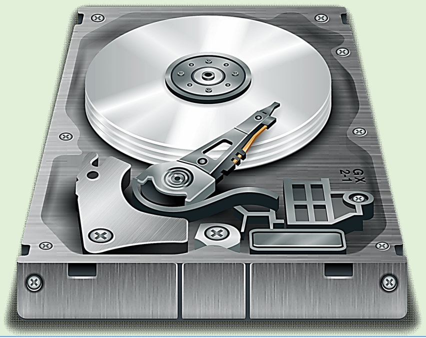
- Stores crucial information essential to operate the system, like the program essential to boot the computer.
- It is not volatile.
- Always retains its data.
- Used in embedded systems or where the programming needs no change.
- Used in calculators and peripheral devices.
- ROM is further classified into 4 types - *ROM*, *PROM*, *EPROM*, and *EEPROM*.

➢Types of Read Only Memory

- PROM (Programmable read-only memory) - It can be programmed by user. Once programmed, the data and instructions in it cannot be changed.
- EPROM (Erasable Programmable read only memory) - It can be reprogrammed. To erase data from it, expose it to ultra violet light. To reprogram it, erase all the previous data.
- EEPROM (Electrically erasable programmable read only memory) - The data can be erased by applying electric field, no need of ultra violet light. We can erase only portions of the chip.

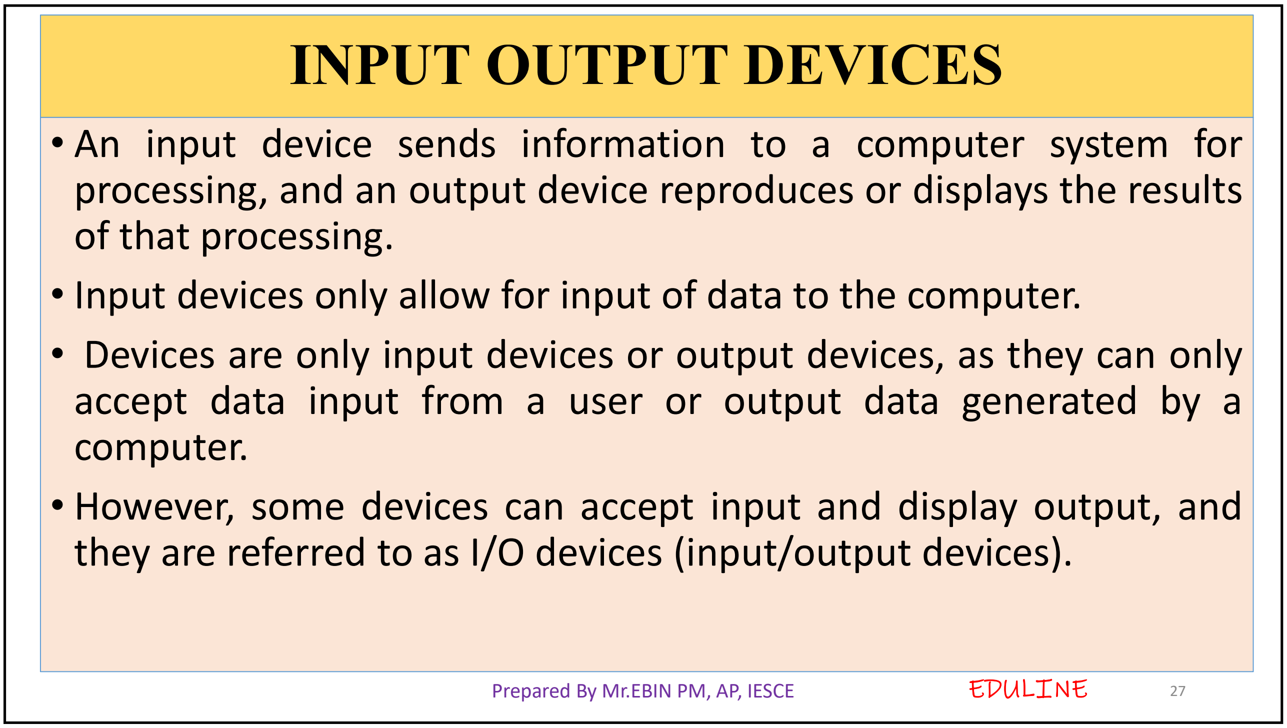
❖Hard Disk

- The hard disk is a device that can store data and have the data survivor power-off and resets of the computer.
- Data stored on a hard disk can be read at a later point, no matter how longtime the computer has been powered off - provided that the computer and hard disk is still working at the time it is again powered on.



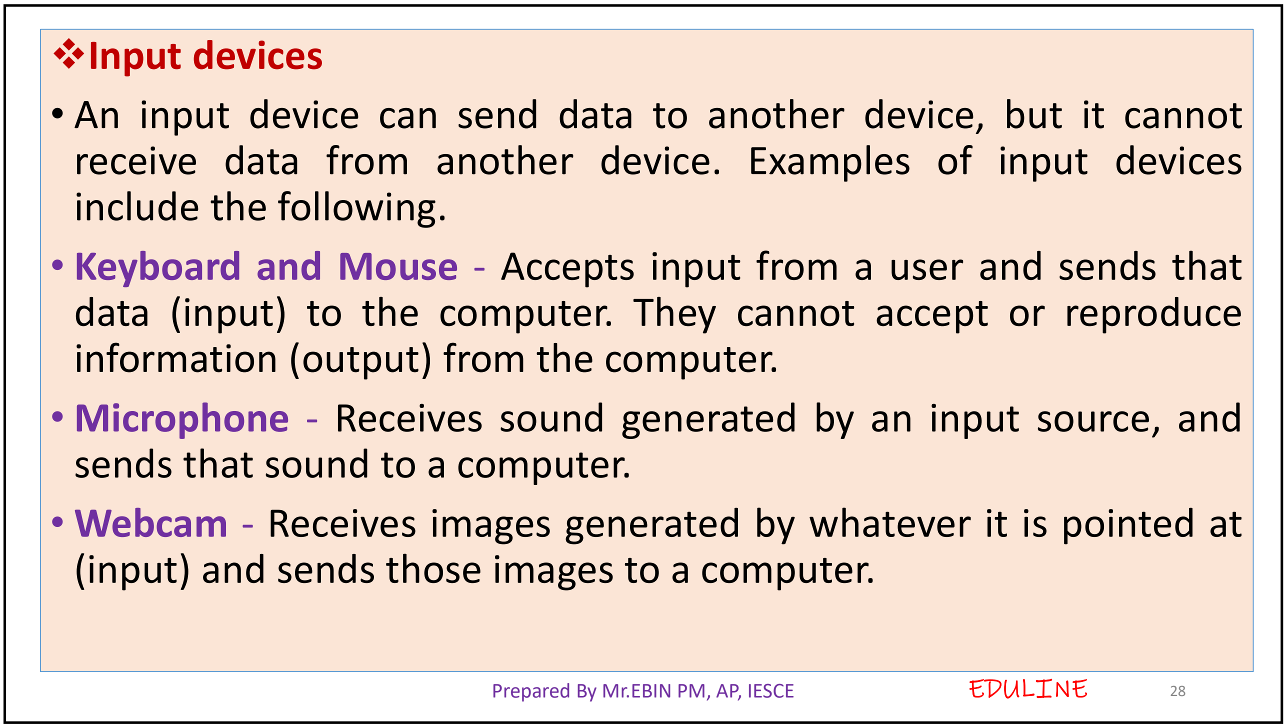
- The hard disk is where you normally store your documents and files.
- It is also on the hard disk that the computer's operating system is stored, and where the programs you have installed on the computer are stored.
- The hard disk is also where your own programs are stored.
- When the computer is told to execute a program, it loads the program from the hard disk into RAM and executes it from there.
- Programs can also read from and write to the hard disk.
- For instance, the program Microsoft Word can read and write Word documents from and to the hard disk

HDD



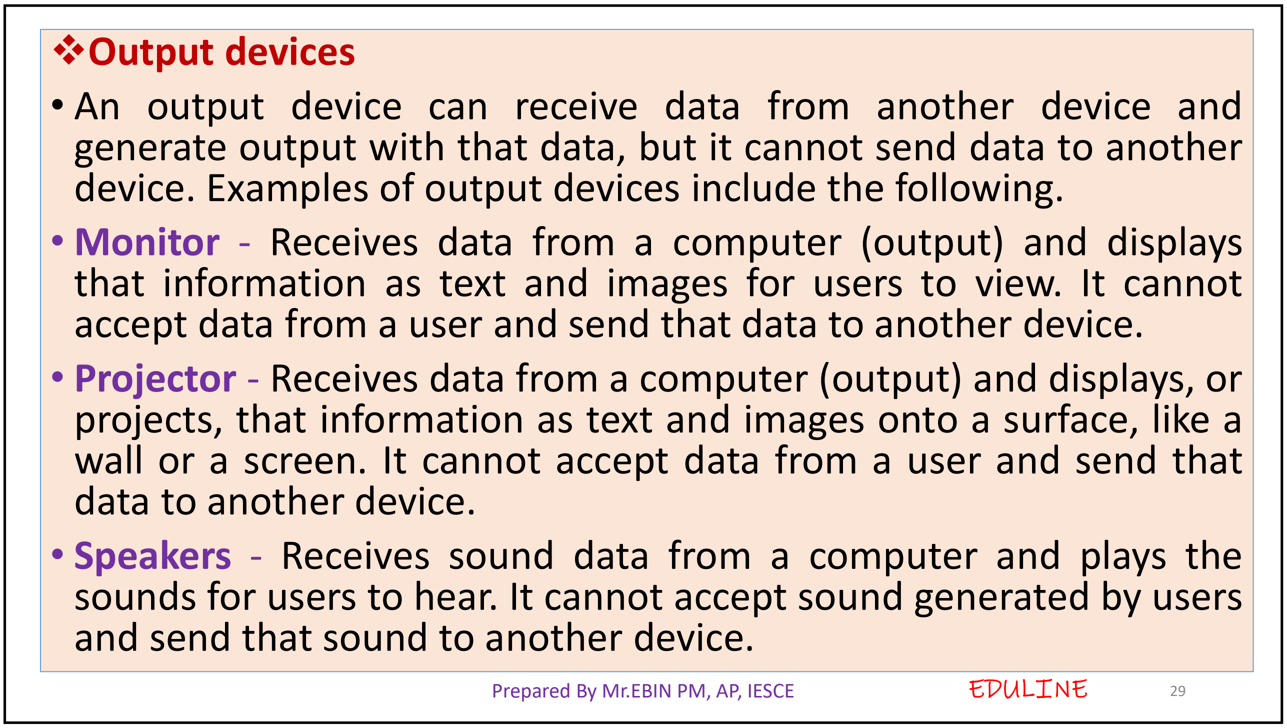
INPUT OUTPUT DEVICES

- An input device sends information to a computer system for processing, and an output device reproduces or displays the results of that processing.
- Input devices only allow for input of data to the computer.
- Devices are only input devices or output devices, as they can only accept data input from a user or output data generated by a computer.
- However, some devices can accept input and display output, and they are referred to as I/O devices (input/output devices).



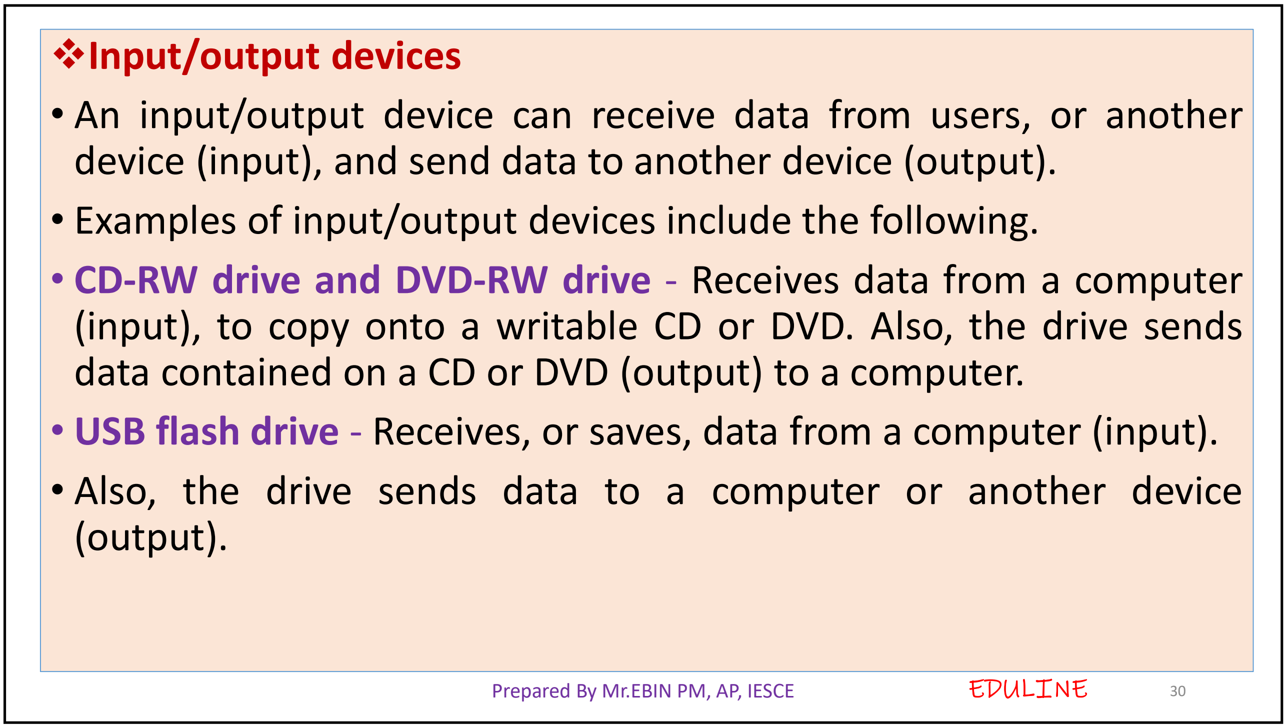
❖Input devices

- An input device can send data to another device, but it cannot receive data from another device. Examples of input devices include the following.
- **Keyboard and Mouse** - Accepts input from a user and sends that data (input) to the computer. They cannot accept or reproduce information (output) from the computer.
- **Microphone** - Receives sound generated by an input source, and sends that sound to a computer.
- **Webcam** - Receives images generated by whatever it is pointed at (input) and sends those images to a computer.



