IMS Engineering College

NH-09, Adhyatmik Nagar, Near Dasna, Distt. Ghaziabad, U.P.
Tel: (0120) 4940000

Department of MCA

CO-1	Demonstrate the knowledge of the basic structure,components,features and generations of computers
CO-2	Describe the concept of computer languages, language translators and construct algorithms to solve problems using programming concepts.
CO-3	Compare and contrast features, functioning & types of operating system and computer networks
CO-4	Demonstrate architecture, functioning & services of the Internet and basics of multimedia
CO-5	Illustrate the emerging trends and technologies in the field of Information Technology.

Unit- I

INTRODUCTION

The term computer is derived from word ***'compute'*** which means to calculate.

Computer is basically a group of electronic devices that can accept data, conduct a series of **arithmetic and logical operations** on it and give result of these operations as information that is useful to people.

It is a machine that performs pre-defined or programmed computations or control operations that are expressible in numerical and logical form at a **high speed** and with **great accuracy**.

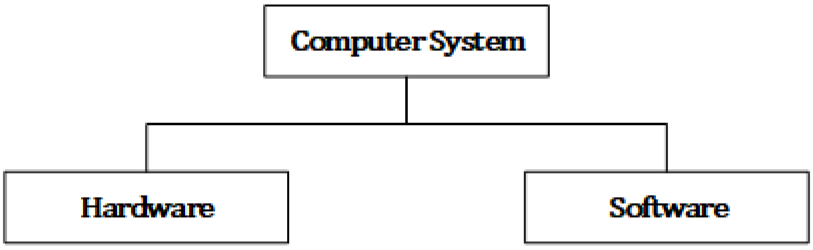
In other words, Computer is an electronic device capable of performing commands and these commands are basically input, output, storage, arithmetic and logical operations.

A computer is a programmable electronic device that accepts raw data and instruction from input device, process data and provides output as information in output device. All the modern computers are digital and represents information in binary digits. They represent data using two distinct values.

In binary system we have only two digits (bits) – 0 and 1 to represent data. Since the bit is a very small unit, virtually, it cannot convey much information while used single. Combination of eight bits is called a byte. A byte is the fundamental unit of data representation in computer. A byte can represent one character of data. Modern computers are capable of processing more than one byte at a time.

COMPUTER SYSTEM

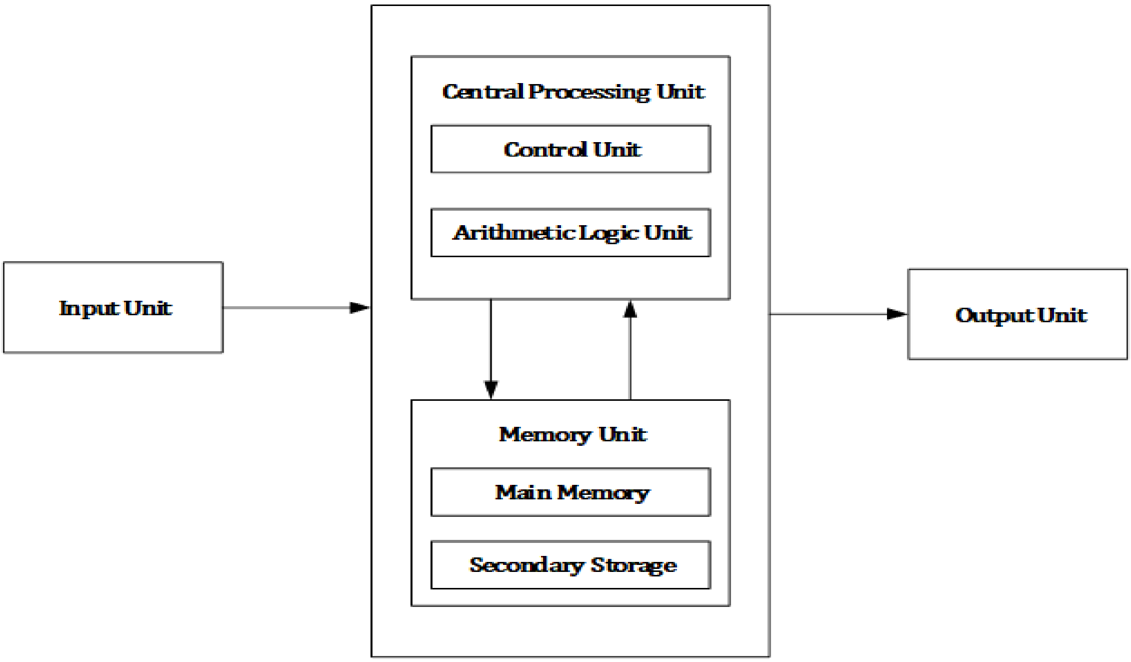
A computer system consists of two major components, namely, **hardware and software**. All physical components that forms computer system are known as **computer hardware**. Software is basically **collection of different programs** that tells computer's hardware what to do.



