Introduction to Computer programming languages

1. Introduction to computers

A computer is a multipurpose electronic device that can receive process and store data. They are used as tools in every part of society together with the Internet. Computers nowadays are complex; there are a lot of different components inside them, and they all serve different purposes. They all need to work together for the computer to work; knowing how a computer works makes it easier to use a computer by being able to understand how a computer will respond.

Computer

A computer is an electronic device, operating under the control of instructions stored in its own memory that can accept data (input), process the data according to specified rules, produce information (output), and store the information for future use.

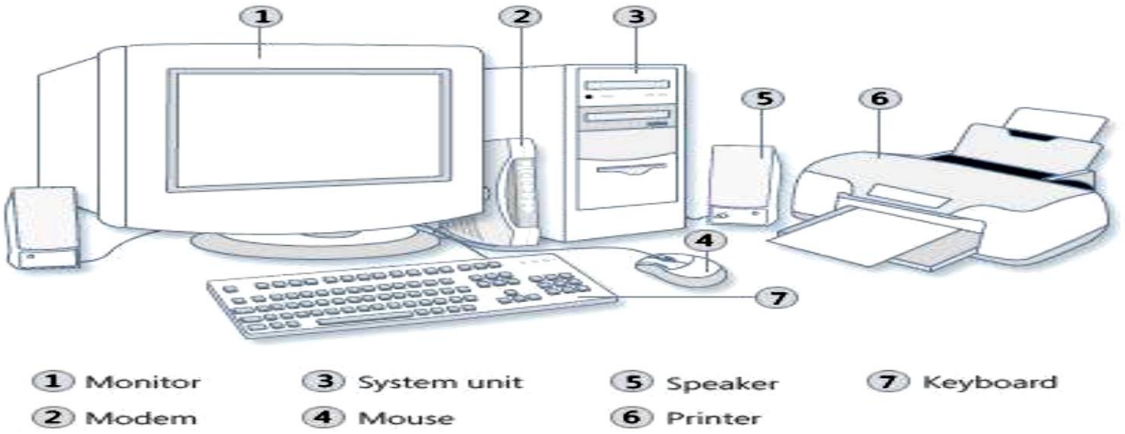
Computer Components

Any kind of computers consists of **Hardware and Software.**

Hardware: Computer hardware is the collection of physical elements that constitutes a computer system. Computer hardware refers to the physical parts or components of a computer such as the

- **Input units:** Mouse, keyboard, scanner …, etc.
- **Output units:** Monitor printer…, etc.
- **Storage units:** computer data storage, hard drive disk (HDD), flash… etc.
- **System unit:** graphic cards, sound cards, motherboard and chips. …, etc.

❖ **Processing unit:** processor CPU.



Software

Software is a generic term for organized collections of computer data and instructions, often broken into two major categories:

System Software that provides task specific functions of the computer. System software consists of an operating system and some fundamental utilities such as disk formatters, file managers, display managers, text editors, user authentication (login) and management tools, and networking and device control software.

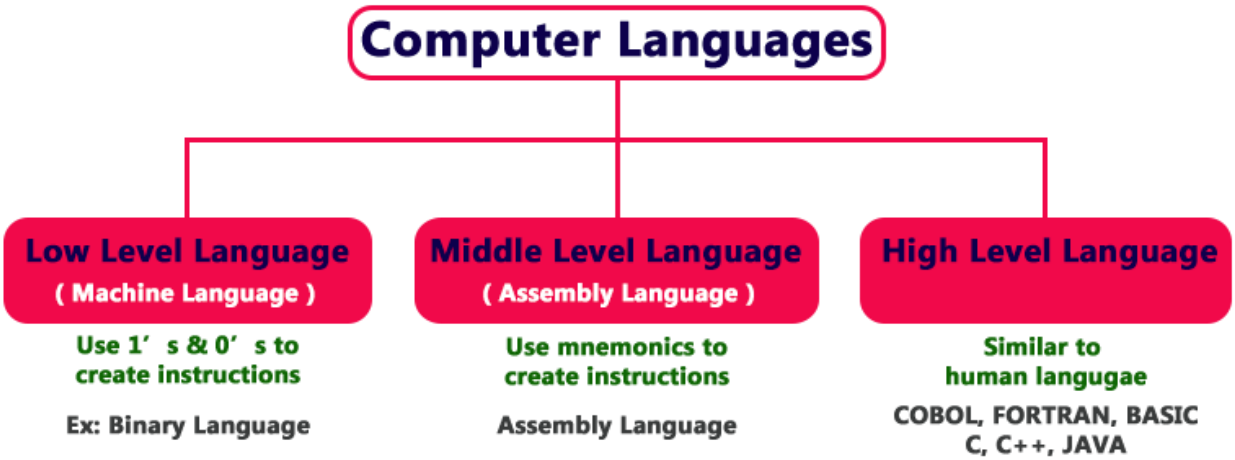
Application Software which is used by users to accomplish specific tasks. Application software may consist of a single program, such as an image viewer; such as Microsoft Office, a software system such as a database management system.