❖Output devices

- An output device can receive data from another device and generate output with that data, but it cannot send data to another device. Examples of output devices include the following.
- **Monitor** - Receives data from a computer (output) and displays that information as text and images for users to view. It cannot accept data from a user and send that data to another device.
- **Projector** - Receives data from a computer (output) and displays, or projects, that information as text and images onto a surface, like a wall or a screen. It cannot accept data from a user and send that data to another device.
- **Speakers** - Receives sound data from a computer and plays the sounds for users to hear. It cannot accept sound generated by users and send that sound to another device.



❖Input/output devices

- An input/output device can receive data from users, or another device (input), and send data to another device (output).
- Examples of input/output devices include the following.
- **CD-RW drive and DVD-RW drive** - Receives data from a computer (input), to copy onto a writable CD or DVD. Also, the drive sends data contained on a CD or DVD (output) to a computer.
- **USB flash drive** - Receives, or saves, data from a computer (input).
- Also, the drive sends data to a computer or another device (output).

SYSTEM SOFTWARES

❖Software

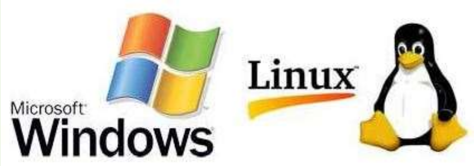
- Software is a collection of instructions that enable the user to interact with a computer , its hardware or perform tasks
- Without software, most computers would be useless. For example, without your Internet browser software, you could not surf the Internet. Without an operating system, the browser could not run on your computer.

There are two types of software

1. System Software
2. Application Software

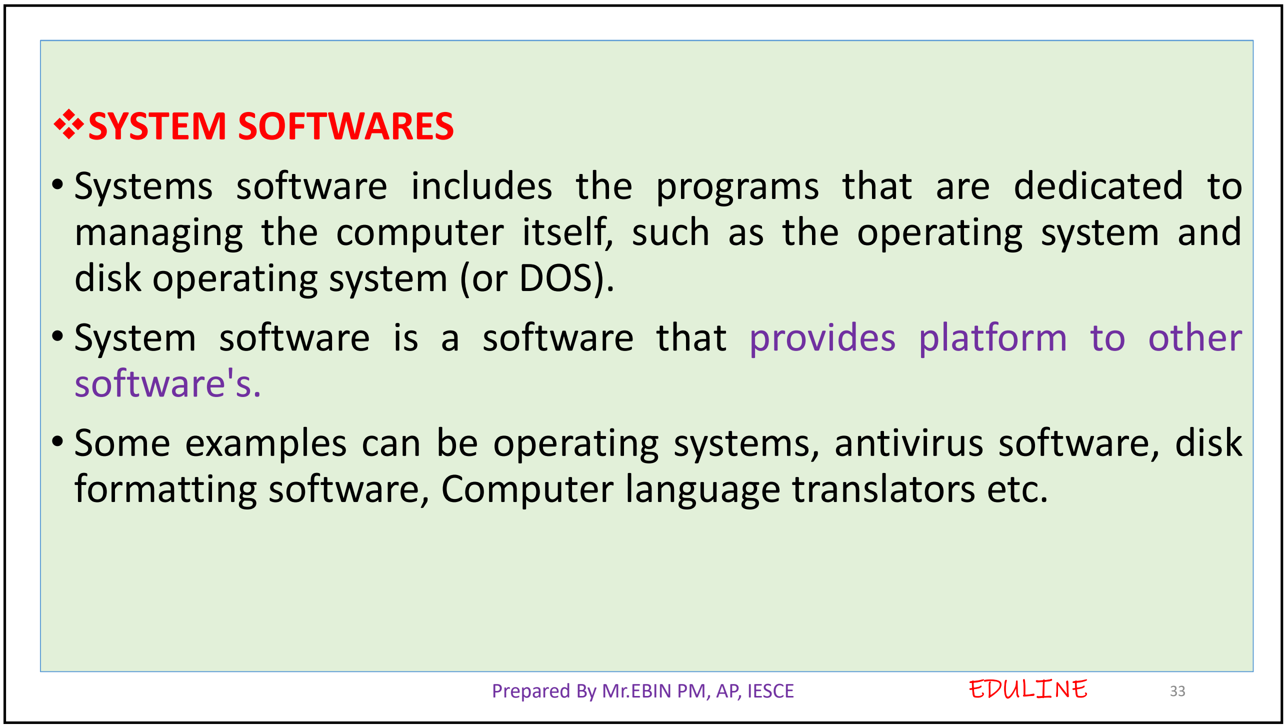
➢Examples of system software are Operating System, Compilers, Interpreter, Assemblers, etc.

➢Examples of Application software are Railways Reservation Software, Microsoft Office Suite Software, Microsoft Word, Microsoft PowerPoint , etc.



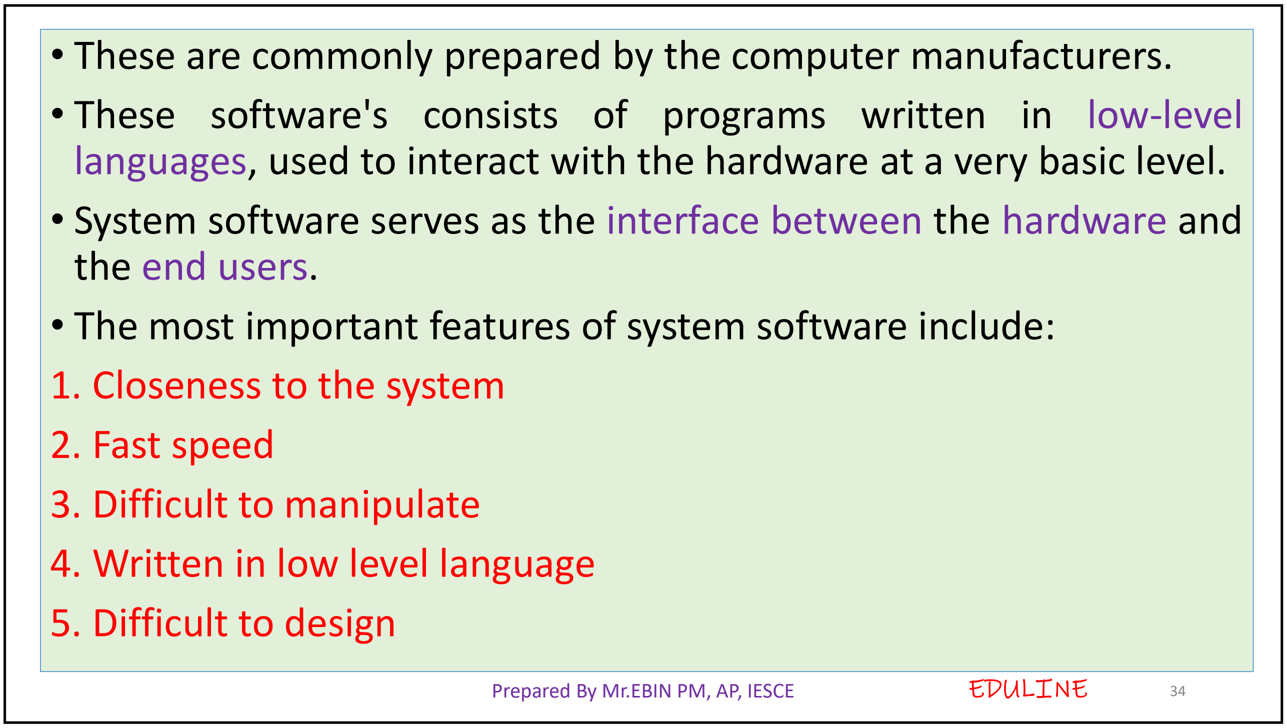
System Software's

Application software's



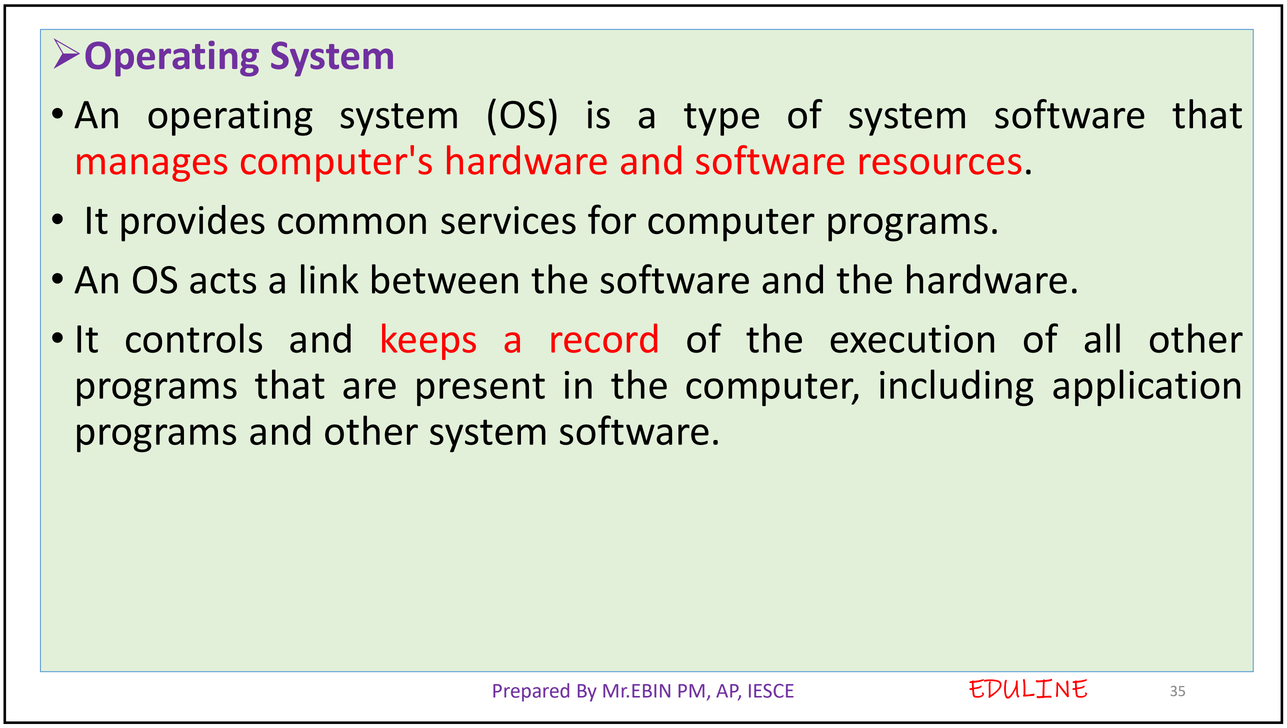
❖SYSTEM SOFTWARES

- Systems software includes the programs that are dedicated to managing the computer itself, such as the operating system and disk operating system (or DOS).
- System software is a software that provides platform to other software's.
- Some examples can be operating systems, antivirus software, disk formatting software, Computer language translators etc.



- These are commonly prepared by the computer manufacturers.
- These software's consists of programs written in low-level languages, used to interact with the hardware at a very basic level.
- System software serves as the interface between the hardware and the end users.
- The most important features of system software include:

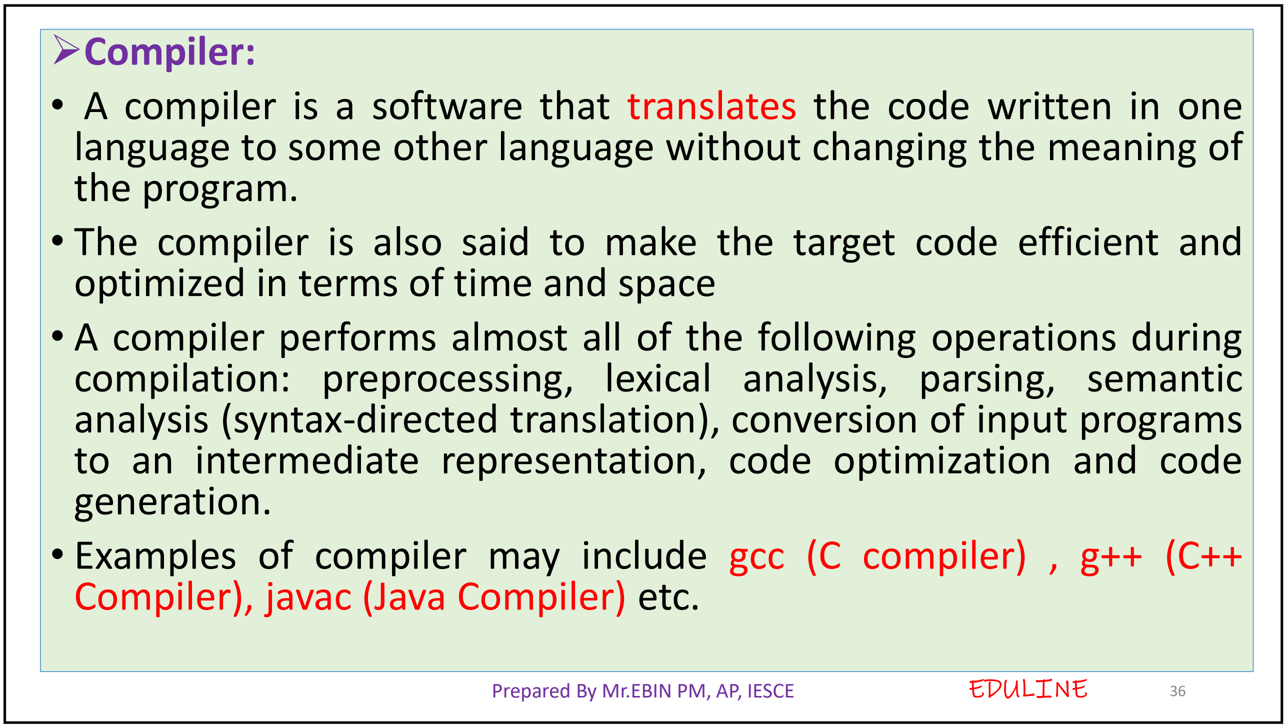
1. Closeness to the system
2. Fast speed
3. Difficult to manipulate
4. Written in low level language
5. Difficult to design



➢Operating System

- An operating system (OS) is a type of system software that manages computer's hardware and software resources.
- It provides common services for computer programs.
- An OS acts a link between the software and the hardware.
- It controls and keeps a record of the execution of all other programs that are present in the computer, including application programs and other system software.