HARDWARE

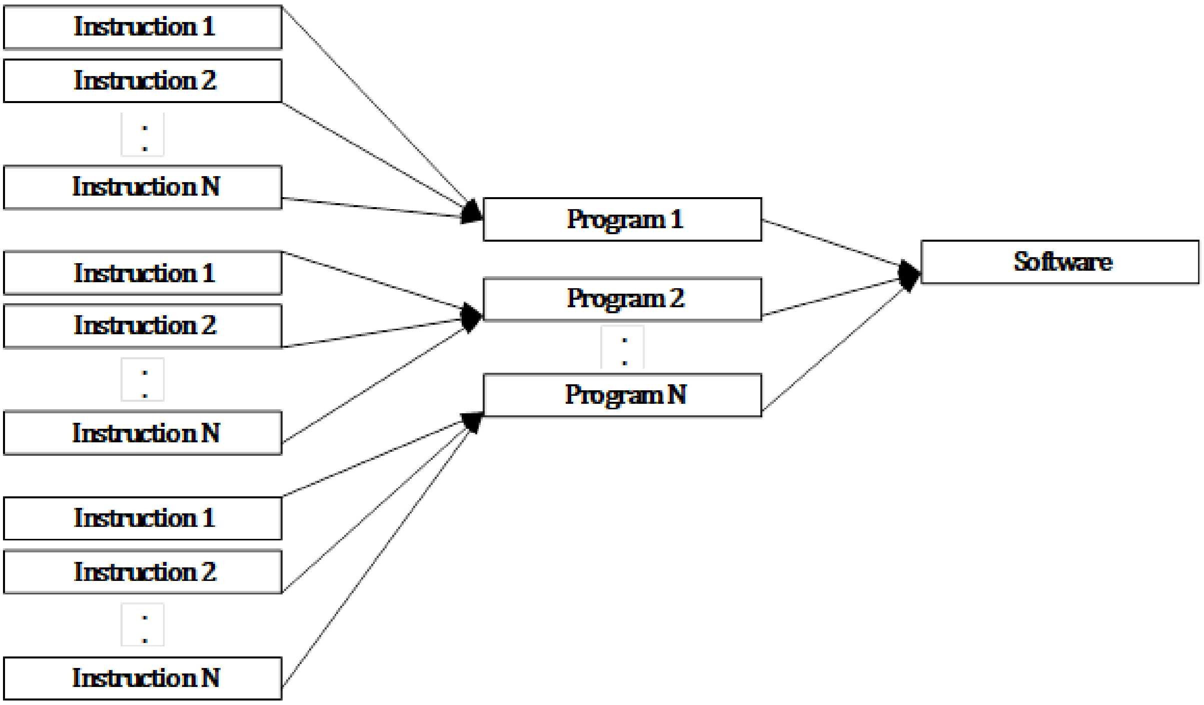
All physical components that make up a computer is known as computer hardware.

It includes all components that we can see and touch i.e. processor, input devices like keyboard, mouse, output devices like visual display unit (VDU), printer, speaker, connecting wires, casing, storage devices etc. Block diagram depicting major components of computer is shown below:



SOFTWARE

Set of instructions that tells the computer hardware what to do is known as computer program. This program or collection of such programs is known as computer software. Concept of software is illustrated in following figure:



Input Devices

Following are some of the important input devices which are used in a computer −

- Keyboard
- Mouse
- Joy Stick
- Light pen
- Track Ball
- Scanner
- Graphic Tablet
- Microphone
- Magnetic Ink Card Reader(MICR)
- Optical Character Reader(OCR)
- Bar Code Reader
- Optical Mark Reader(OMR)

Keyboard

Keyboard is the most common and very popular input device which helps to input data to the computer. The layout of the keyboard is like that of traditional typewriter, although there are some additional keys provided for performing additional functions.

Keyboards are of two sizes 84 keys or 101/102 keys, but now keyboards with 104 keys or 108 keys are also available for Windows and Internet.

The keys on the keyboard are as follows −

S.No	Keys & Description
1	**Typing Keys** These keys include the letter keys (A-Z) and digit keys (09) which generally give the same layout as that of typewriters.
2	**Numeric Keypad** It is used to enter the numeric data or cursor movement. Generally, it consists of a set of 17 keys that are laid out in the same configuration used by most adding machines and calculators.
3	**Function Keys** The twelve function keys are present on the keyboard which are arranged in a row at the top of the keyboard. Each function key has a unique meaning and is used for some specific purpose.
4	**Control keys** These keys provide cursor and screen control. It includes four directional arrow keys. Control keys also include Home, End, Insert, Delete, Page Up, Page Down, Control(Ctrl), Alternate(Alt), Escape(Esc).
5	**Special Purpose Keys** Keyboard also contains some special purpose keys such as Enter, Shift, Caps Lock, Num Lock, Space bar, Tab, and Print Screen.

Mouse

Mouse is the most popular pointing device. It is a very famous cursor-control device having a small palm size box with a round ball at its base, which senses the movement of the mouse and sends corresponding signals to the CPU when the mouse buttons are pressed.

Generally, it has two buttons called the left and the right button and a wheel is present between the buttons. A mouse can be used to control the position of the cursor on the screen, but it cannot be used to enter text into the computer.

Advantages

- Easy to use

- Not very expensive
- Moves the cursor faster than the arrow keys of the keyboard.

Joystick

Joystick is also a pointing device, which is used to move the cursor position on a monitor screen. It is a stick having a spherical ball at its both lower and upper ends. The lower spherical ball moves in a socket. The joystick can be moved in all four directions.

The function of the joystick is similar to that of a mouse. It is mainly used in Computer Aided Designing (CAD) and playing computer games.

Light Pen

Light pen is a pointing device similar to a pen. It is used to select a displayed menu item or draw pictures on the monitor screen. It consists of a photocell and an optical system placed in a small tube.

When the tip of a light pen is moved over the monitor screen and the pen button is pressed, its photocell sensing element detects the screen location and sends the corresponding signal to the CPU.

Track Ball

Track ball is an input device that is mostly used in notebook or laptop computer, instead of a mouse. This is a ball which is half inserted and by moving fingers on the ball, the pointer can be moved.

Since the whole device is not moved, a track ball requires less space than a mouse. A track ball comes in various shapes like a ball, a button, or a square.

Scanner

Scanner is an input device, which works more like a photocopy machine. It is used when some information is available on paper and it is to be transferred to the hard disk of the computer for further manipulation.

Scanner captures images from the source which are then converted into a digital form that can be stored on the disk. These images can be edited before they are printed.

Digitizer

Digitizer is an input device which converts analog information into digital form. Digitizer can convert a signal from the television or camera into a series of numbers that could be stored in a computer. They can be used by the computer to create a picture of whatever the camera had been pointed at.

Digitizer is also known as Tablet or Graphics Tablet as it converts graphics and pictorial data into binary inputs. A graphic tablet as digitizer is used for fine works of drawing and image manipulation applications.

Microphone

Microphone is an input device to input sound that is then stored in a digital form.

The microphone is used for various applications such as adding sound to a multimedia presentation or for mixing music.

Magnetic Ink Card Reader (MICR)

MICR input device is generally used in banks as there are large number of cheques to be processed every day. The bank's code number and cheque number are printed on the cheques with a special type of ink that contains particles of magnetic material that are machine readable.

This reading process is called Magnetic Ink Character Recognition (MICR). The main advantages of MICR is that it is fast and less error prone.

Optical Character Reader (OCR)

OCR is an input device used to read a printed text.

OCR scans the text optically, character by character, converts them into a machine readable code, and stores the text on the system memory.

Bar Code Readers

Bar Code Reader is a device used for reading bar coded data (data in the form of light and dark lines). Bar coded data is generally used in labelling goods, numbering the books, etc. It may be a handheld scanner or may be embedded in a stationary scanner.

Bar Code Reader scans a bar code image, converts it into an alphanumeric value, which is then fed to the computer that the bar code reader is connected to.

Optical Mark Reader (OMR)

OMR is a special type of optical scanner used to recognize the type of mark made by pen or pencil. It is used where one out of a few alternatives is to be selected and marked.

It is specially used for checking the answer sheets of examinations having multiple choice questions.

Output Devices

Following are some of the important output devices used in a computer.

- Monitors
- Graphic Plotter
- Printer

Monitors

Monitors, commonly called as **Visual Display Unit** (VDU), are the main output device of a computer. It forms images from tiny dots, called pixels that are arranged in a rectangular form. The sharpness of the image depends upon the number of pixels.

There are two kinds of viewing screen used for monitors.

- Cathode-Ray Tube (CRT)
- Flat-Panel Display

Cathode-Ray Tube (CRT) Monitor

The CRT display is made up of small picture elements called pixels. The smaller the pixels, the better the image clarity or resolution. It takes more than one illuminated pixel to form a whole character, such as the letter 'e' in the word help.

A finite number of characters can be displayed on a screen at once. The screen can be divided into a series of character boxes - fixed location on the screen where a standard character can be placed. Most screens are capable of displaying 80 characters of data horizontally and 25 lines vertically.

There are some disadvantages of CRT −

- Large in Size
- High power consumption

Flat-Panel Display Monitor

The flat-panel display refers to a class of video devices that have reduced volume, weight and power requirement in comparison to the CRT. You can hang them on walls or wear them on your wrists. Current uses of flat-panel displays include calculators, video games, monitors, laptop computer, and graphics display.

The flat-panel display is divided into two categories −

- **Emissive Displays** − Emissive displays are devices that convert electrical energy into light. For example, plasma panel and LED (Light-Emitting Diodes).
- **Non-Emissive Displays** − Non-emissive displays use optical effects to convert sunlight or light from some other source into graphics patterns. For example, LCD (Liquid-Crystal Device).

Printers

Printer is an output device, which is used to print information on paper.

There are two types of printers −

- Impact Printers
- Non-Impact Printers

Impact Printers

Impact printers print the characters by striking them on the ribbon, which is then pressed on the paper.

Characteristics of Impact Printers are the following −

- Very low consumable costs
- Very noisy
- Useful for bulk printing due to low cost
- There is physical contact with the paper to produce an image

These printers are of two types −

- Character printers
- Line printers

Character Printers

Character printers are the printers which print one character at a time.

These are further divided into two types:

- Dot Matrix Printer(DMP)
- Daisy Wheel

Dot Matrix Printer

In the market, one of the most popular printers is Dot Matrix Printer. These printers are popular because of their ease of printing and economical price. Each character printed is in the form of pattern of dots and head consists of a Matrix of Pins of size (5*7, 7*9, 9*7 or 9*9) which come out to form a character which is why it is called Dot Matrix Printer.

Advantages

- Inexpensive
- Widely Used
- Other language characters can be printed

Disadvantages

- Slow Speed
- Poor Quality

Daisy Wheel

Head is lying on a wheel and pins corresponding to characters are like petals of Daisy (flower) which is why it is called Daisy Wheel Printer. These printers are generally used for word-processing in offices that require a few letters to be sent here and there with very nice quality.

Advantages

- More reliable than DMP
- Better quality
- Fonts of character can be easily changed

Disadvantages

- Slower than DMP
- Noisy
- More expensive than DMP

Line Printers

Line printers are the printers which print one line at a time.

These are of two types −

- Drum Printer
- Chain Printer

Drum Printer

This printer is like a drum in shape hence it is called drum printer. The surface of the drum is divided into a number of tracks. Total tracks are equal to the size of the paper, i.e. for a paper width of 132 characters, drum will have 132 tracks. A character set is embossed on the track. Different character sets available in the market are 48 character set, 64 and 96 characters set. One rotation of drum prints one line. Drum printers are fast in speed and can print 300 to 2000 lines per minute.