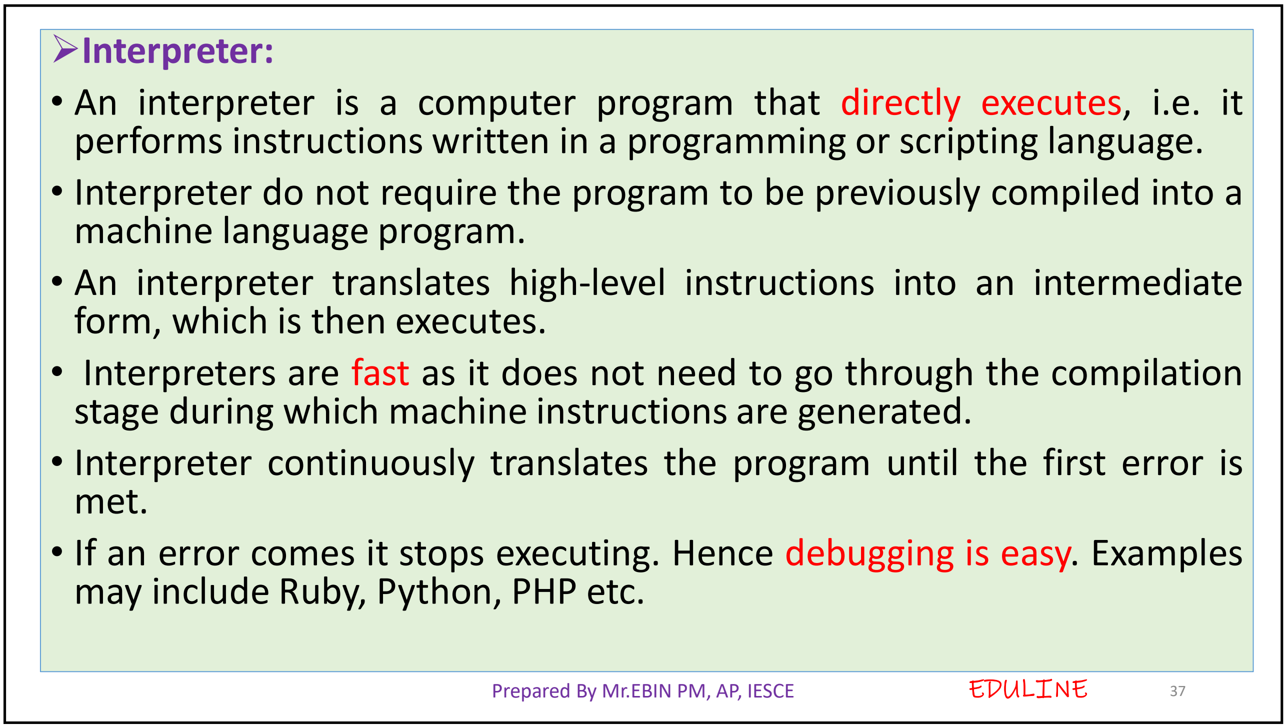
Prepared By Mr.EBIN PM, AP, IESCE EDULINE 35



➢Compiler:

- A compiler is a software that translates the code written in one language to some other language without changing the meaning of the program.
- The compiler is also said to make the target code efficient and optimized in terms of time and space
- A compiler performs almost all of the following operations during compilation: preprocessing, lexical analysis, parsing, semantic analysis (syntax-directed translation), conversion of input programs to an intermediate representation, code optimization and code generation.
- Examples of compiler may include gcc (C compiler) , g++ (C++ Compiler), javac (Java Compiler) etc.

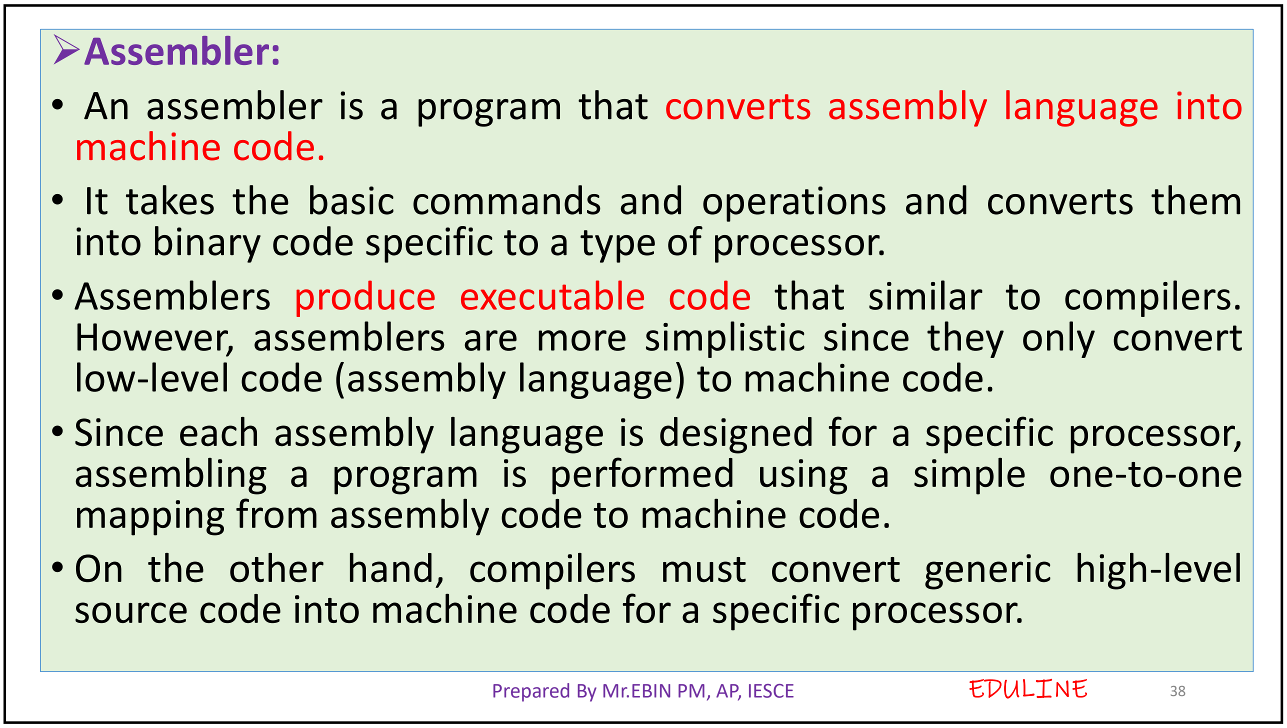
Prepared By Mr.EBIN PM, AP, IESCE EDULINE 36



➢Interpreter:

- An interpreter is a computer program that directly executes, i.e. it performs instructions written in a programming or scripting language.
- Interpreter do not require the program to be previously compiled into a machine language program.
- An interpreter translates high-level instructions into an intermediate form, which is then executes.
- Interpreters are fast as it does not need to go through the compilation stage during which machine instructions are generated.
- Interpreter continuously translates the program until the first error is met.
- If an error comes it stops executing. Hence debugging is easy. Examples may include Ruby, Python, PHP etc.

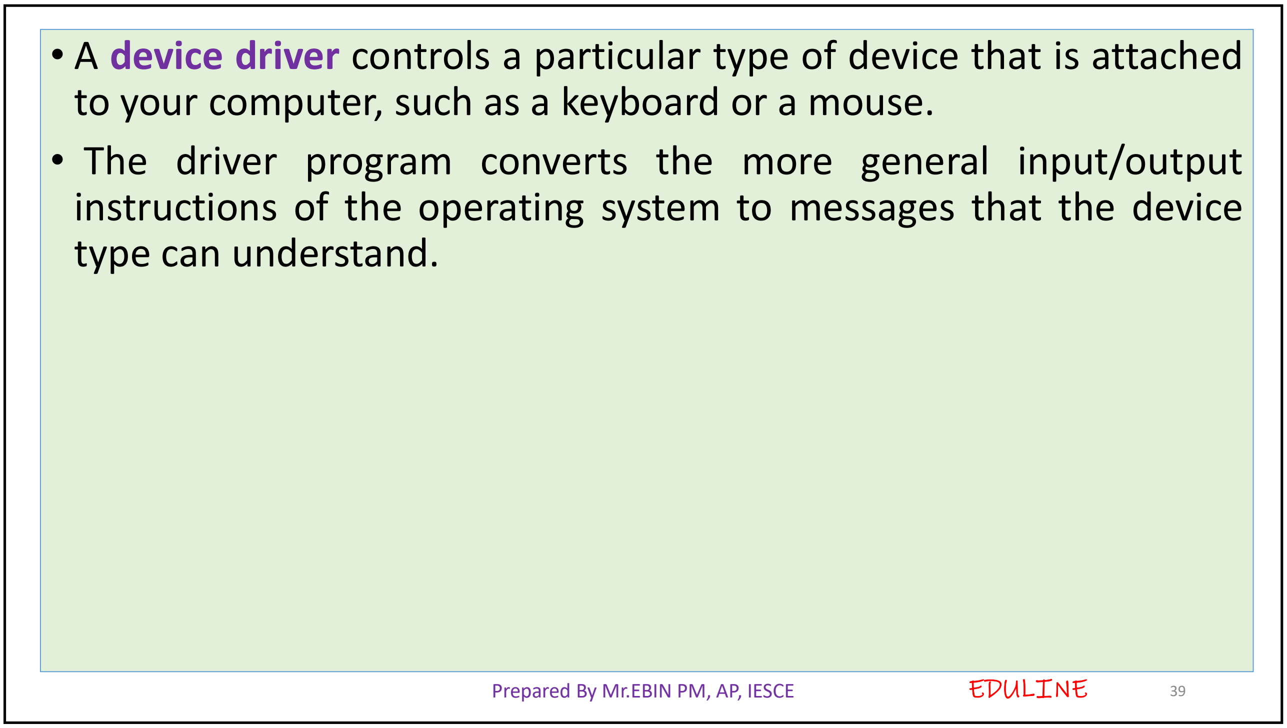
Prepared By Mr.EBIN PM, AP, IESCE EDULINE 37



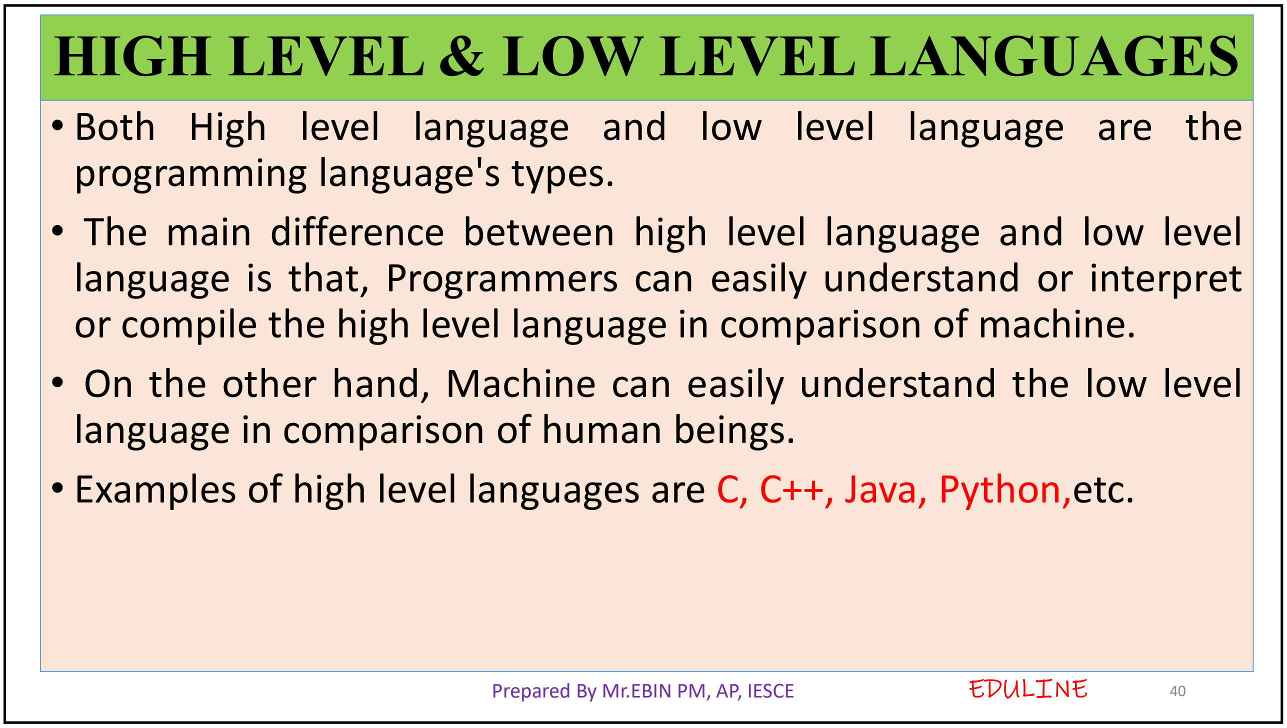
➢Assembler:

- An assembler is a program that converts assembly language into machine code.
- It takes the basic commands and operations and converts them into binary code specific to a type of processor.
- Assemblers produce executable code that similar to compilers. However, assemblers are more simplistic since they only convert low-level code (assembly language) to machine code.
- Since each assembly language is designed for a specific processor, assembling a program is performed using a simple one-to-one mapping from assembly code to machine code.
- On the other hand, compilers must convert generic high-level source code into machine code for a specific processor.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 38