Advantages

- Very high speed

Disadvantages

- Very expensive
- Characters fonts cannot be changed

Chain Printer

In this printer, a chain of character sets is used, hence it is called Chain Printer. A standard character set may have 48, 64, or 96 characters.

Advantages

- Character fonts can easily be changed.
- Different languages can be used with the same printer.

Disadvantages

- Noisy

Non-impact Printers

Non-impact printers print the characters without using the ribbon. These printers print a complete page at a time, thus they are also called as Page Printers.

These printers are of two types −

- Laser Printers
- Inkjet Printers

Characteristics of Non-impact Printers

- Faster than impact printers
- They are not noisy
- High quality
- Supports many fonts and different character size

Laser Printers

These are non-impact page printers. They use laser lights to produce the dots needed to form the characters to be printed on a page.

Advantages

- Very high speed
- Very high quality output
- Good graphics quality
- Supports many fonts and different character size

Disadvantages

- Expensive
- Cannot be used to produce multiple copies of a document in a single printing

Inkjet Printers

Inkjet printers are non-impact character printers based on a relatively new technology. They print characters by spraying small drops of ink onto paper. Inkjet printers produce high quality output with presentable features.

They make less noise because no hammering is done and these have many styles of printing modes available. Color printing is also possible. Some models of Inkjet printers can produce multiple copies of printing also.

Advantages

- High quality printing

- More reliable

Disadvantages

- Expensive as the cost per page is high
- Slow as compared to laser printer

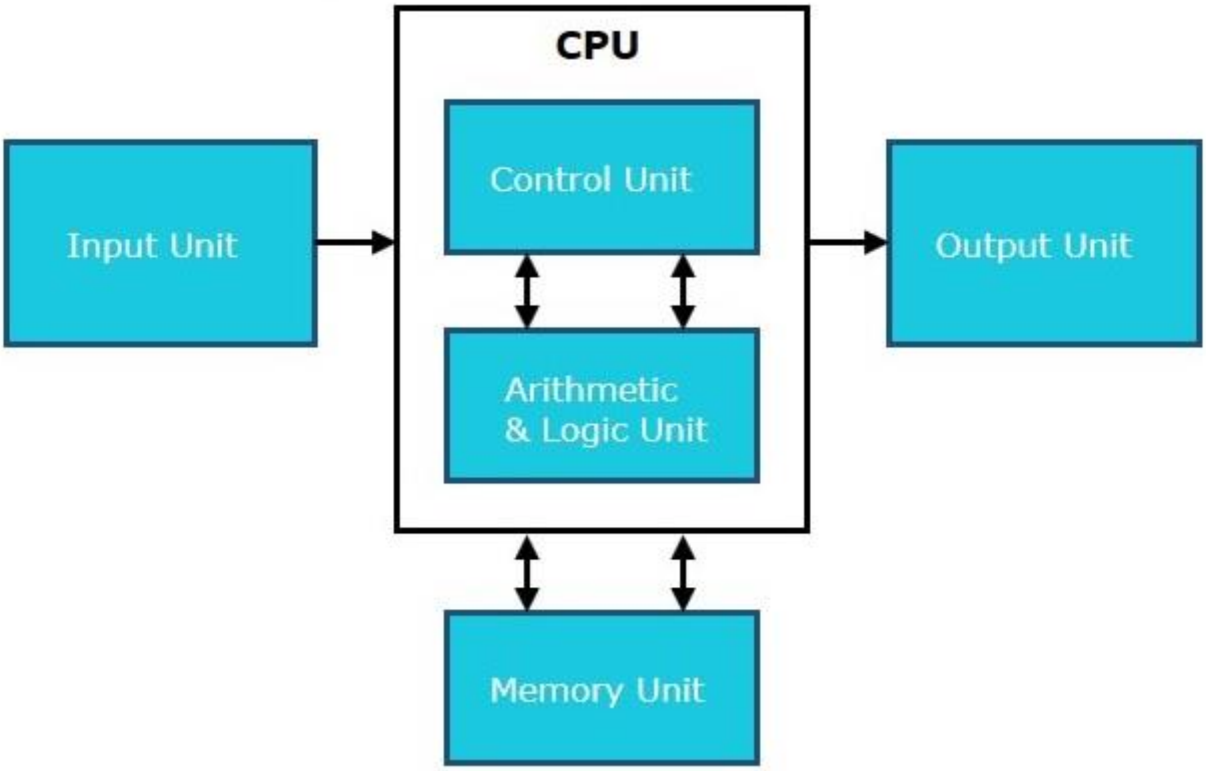
CPU (Central Processing Unit)

Central Processing Unit (CPU) consists of the following features −

- CPU is considered as the brain of the computer.
- CPU performs all types of data processing operations.
- It stores data, intermediate results, and instructions (program).
- It controls the operation of all parts of the computer.

CPU itself has following three components.

- Memory or Storage Unit
- Control Unit
- ALU(Arithmetic Logic Unit)



Memory or Storage Unit

This unit can store instructions, data, and intermediate results. This unit supplies information to other units of the computer when needed. It is also known as internal storage unit or the main memory or the primary storage or Random Access Memory (RAM).

Its size affects speed, power, and capability. Primary memory and secondary memory are two types of memories in the computer. Functions of the memory unit are −

- It stores all the data and the instructions required for processing.
- It stores intermediate results of processing.
- It stores the final results of processing before these results are released to an output device.
- All inputs and outputs are transmitted through the main memory.

Control Unit

This unit controls the operations of all parts of the computer but does not carry out any actual data processing operations.

Functions of this unit are −

- It is responsible for controlling the transfer of data and instructions among other units of a computer.
- It manages and coordinates all the units of the computer.
- It obtains the instructions from the memory, interprets them, and directs the operation of the computer.
- It communicates with Input/Output devices for transfer of data or results from storage.
- It does not process or store data.

ALU (Arithmetic Logic Unit)

This unit consists of two subsections namely,

- Arithmetic Section
- Logic Section

Arithmetic Section

Function of arithmetic section is to perform arithmetic operations like addition, subtraction, multiplication, and division. All complex operations are done by making repetitive use of the above operations.

Logic Section

Function of logic section is to perform logic operations such as comparing, selecting, matching, and merging of data.

Memory

A memory is just like a human brain. It is used to store data and instructions. Computer memory is the storage space in the computer, where data is to be processed and instructions required for processing are stored. The memory is divided into large number of small parts called cells. Each location or cell has a unique address, which varies from zero to memory size minus one. For example, if the computer has 64k words, then this memory unit has 64 * 1024 = 65536 memory locations. The address of these locations varies from 0 to 65535.

Memory is primarily of three types −

- Cache Memory
- Primary Memory/Main Memory
- Secondary Memory

Cache Memory

Cache memory is a very high speed semiconductor memory which can speed up the CPU. It acts as a buffer between the CPU and the main memory. It is used to hold those parts of data and program which are most frequently used by the CPU. The parts of data and programs are transferred from the disk to cache memory by the operating system, from where the CPU can access them.

Advantages

The advantages of cache memory are as follows −

- Cache memory is faster than main memory.
- It consumes less access time as compared to main memory.
- It stores the program that can be executed within a short period of time.
- It stores data for temporary use.

Disadvantages

The disadvantages of cache memory are as follows −

- Cache memory has limited capacity.
- It is very expensive.

Primary Memory (Main Memory)

Primary memory holds only those data and instructions on which the computer is currently working. It has a limited capacity and data is lost when power is switched off. It is generally made up of semiconductor device. These memories are not as fast as registers. The data and instruction required to be processed resides in the main memory. It is divided into two subcategories RAM and ROM.

Characteristics of Main Memory

- These are semiconductor memories.
- It is known as the main memory.
- Usually volatile memory.
- Data is lost in case power is switched off.
- It is the working memory of the computer.
- Faster than secondary memories.
- A computer cannot run without the primary memory.

Secondary Memory

This type of memory is also known as external memory or non-volatile. It is slower than the main memory. These are used for storing data/information permanently. CPU directly does not access these memories, instead they are accessed via input-output routines. The contents of secondary memories are first transferred to the main memory, and then the CPU can access it. For example, disk, CD-ROM, DVD, etc.

Characteristics of Secondary Memory

- These are magnetic and optical memories.
- It is known as the backup memory.
- It is a non-volatile memory.
- Data is permanently stored even if power is switched off.
- It is used for storage of data in a computer.
- Computer may run without the secondary memory.
- Slower than primary memories.

Software

Software is a set of programs, which is designed to perform a well-defined function. A program is a sequence of instructions written to solve a particular problem.

There are two types of software −

- System Software
- Application Software

System Software

The system software is a collection of programs designed to operate, control, and extend the processing capabilities of the computer itself. System software is generally prepared by the computer manufacturers. These software products comprise of programs written in low-level languages, which interact with the hardware at a very basic level. System software serves as the interface between the hardware and the end users.

Some examples of system software are Operating System, Compilers, Interpreter, Assemblers, etc.

Here is a list of some of the most prominent features of a system software −

- Close to the system
- Fast in speed
- Difficult to design
- Difficult to understand
- Less interactive
- Smaller in size

- Difficult to manipulate
- Generally written in low-level language

Application Software

Application software products are designed to satisfy a particular need of a particular environment. All software applications prepared in the computer lab can come under the category of Application software.

Application software may consist of a single program, such as Microsoft's notepad for writing and editing a simple text. It may also consist of a collection of programs, often called a software package, which work together to accomplish a task, such as a spreadsheet package.

Examples of Application software are the following −

- Payroll Software
- Student Record Software
- Inventory Management Software
- Income Tax Software
- Railways Reservation Software
- Microsoft Office Suite Software
- Microsoft Word
- Microsoft Excel
- Microsoft PowerPoint

Features of application software are as follows −

- Close to the user
- Easy to design
- More interactive
- Slow in speed
- Generally written in high-level language
- Easy to understand
- Easy to manipulate and use
- Bigger in size and requires large storage space

Language Processors: Assembler, Compiler and Interpreter

Language Processors –

Compilers, interpreters, translate programs written in high-level languages into machine code that a computer understands. And assemblers translate programs written in low-level or assembly language into machine code. In the compilation process, there are several stages. To help programmers write error-free code, tools are available.

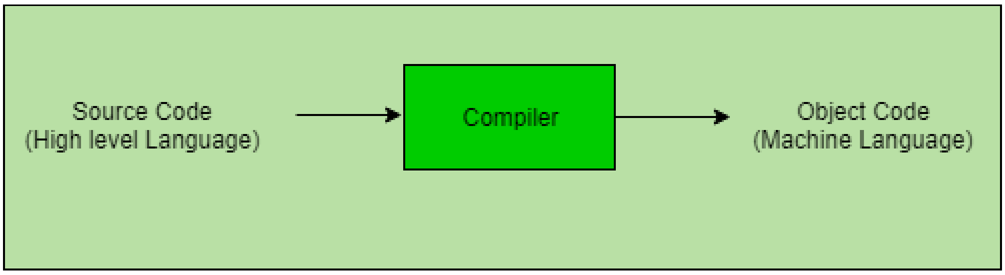
Assembly language is machine-dependent, yet mnemonics used to represent instructions in it are not directly understandable by machine and high-Level language is machine-independent. A computer understands instructions in machine code, i.e. in the form of 0s and 1s. It is a tedious task to write a computer program directly in machine code. The programs are written mostly in high-level languages like Java, C++, Python etc. and are called source code. These source code cannot be executed directly by the computer and must be converted into machine language to be executed. Hence, a special translator system software is used to translate the program written in a high-level language into machine code is called Language Processor and the program after translated into machine code (object program/object code).