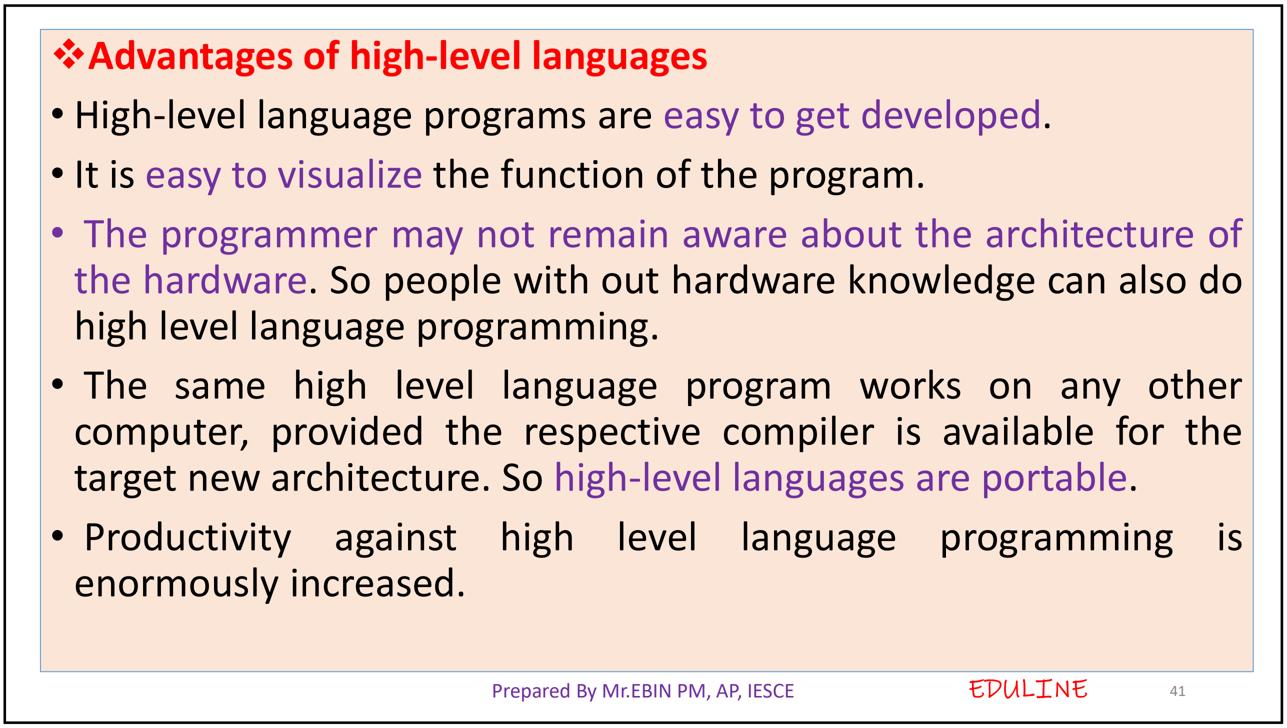


- A **device driver** controls a particular type of device that is attached to your computer, such as a keyboard or a mouse.
- The driver program converts the more general input/output instructions of the operating system to messages that the device type can understand.



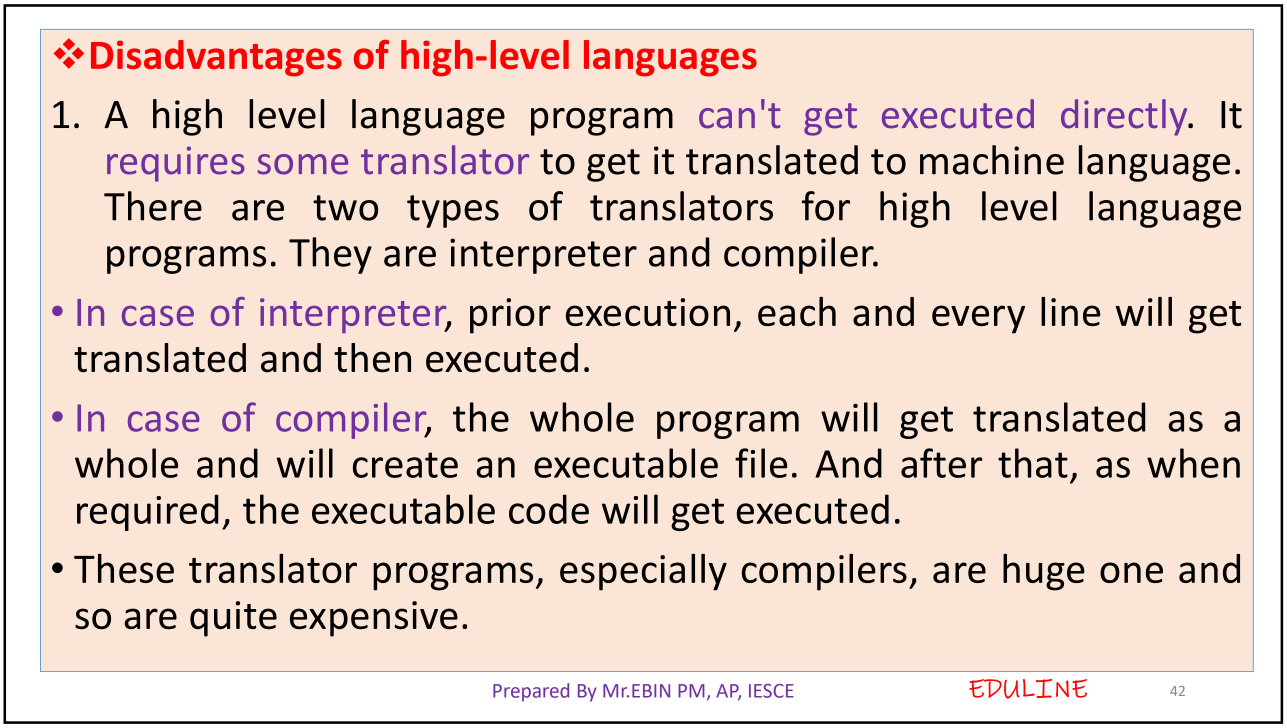
HIGH LEVEL & LOW LEVEL LANGUAGES

- Both High level language and low level language are the programming language's types.
- The main difference between high level language and low level language is that, Programmers can easily understand or interpret or compile the high level language in comparison of machine.
- On the other hand, Machine can easily understand the low level language in comparison of human beings.
- Examples of high level languages are C, C++, Java, Python,etc.



❖Advantages of high-level languages

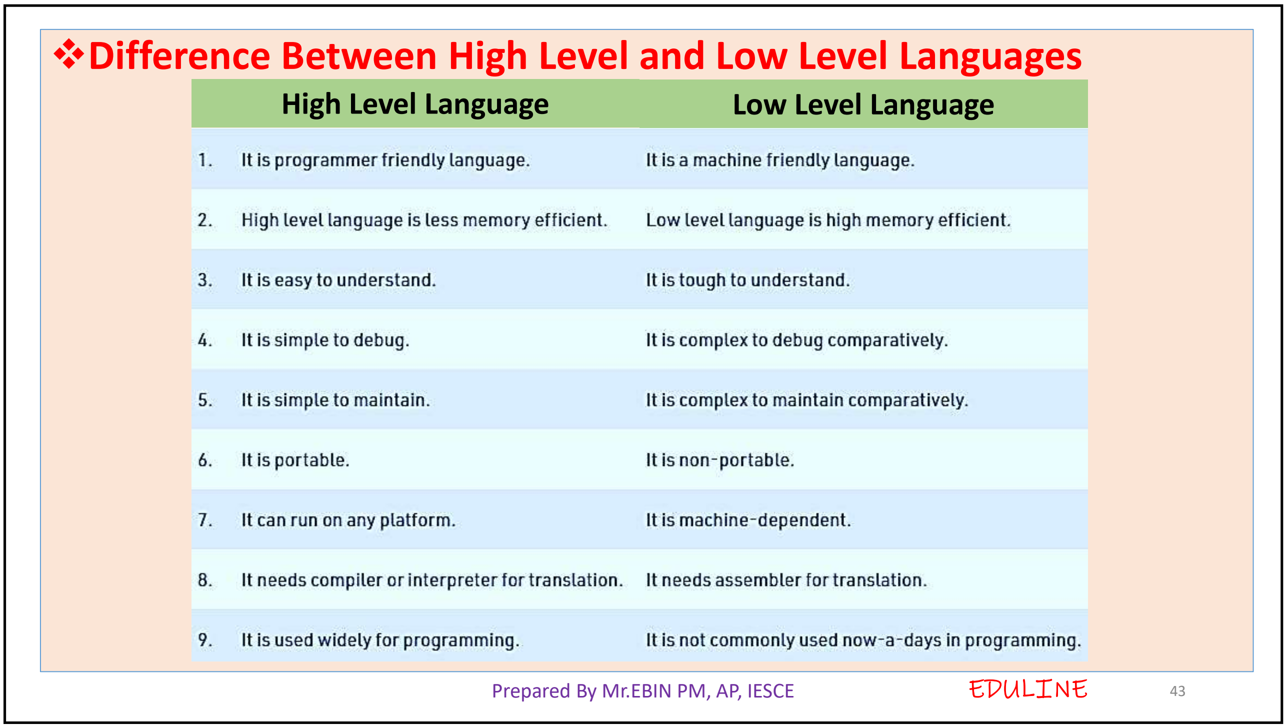
- High-level language programs are easy to get developed.
- It is easy to visualize the function of the program.
- The programmer may not remain aware about the architecture of the hardware. So people with out hardware knowledge can also do high level language programming.
- The same high level language program works on any other computer, provided the respective compiler is available for the target new architecture. So high-level languages are portable.
- Productivity against high level language programming is enormously increased.



❖Disadvantages of high-level languages

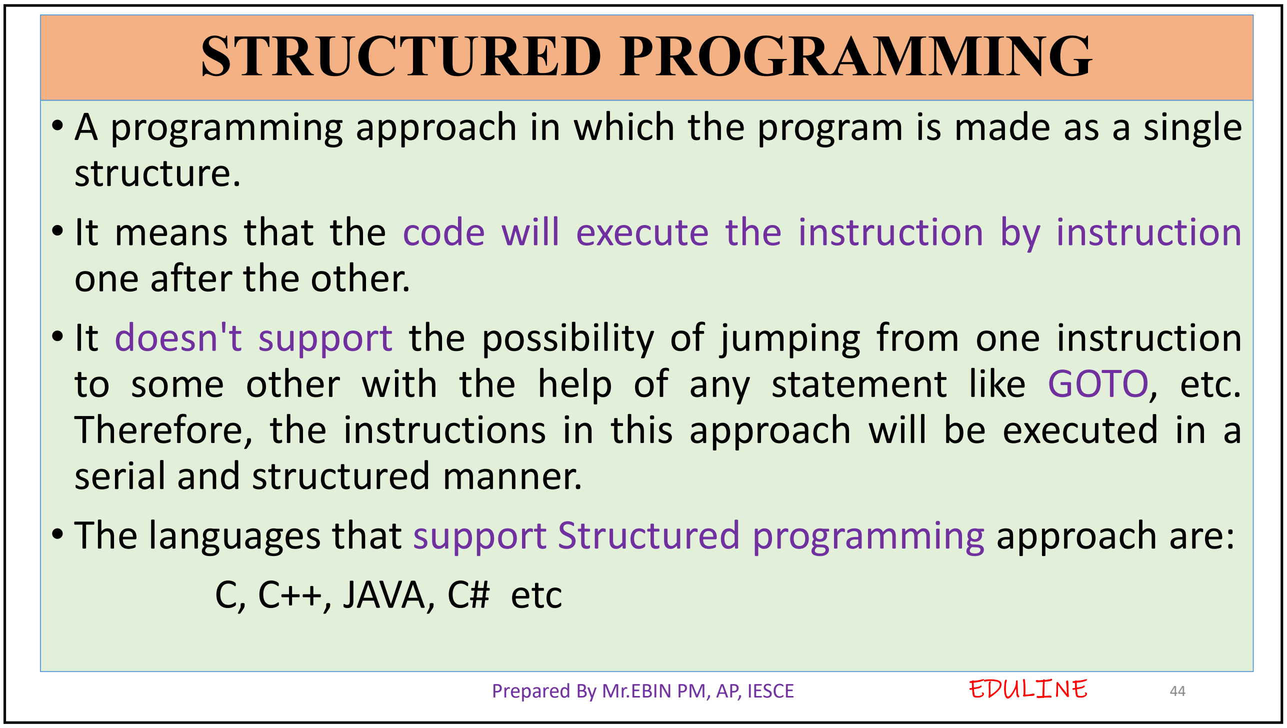
1. A high level language program can't get executed directly. It requires some translator to get it translated to machine language. There are two types of translators for high level language programs. They are interpreter and compiler.

- In case of interpreter, prior execution, each and every line will get translated and then executed.
- In case of compiler, the whole program will get translated as a whole and will create an executable file. And after that, as when required, the executable code will get executed.
- These translator programs, especially compilers, are huge one and so are quite expensive.



❖Difference Between High Level and Low Level Languages

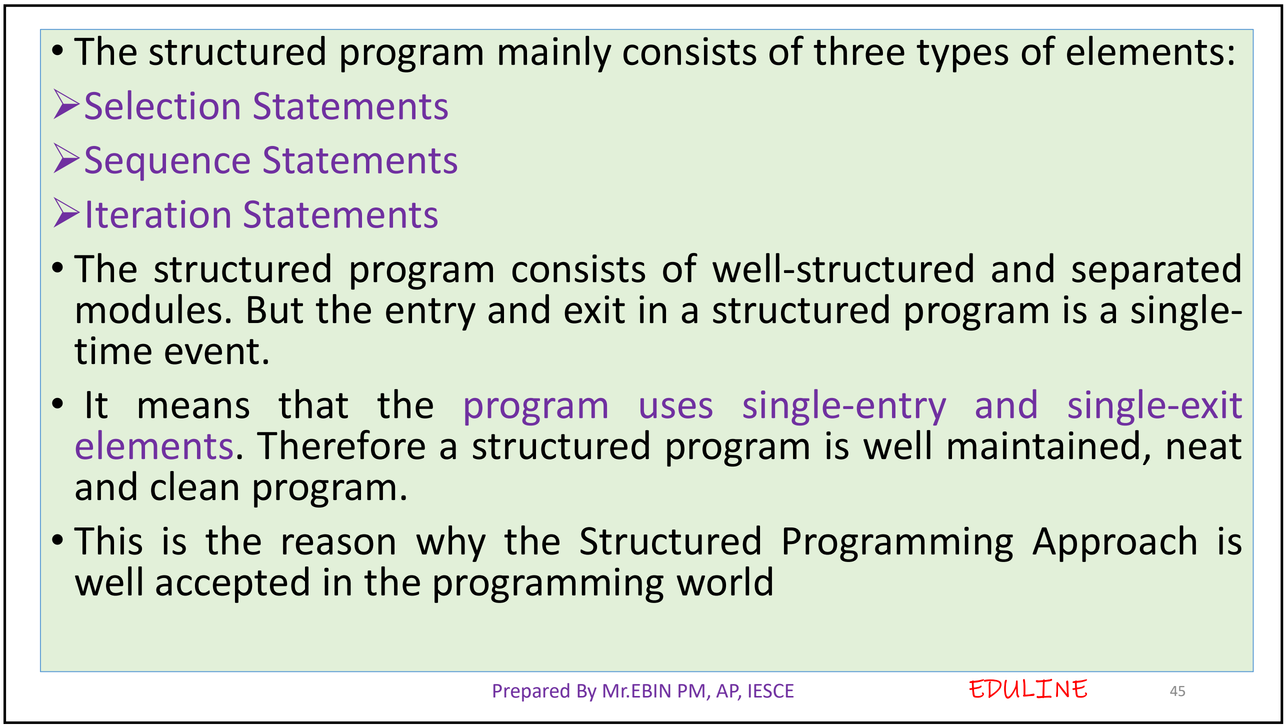
	High Level Language	Low Level Language
1.	It is programmer friendly language.	It is a machine friendly language.
2.	High level language is less memory efficient.	Low level language is high memory efficient.
3.	It is easy to understand.	It is tough to understand.
4.	It is simple to debug.	It is complex to debug comparatively.
5.	It is simple to maintain.	It is complex to maintain comparatively.
6.	It is portable.	It is non-portable.
7.	It can run on any platform.	It is machine-dependent.
8.	It needs compiler or interpreter for translation.	It needs assembler for translation.
9.	It is used widely for programming.	It is not commonly used now-a-days in programming.