The language processors can be any of the following three types:

1. Compiler :

The language processor that reads the complete source program written in high-level language as a whole in one go and translates it into an equivalent program in machine language is called a Compiler. Example: C, C++, C#.

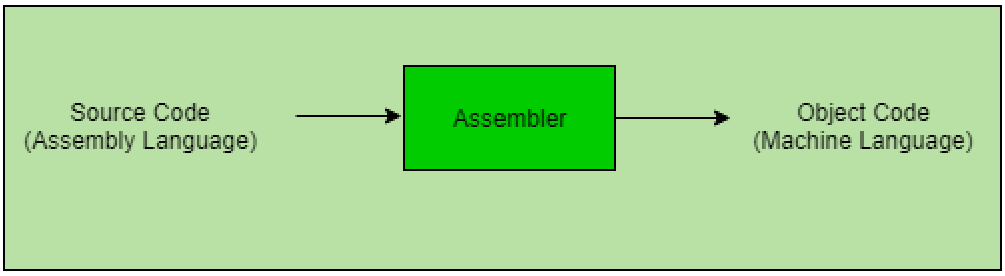
In a compiler, the source code is translated to object code successfully if it is free of errors. The compiler specifies the errors at the end of the compilation with line numbers when there are any errors in the source code. The errors must be removed before the compiler can successfully recompile the source code again the object program can be executed number of times without translating it again.



2. Assembler :

The Assembler is used to translate the program written in Assembly language into machine code. The source program is an input of an assembler that contains assembly language instructions. The output generated by the assembler is the object code or machine code understandable by the computer. Assembler is basically the 1st interface that is able to communicate humans with the machine. We need an Assembler to fill the gap between human and machine so that they can communicate with each other. code written in assembly language is some sort of mnemonics(instructions) like ADD, MUL, MUX, SUB, DIV, MOV and so on. and the assembler is basically able to convert these mnemonics in Binary code. Here, these mnemonics also depend upon the architecture of the machine.

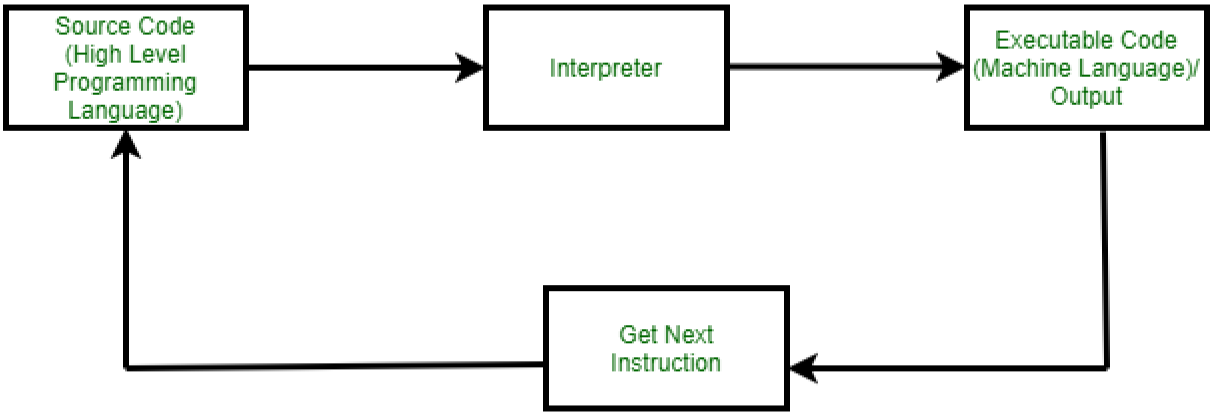
For example, the architecture of intel 8085 and intel 8086 are different.



3. Interpreter :

The translation of a single statement of the source program into machine code is done by a language processor and executes immediately before moving on to the next line is called an interpreter. If there is an error in the statement, the interpreter terminates its translating process at that statement and displays an error message. The interpreter moves on to the next line for execution only after the removal of the error. An Interpreter directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code. An interpreter translates one line at a time and then executes it.

Example: Perl, Python and Matlab.

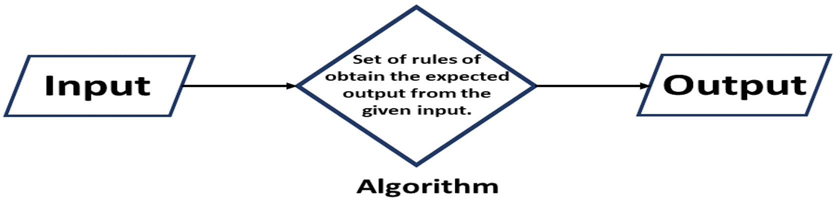


Difference between Compiler and Interpreter –

Compiler	Interpreter
A compiler is a program that converts the entire source code of a programming language into executable machine code for a CPU.	An interpreter takes a source program and runs it line by line, translating each line as it comes to it
The compiler takes a large amount of time to analyze the entire source code but the overall execution time of the program is comparatively faster.	An interpreter takes less amount of time to analyze the source code but the overall execution time of the program is slower.
The compiler generates the error message only after scanning the whole program, so debugging is comparatively hard as the error can be present anywhere in the program.	Its Debugging is easier as it continues translating the program until the error is met.
The compiler requires a lot of memory for generating object codes.	It requires less memory than a compiler because no object code is generated.
Generates intermediate object code.	No intermediate object code is generated.
For Security purpose compiler is more useful.	The interpreter is a little vulnerable in case of security.
Examples: C, C++, C#	Examples: Python, Perl, JavaScript, Ruby

What is an Algorithm?