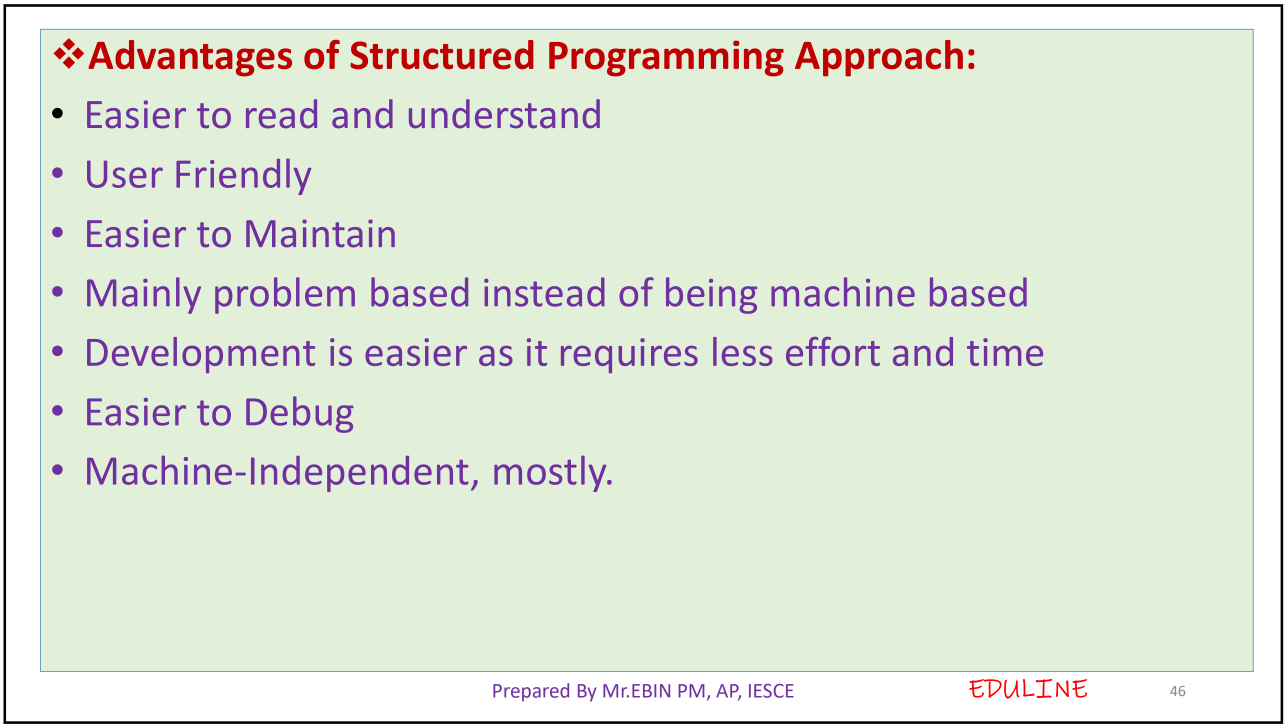
STRUCTURED PROGRAMMING

- A programming approach in which the program is made as a single structure.
- It means that the code will execute the instruction by instruction one after the other.
- It doesn't support the possibility of jumping from one instruction to some other with the help of any statement like GOTO, etc. Therefore, the instructions in this approach will be executed in a serial and structured manner.
- The languages that support Structured programming approach are:

C, C++, JAVA, C# etc

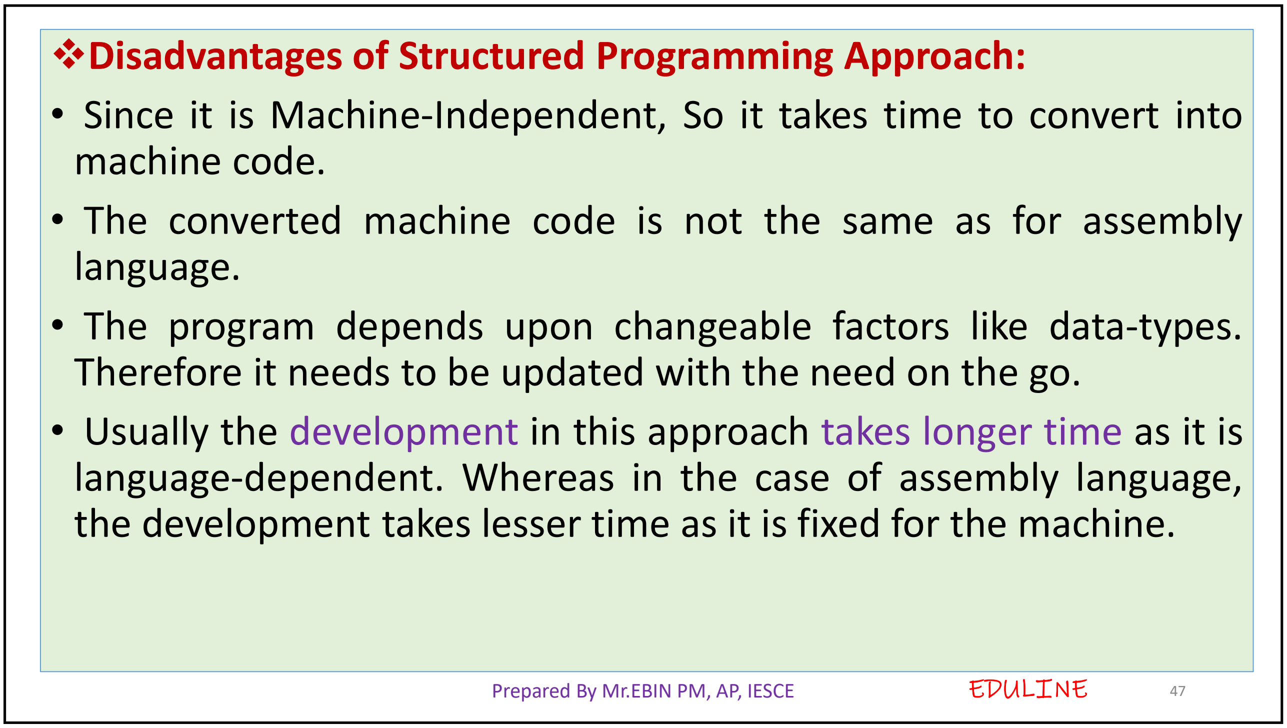


- The structured program mainly consists of three types of elements:
  - Selection Statements
  - Sequence Statements
  - Iteration Statements
- The structured program consists of well-structured and separated modules. But the entry and exit in a structured program is a single-time event.
- It means that the program uses single-entry and single-exit elements. Therefore a structured program is well maintained, neat and clean program.
- This is the reason why the Structured Programming Approach is well accepted in the programming world



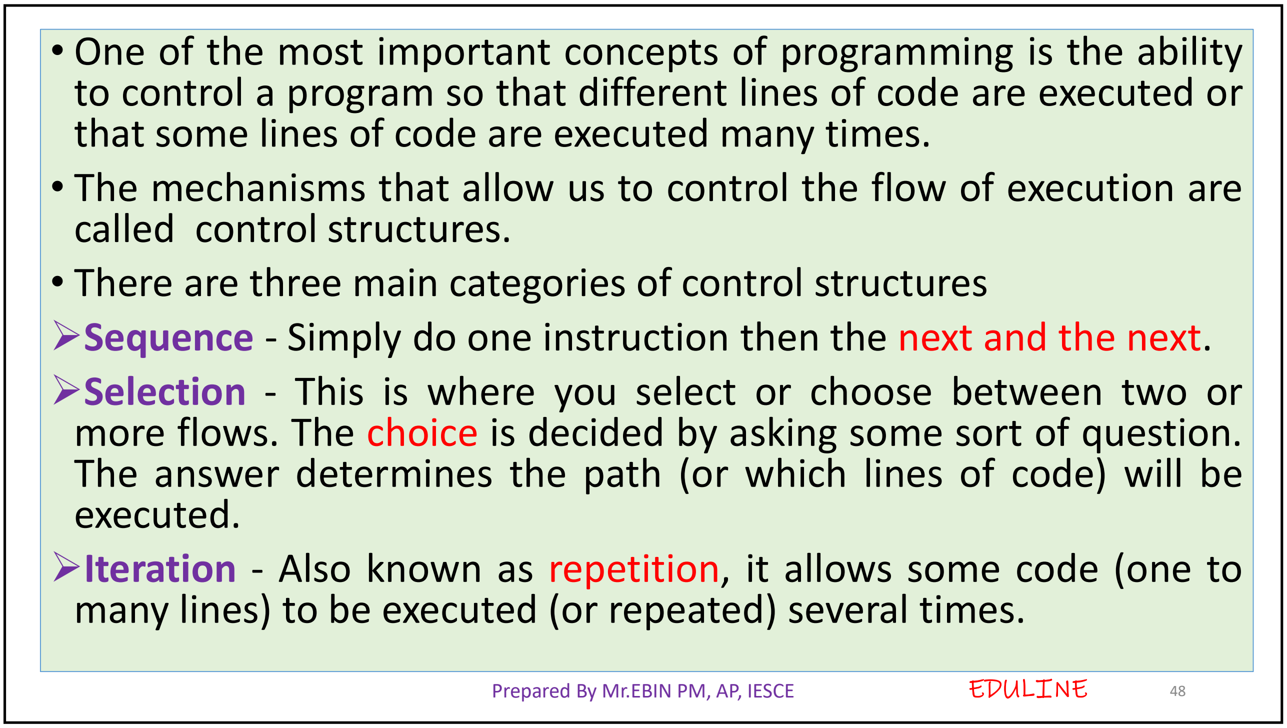
❖Advantages of Structured Programming Approach:

- Easier to read and understand
- User Friendly
- Easier to Maintain
- Mainly problem based instead of being machine based
- Development is easier as it requires less effort and time
- Easier to Debug
- Machine-Independent, mostly.



❖Disadvantages of Structured Programming Approach:

- Since it is Machine-Independent, So it takes time to convert into machine code.
- The converted machine code is not the same as for assembly language.
- The program depends upon changeable factors like data-types. Therefore it needs to be updated with the need on the go.
- Usually the development in this approach takes longer time as it is language-dependent. Whereas in the case of assembly language, the development takes lesser time as it is fixed for the machine.



- One of the most important concepts of programming is the ability to control a program so that different lines of code are executed or that some lines of code are executed many times.
- The mechanisms that allow us to control the flow of execution are called control structures.
- There are three main categories of control structures

➢**Sequence** - Simply do one instruction then the next and the next.

➢**Selection** - This is where you select or choose between two or more flows. The choice is decided by asking some sort of question. The answer determines the path (or which lines of code) will be executed.

➢**Iteration** - Also known as repetition, it allows some code (one to many lines) to be executed (or repeated) several times.

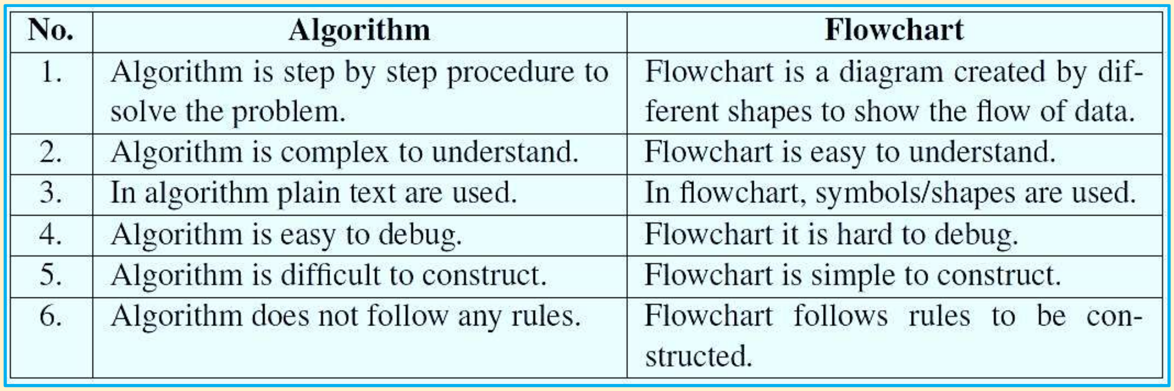
ALGORITHM & FLOWCHART

➢**Algorithm** - An algorithm is the list of instructions and rules that a computer needs to do to complete a task.

- Therefore algorithms are simply a series of instructions that are followed, step by step, to do something useful or solve a problem.

➢**Flow chart** is a graphical representation of an algorithm.

- It is a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing.
- The process of drawing a flowchart for an algorithm is known as flowcharting.



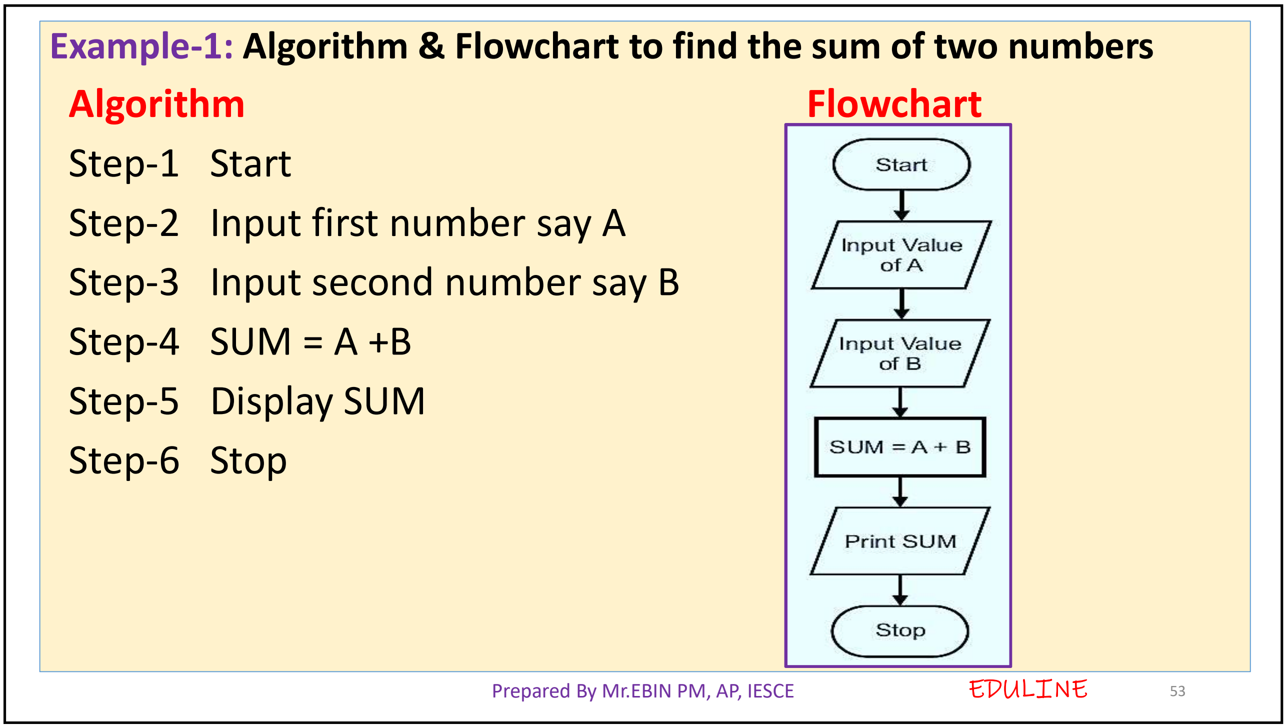
No.	Algorithm	Flowchart
1.	Algorithm is step by step procedure to solve the problem.	Flowchart is a diagram created by different shapes to show the flow of data.
2.	Algorithm is complex to understand.	Flowchart is easy to understand.
3.	In algorithm plain text are used.	In flowchart, symbols/shapes are used.
4.	Algorithm is easy to debug.	Flowchart it is hard to debug.
5.	Algorithm is difficult to construct.	Flowchart is simple to construct.
6.	Algorithm does not follow any rules.	Flowchart follows rules to be constructed.

❖ Algorithm has the following characteristics

- **Input:** An algorithm may or may not require input
- **Output:** Each algorithm is expected to produce at least one result
- **Definiteness:** Each instruction must be clear and unambiguous.
- **Finiteness:** If the instructions of an algorithm are executed, the algorithm should terminate after finite number of steps

Flowchart is diagrammatic/Graphical representation of sequence of steps to solve a problem. To draw a flowchart following standard symbols are use

Symbol Name	Symbol	function
Oval	⬭	Used to represent start and end of flowchart
Parallelogram	▱	Used for input and output operation
Rectangle	▭	Processing: Used for arithmetic operations and data-manipulations
Diamond	◇	Decision making. Used to represent the operation in which there are two/three alternatives, true and false etc
Arrows	←↕→	Flow line Used to indicate the flow of logic by connecting symbols
Circle	○	Page Connector
	⬠	Off Page Connector
	◫	Predefined Process / Function Used to represent a group of statements performing one processing task.
	⬡	Preprocessor
	--[	Comments



Example-1: Algorithm & Flowchart to find the sum of two numbers

Algorithm

```
Step-1   Start
Step-2   Input first number say A
Step-3   Input second number say B
Step-4   SUM = A +B
Step-5   Display SUM
Step-6   Stop
```

Flowchart

Start

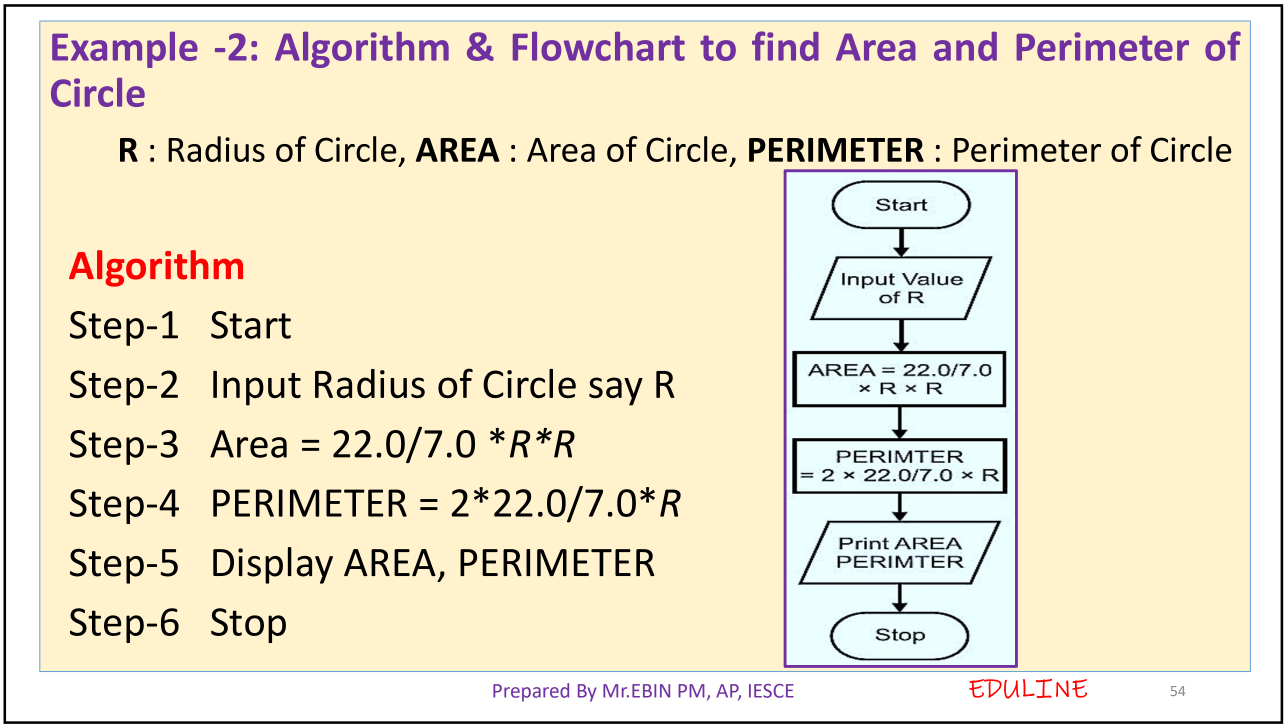
Input Value of A

Input Value of B

SUM = A + B

Print SUM

Stop



Example -2: Algorithm & Flowchart to find Area and Perimeter of Circle

R : Radius of Circle, **AREA** : Area of Circle, **PERIMETER** : Perimeter of Circle

Algorithm

```
Step-1   Start
Step-2   Input Radius of Circle say R
Step-3   Area = 22.0/7.0 *R*R
Step-4   PERIMETER = 2*22.0/7.0*R
Step-5   Display AREA, PERIMETER
Step-6   Stop
```