- An algorithm is a set of commands that must be followed for a computer to perform calculations or other problem-solving operations.
- According to its formal definition, an algorithm is a finite set of instructions carried out in a specific order to perform a particular task.
- It is not the entire program or code; it is simple logic to a problem represented as an informal description in the form of a flowchart or pseudocode.


Input
Set of rules of obtain the expected output from the given input.
Algorithm
Output

- Problem: A problem can be defined as a real-world problem or real-world instance problem for which you need to develop a program or set of instructions. An algorithm is a set of instructions.

- Algorithm: An algorithm is defined as a step-by-step process that will be designed for a problem.
- Input: After designing an algorithm, the algorithm is given the necessary and desired inputs.
- Processing unit: The input will be passed to the processing unit, producing the desired output.
- Output: The outcome or result of the program is referred to as the output.

After defining what an algorithm is, you will now look at algorithm characteristics.

How do Algorithms Work?

Algorithms are step-by-step procedures designed to solve specific problems and perform tasks efficiently in the realm of computer science and mathematics. These powerful sets of instructions form the backbone of modern technology and govern everything from web searches to artificial intelligence. Here's how algorithms work:

- Input: Algorithms take input data, which can be in various formats, such as numbers, text, or images.
- Processing: The algorithm processes the input data through a series of logical and mathematical operations, manipulating and transforming it as needed.
- Output: After the processing is complete, the algorithm produces an output, which could be a result, a decision, or some other meaningful information.
- Efficiency: A key aspect of algorithms is their efficiency, aiming to accomplish tasks quickly and with minimal resources.
- Optimization: Algorithm designers constantly seek ways to optimize their algorithms, making them faster and more reliable.
- Implementation: Algorithms are implemented in various programming languages, enabling computers to execute them and produce desired outcomes.

What is the Need for Algorithms?

You require algorithms for the following reasons:

Scalability

It aids in your understanding of scalability. When you have a sizable real-world problem, you must break it down into small steps to analyze it quickly.

Performance

The real world is challenging to break down into smaller steps. If a problem can be easily divided into smaller steps, it indicates that the problem is feasible.

After understanding what is an algorithm, why you need an algorithm, you will look at how to write one using an example.

Types of Algorithms

1. **Brute Force Algorithm**: A straightforward approach that exhaustively tries all possible solutions, suitable for small problem instances but may become impractical for larger ones due to its high time complexity.

2. **Recursive Algorithm**: A method that breaks a problem into smaller, similar subproblems and repeatedly applies itself to solve them until reaching a base case, making it effective for tasks with recursive structures.

3. **Encryption Algorithm**: Utilized to transform data into a secure, unreadable form using cryptographic techniques, ensuring confidentiality and privacy in digital communications and transactions.

4. **Backtracking Algorithm**: A trial-and-error technique used to explore potential solutions by undoing choices when they lead to an incorrect outcome, commonly employed in puzzles and optimization problems.

5. **Searching Algorithm**: Designed to find a specific target within a dataset, enabling efficient retrieval of information from sorted or unsorted collections.

6. **Sorting Algorithm**: Aimed at arranging elements in a specific order, like numerical or alphabetical, to enhance data organization and retrieval.

7. **Hashing Algorithm**: Converts data into a fixed-size hash value, enabling rapid data access and retrieval in hash tables, commonly used in databases and password storage.

8. **Divide and Conquer Algorithm**: Breaks a complex problem into smaller subproblems, solves them independently, and then combines their solutions to address the original problem effectively.

9. **Greedy Algorithm**: Makes locally optimal choices at each step in the hope of finding a global optimum, useful for optimization problems but may not always lead to the best solution.

10. **Dynamic Programming Algorithm**: Stores and reuses intermediate results to avoid redundant computations, enhancing the efficiency of solving complex problems.

11. **Randomized Algorithm:** Utilizes randomness in its steps to achieve a solution, often used in situations where an approximate or probabilistic answer suffices.

How to Write an Algorithm?

- There are no well-defined standards for writing algorithms. It is, however, a problem that is resource-dependent. Algorithms are never written with a specific programming language in mind.
- As you all know, basic code constructs such as loops like do, for, while, all programming languages share flow control such as if-else, and so on. An algorithm can be written using these common constructs.

- Algorithms are typically written in a step-by-step fashion, but this is not always the case. Algorithm writing is a process that occurs after the problem domain has been well-defined. That is, you must be aware of the problem domain for which you are developing a solution.

Example

Now, use an example to learn how to write algorithms.