Start

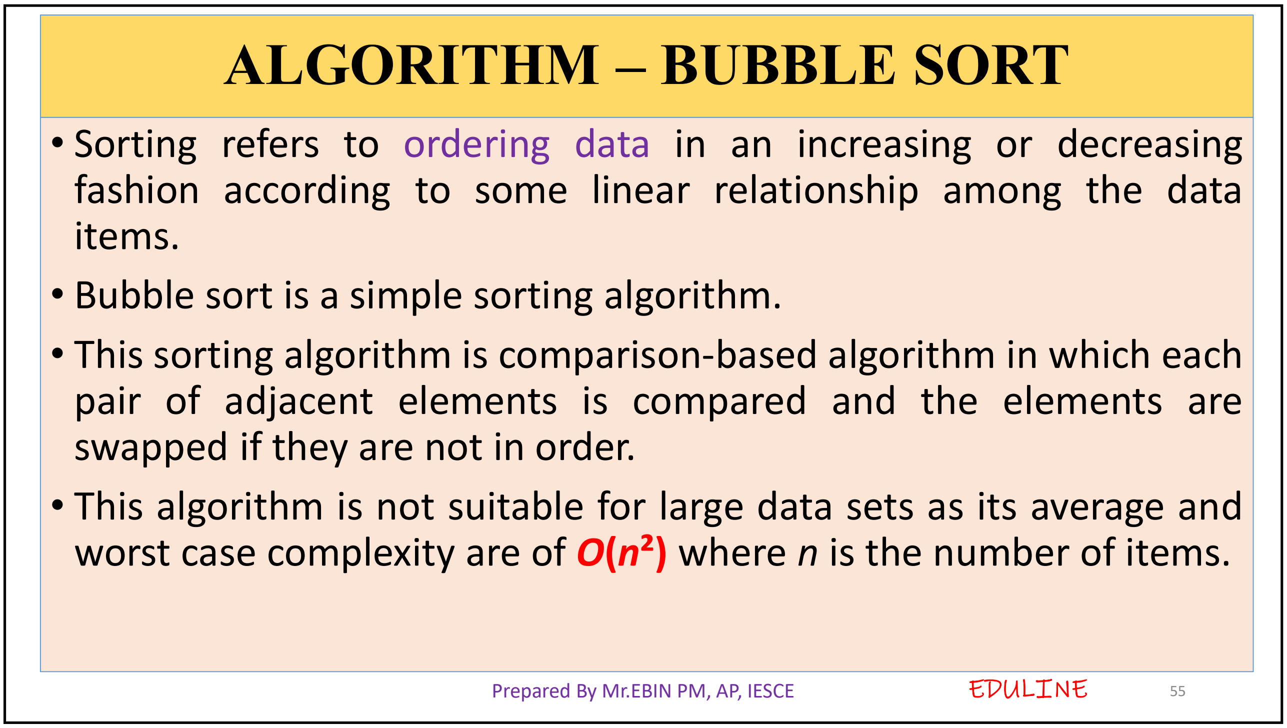
Input Value of R

AREA = 22.0/7.0 × R × R

PERIMTER = 2 × 22.0/7.0 × R

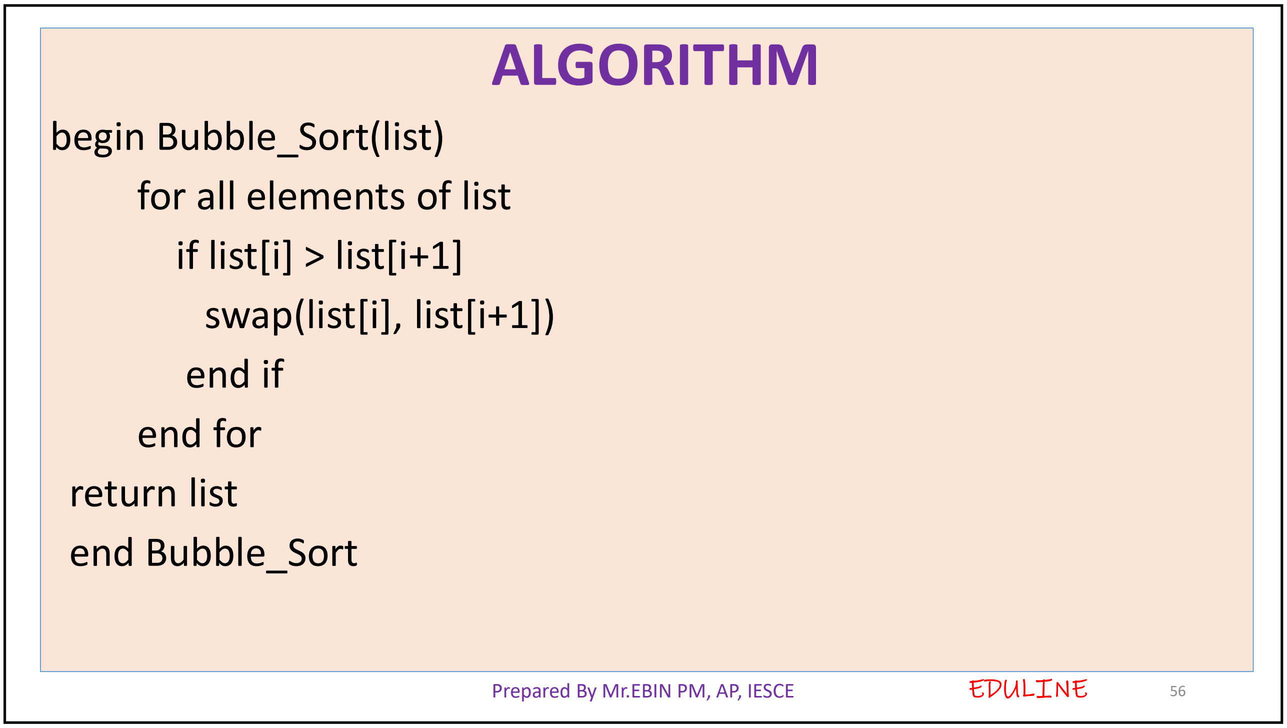
Print AREA PERIMTER

Stop



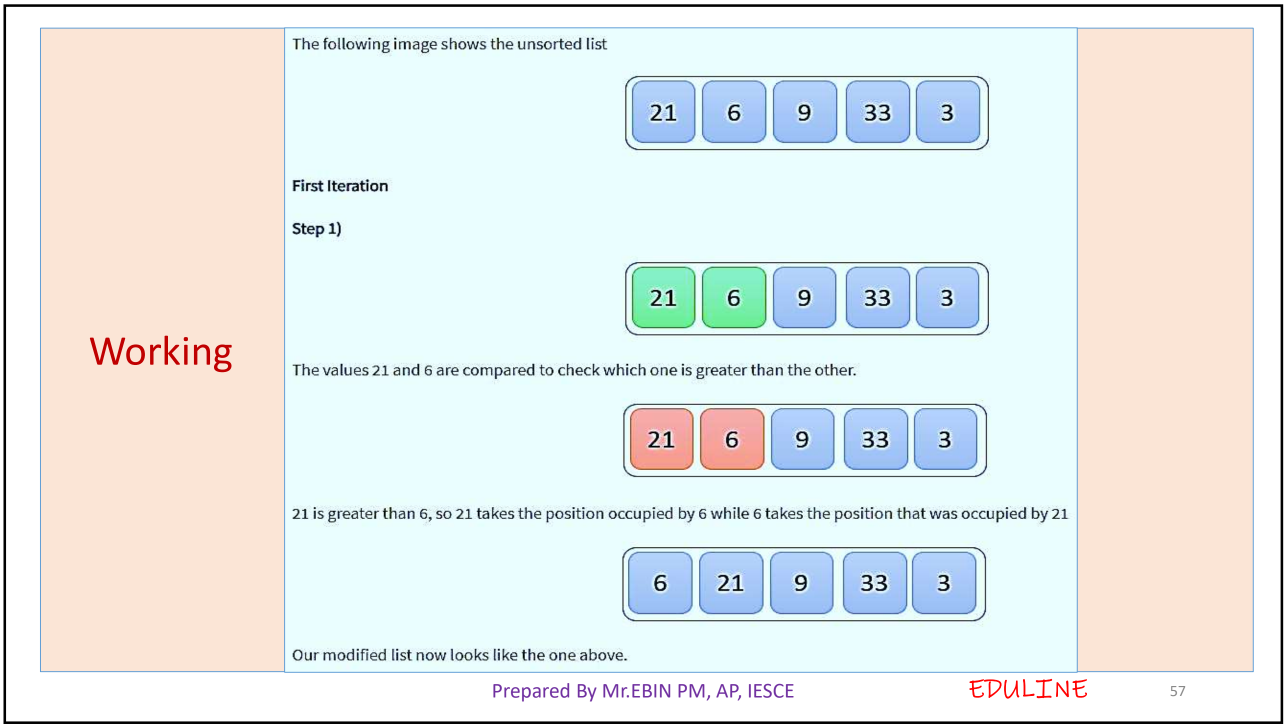
ALGORITHM – BUBBLE SORT

- Sorting refers to ordering data in an increasing or decreasing fashion according to some linear relationship among the data items.
- Bubble sort is a simple sorting algorithm.
- This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.
- This algorithm is not suitable for large data sets as its average and worst case complexity are of ***$O(n^2)$*** where *n* is the number of items.



ALGORITHM

```
begin Bubble_Sort(list)
         for all elements of list
             if list[i] > list[i+1]
                swap(list[i], list[i+1])
              end if
         end for
  return list
  end Bubble_Sort
```



Working

The following image shows the unsorted list

21	6	9	33	3

First Iteration

Step 1)

21	6	9	33	3

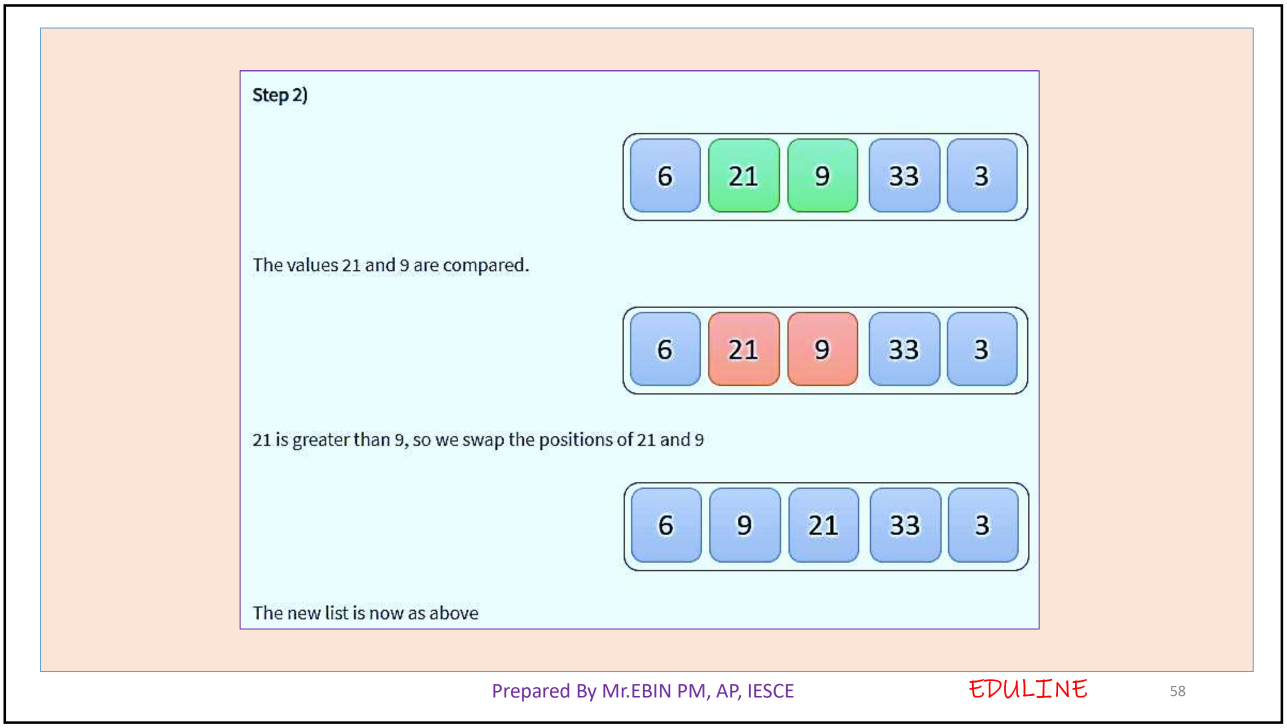
The values 21 and 6 are compared to check which one is greater than the other.

21	6	9	33	3

21 is greater than 6, so 21 takes the position occupied by 6 while 6 takes the position that was occupied by 21

6	21	9	33	3

Our modified list now looks like the one above.



Step 2)

6	21	9	33	3

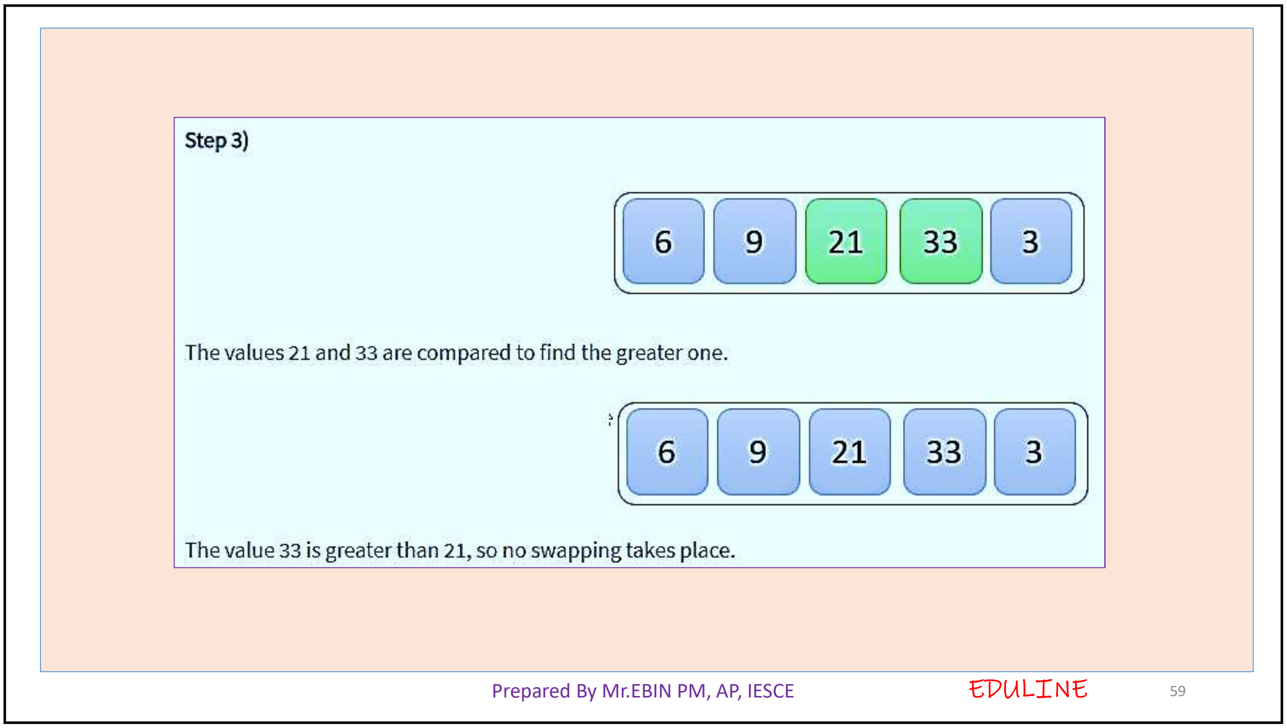
The values 21 and 9 are compared.

6	21	9	33	3

21 is greater than 9, so we swap the positions of 21 and 9

6	9	21	33	3

The new list is now as above



Step 3)

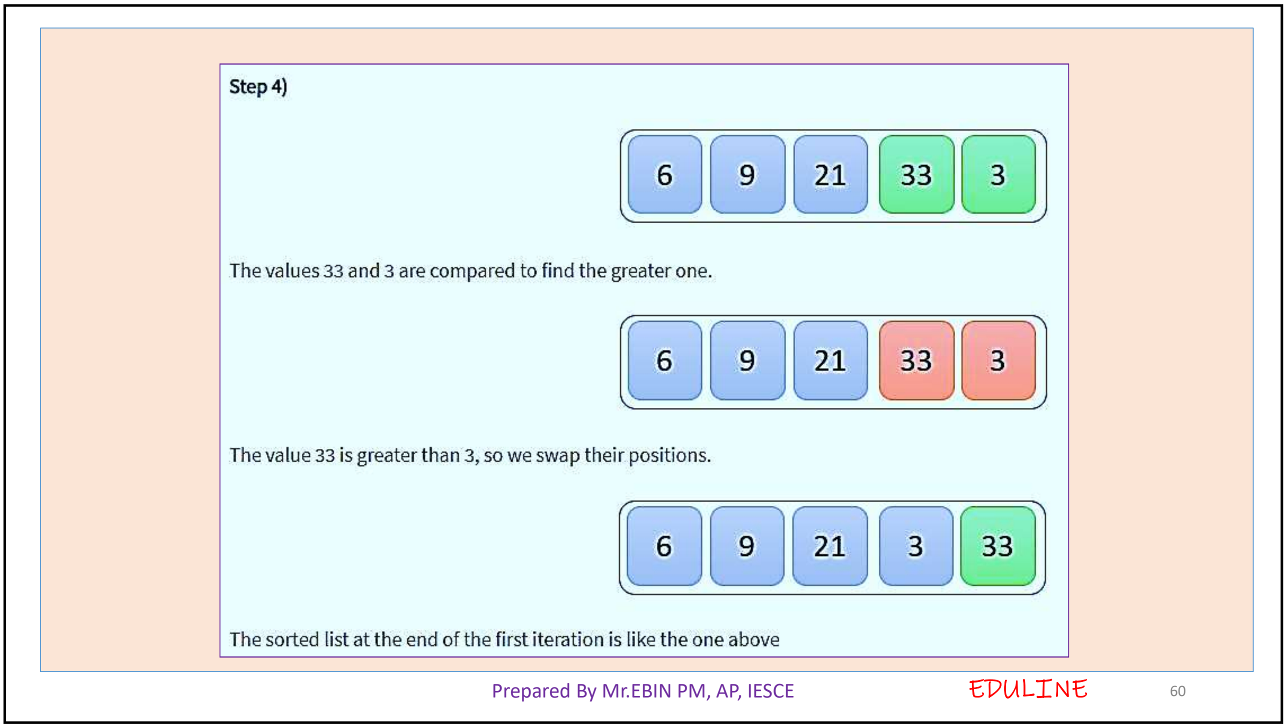
6 9 21 33 3

The values 21 and 33 are compared to find the greater one.

6 9 21 33 3

The value 33 is greater than 21, so no swapping takes place.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 59



Step 4)

6 9 21 33 3

The values 33 and 3 are compared to find the greater one.

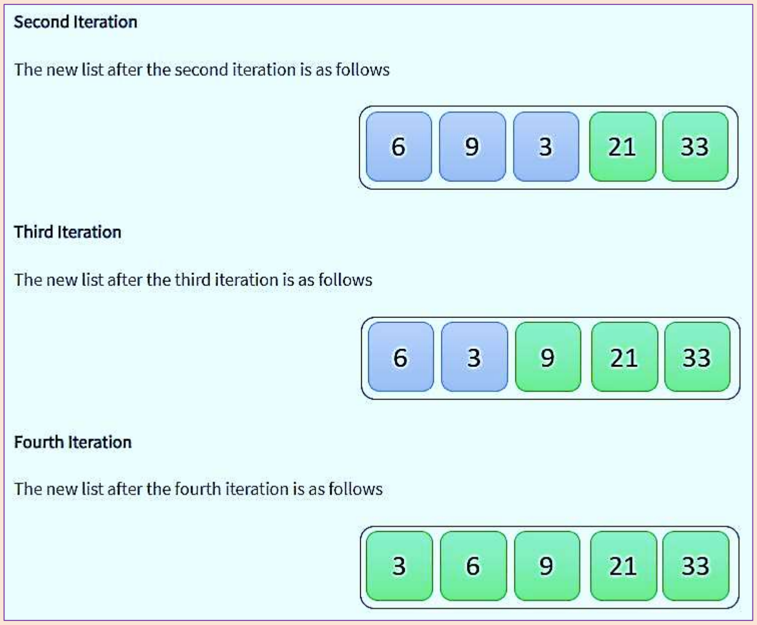
6 9 21 33 3

The value 33 is greater than 3, so we swap their positions.

6 9 21 3 33

The sorted list at the end of the first iteration is like the one above

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 60



Second Iteration

The new list after the second iteration is as follows

6 9 3 21 33

Third Iteration

The new list after the third iteration is as follows

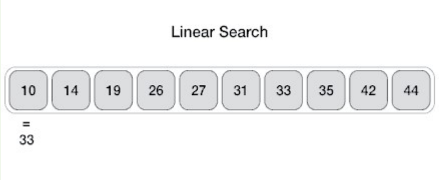
6 3 9 21 33

Fourth Iteration

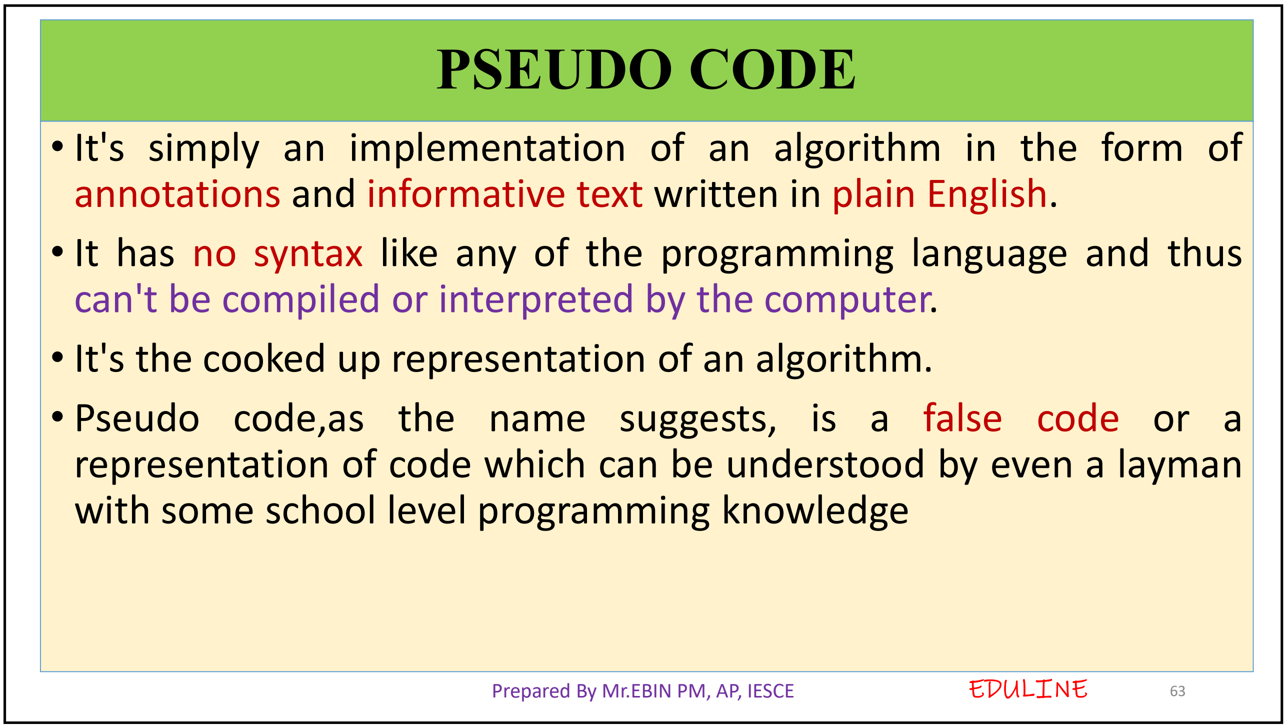
The new list after the fourth iteration is as follows

3 6 9 21 33

ALGORITHM – LINEAR SEARCH

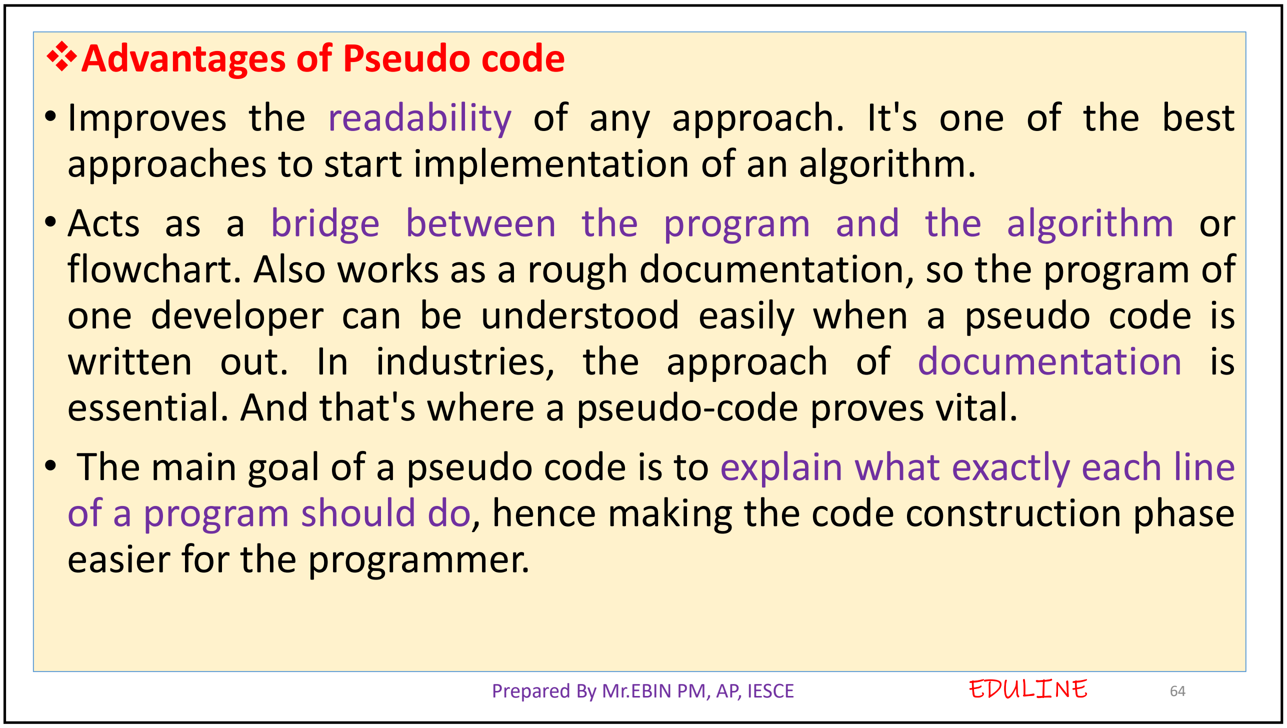


```
Linear Search (Array A, Value x)
Step 1: Set i to 1
Step 2: if i > n then go to step 7
Step 3: if A[i] = x then go to step 6
Step 4: Set i to i+1
Step 5: Go to Step 2
Step 6: Print Element x Found at index i and go to step 8
Step 7: Print element not found
Step 8: Exit
```



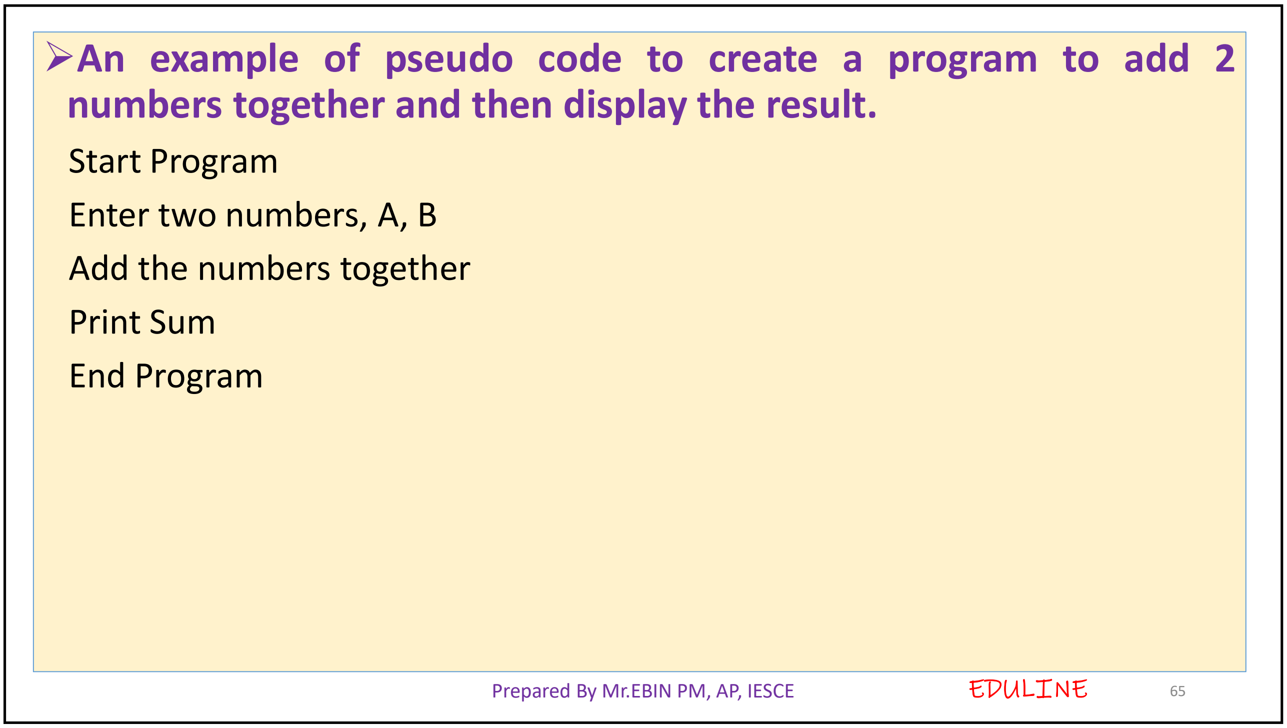
PSEUDO CODE

- It's simply an implementation of an algorithm in the form of annotations and informative text written in plain English.
- It has no syntax like any of the programming language and thus can't be compiled or interpreted by the computer.
- It's the cooked up representation of an algorithm.
- Pseudo code,as the name suggests, is a false code or a representation of code which can be understood by even a layman with some school level programming knowledge



❖Advantages of Pseudo code

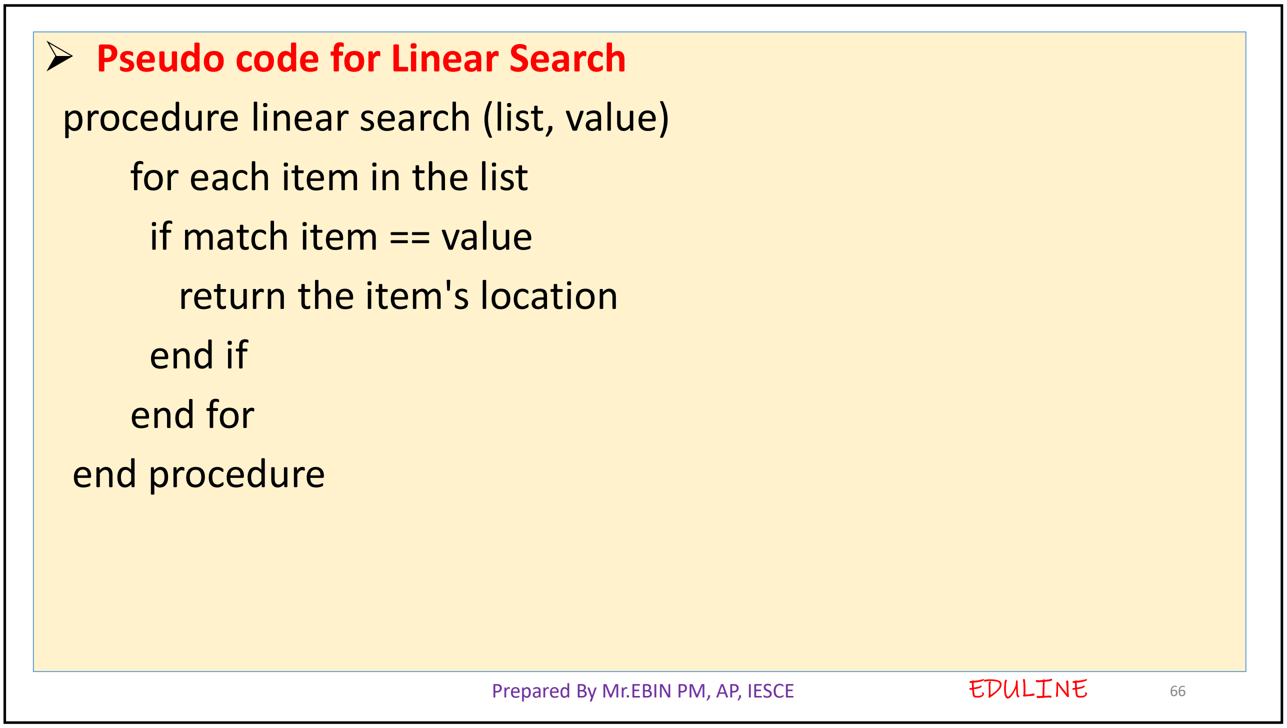
- Improves the readability of any approach. It's one of the best approaches to start implementation of an algorithm.
- Acts as a bridge between the program and the algorithm or flowchart. Also works as a rough documentation, so the program of one developer can be understood easily when a pseudo code is written out. In industries, the approach of documentation is essential. And that's where a pseudo-code proves vital.
- The main goal of a pseudo code is to explain what exactly each line of a program should do, hence making the code construction phase easier for the programmer.



- **An example of pseudo code to create a program to add 2 numbers together and then display the result.**

```
Start Program
Enter two numbers, A, B
Add the numbers together
Print Sum
End Program
```

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 65



- **Pseudo code for Linear Search**

```
procedure linear search (list, value)
      for each item in the list
        if match item == value
           return the item's location
        end if
      end for
 end procedure
```

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 66