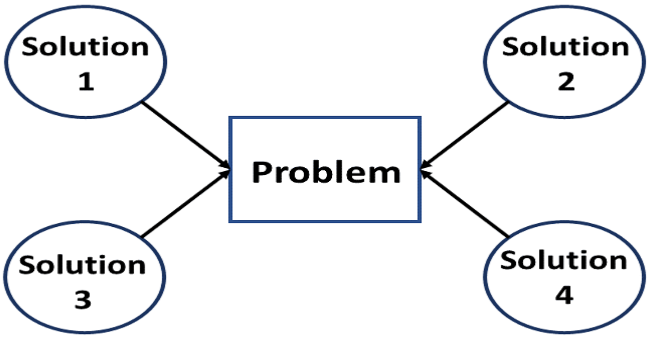
Problem: Create an algorithm that multiplies two numbers and displays the output.

```
Step 1 − Start
Step 2 − declare three integers x, y & z
Step 3 − define values of x & y
Step 4 − multiply values of x & y
Step 5 − store result of step 4 to z
Step 6 − print z
Step 7 − Stop
```

Algorithms instruct programmers on how to write code. In addition, the algorithm can be written as:

```
Step 1 − Start mul
Step 2 − get values of x & y
Step 3 − z ← x * y
Step 4 − display z
Step 5 − Stop
```

In algorithm design and analysis, the second method is typically used to describe an algorithm. It allows the analyst to analyze the algorithm while ignoring all unwanted definitions easily. They can see which operations are being used and how the process is progressing. It is optional to write step numbers. To solve a given problem, you create an algorithm. A problem can be solved in a variety of ways.



As a result, many solution algorithms for a given problem can be derived. The following step is to evaluate the proposed solution algorithms and implement the most appropriate solution.

As you progress through this "what is an Algorithm" tutorial, you will learn about some of the components of an algorithm.

Factors of an Algorithm

The following are the factors to consider when designing an algorithm:

- **Modularity**: This feature was perfectly designed for the algorithm if you are given a problem and break it down into small-small modules or small-small steps, which is a basic definition of an algorithm.
- **Correctness**: An algorithm's correctness is defined as when the given inputs produce the desired output, indicating that the algorithm was designed correctly. An algorithm's analysis has been completed correctly.
- **Maintainability**: It means that the algorithm should be designed in a straightforward, structured way so that when you redefine the algorithm, no significant changes are made to the algorithm.
- **Functionality**: It takes into account various logical steps to solve a real-world problem.
- **Robustness**: Robustness refers to an algorithm's ability to define your problem clearly.
- **User-friendly**: If the algorithm is difficult to understand, the designer will not explain it to the programmer.
- **Simplicity:** If an algorithm is simple, it is simple to understand.
- **Extensibility**: Your algorithm should be extensible if another algorithm designer or programmer wants to use it.

You will now see why an algorithm is so essential after understanding some of its components.

Qualities of a Good Algorithm

- **Efficiency**: A good algorithm should perform its task quickly and use minimal resources.

- **Correctness**: It must produce the correct and accurate output for all valid inputs.
- **Clarity**: The algorithm should be easy to understand and comprehend, making it maintainable and modifiable.
- **Scalability**: It should handle larger data sets and problem sizes without a significant decrease in performance.
- **Reliability**: The algorithm should consistently deliver correct results under different conditions and environments.
- **Optimality**: Striving for the most efficient solution within the given problem constraints.
- **Robustness**: Capable of handling unexpected inputs or errors gracefully without crashing.
- **Adaptability**: Ideally, it can be applied to a range of related problems with minimal adjustments.
- **Simplicity**: Keeping the algorithm as simple as possible while meeting its requirements, avoiding unnecessary complexity.

The Complexity of an Algorithm

The algorithm's performance can be measured in two ways:

Time Complexity

The amount of time required to complete an algorithm's execution is called time complexity. The big O notation is used to represent an algorithm's time complexity. The asymptotic notation for describing time complexity, in this case, is big O notation. The time complexity is calculated primarily by counting the number of steps required to complete the execution. Let us look at an example of time complexity.

```
mul = 1;

// Suppose you have to calculate the multiplication of n numbers.

for i=1 to n

mul = mul *1;

// when the loop ends, then mul holds the multiplication of the n numbers

return mul;
```

The time complexity of the loop statement in the preceding code is at least n, and as the value of n escalates, so does the time complexity. While the code's complexity, i.e., returns mul, will be constant because its value is not dependent on the importance of n and will provide the result in a single step. The worst-time complexity is generally considered because it is the maximum time required for any given input size.

Space Complexity

The amount of space an algorithm requires to solve a problem and produce an output is called its space complexity. Space complexity, like time complexity, is expressed in big O notation.

The space is required for an algorithm for the following reasons:

1. To store program instructions.
2. To store track of constant values.
3. To store track of variable values.
4. To store track of function calls, jumping statements, and so on.

Space Complexity = Auxiliary Space + Input Size

Finally after understanding what is an algorithm, its analysis and approches, you will look at different types of algorithms.

Advantage and Disadvantages of Algorithms

Advantages of Algorithms:

- Efficiency: Algorithms streamline processes, leading to faster and more optimized solutions.
- Reproducibility: They yield consistent results when provided with the same inputs.
- Problem-solving: Algorithms offer systematic approaches to tackle complex problems effectively.
- Scalability: Many algorithms can handle larger datasets and scale with increasing input sizes.
- Automation: They enable automation of tasks, reducing the need for manual intervention.

Disadvantages of Algorithms:

- Complexity: Developing sophisticated algorithms can be challenging and time-consuming.
- Limitations: Some problems may not have efficient algorithms, leading to suboptimal solutions.
- Resource Intensive: Certain algorithms may require significant computational resources.
- Inaccuracy: Inappropriate algorithm design or implementation can result in incorrect outputs.
- Maintenance: As technology evolves, algorithms may require updates to stay relevant and effective.