2. Computer programming language

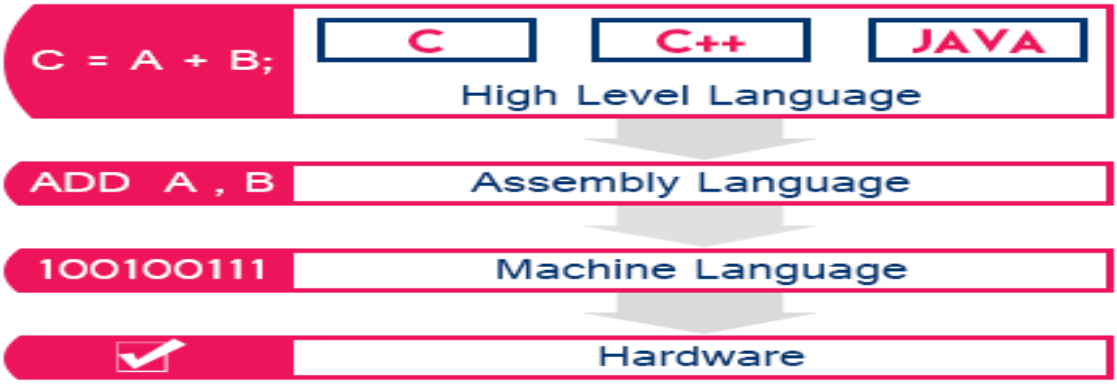
It is the process of designing and building an executable computer program for accomplishing a specific computing task. Programming involves tasks such as analysis, generating algorithms, profiling algorithms accuracy and resource consumption, and the implementation of algorithms in a chosen programming language (commonly referred to as coding). The source code of a program is written in one or more programming languages. The purpose of programming is to find a sequence of

instructions that will automate the performance of a task for solving a given problem. The process of programming thus often requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic. Programs are written either in one of high-level programming languages (such as BASIC, C, Java, pascal, matlab, python) which are easier but execute relatively slowly, or in one of low-level languages (assembly language or machine language) which are very complex but execute very fast.

Generally, we use languages like English, Hindi, and French etc., to make communication between two persons. That means, when we want to make communication between two persons we need a language through which persons can express their feelings. Similarly, when we want to make communication between user and computer or between two or more computers we need a language through which user can give information to computer and vice versa. When user wants to give any instruction to the computer the user needs a specific language and that language is known as computer language. User interacts with the computer using programs and that programs are created using computer programming languages like C, C++, Java, etc. Computer Languages are classified into three languages :

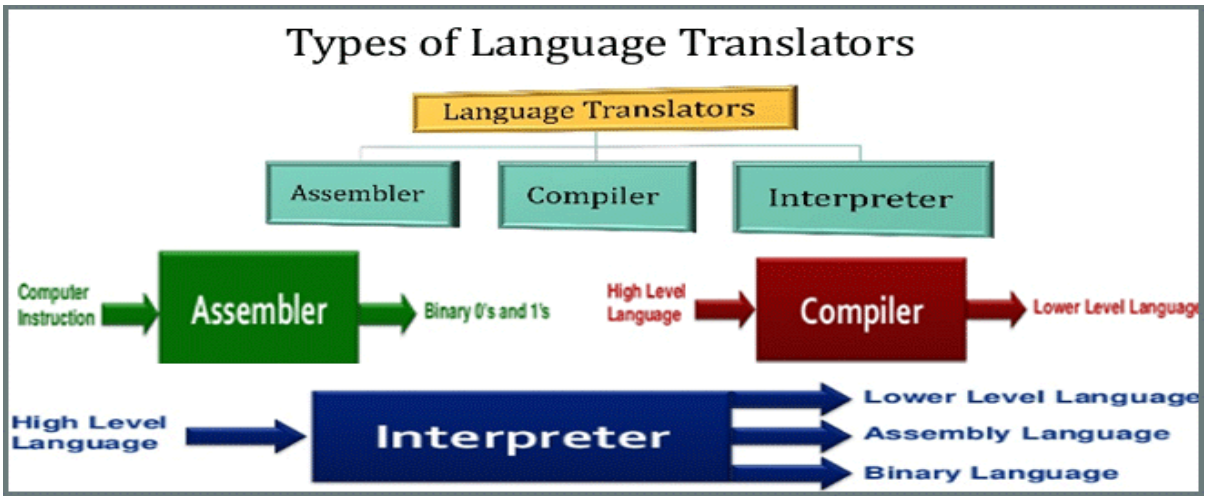


The following figure provides few key points related to the computer languages.



3. Language translators

We generally write a computer program using a high-level language. A high-level language is one which is understandable by us humans. It contains words and phrases from the English (or other) language. But a computer does not understand high-level language. It only understands program written in 0's and 1's in binary, called the machine code. A program written in high-level language is called a source code. We need to convert the source code into machine code and this is accomplished by compilers and interpreters. Hence, a compiler or an interpreter is a program that converts program written in high-level language into machine code understood by the computer.



4. A computer program

In the form of a human-readable, computer programming language is called source code Translators checks program syntax during translation process. The syntax of a computer language is the set of rules that defines the combinations of symbols that are considered to be a correctly structured document or fragment in that language.

Computer language syntax is generally distinguished into three levels:

- Words – the lexical level, determining how characters form tokens;
- Phrases – the grammar level, narrowly speaking, determining how tokens form phrases;
- Context – determining what objects or variables names refer to, if types are valid, etc.

The meaning given to a combination of symbols is handled by semantics; the idea is to examine these notions not from a programmer's point of view but from the language designer's point of view.

5. Algorithm Definition

An algorithm is a set of instructions, sometimes called a procedure or a function that is used to perform a certain task. This can be a simple process, such as adding two numbers together, or a complex function, such as adding effects to an image. Most computer programmers spend a large percentage of their time creating algorithms. (The rest of their time is spent debugging the algorithms that don't work properly.) The goal is to create efficient algorithms that do not waste more computer resources (such as RAM and CPU time) than necessary. This can be difficult, because an algorithm that performs well on one set of data may perform poorly on other data. As you might guess, poorly written algorithms can cause programs to run slowly and even crash. Therefore, software updates are often introduced, touting "improved stability and performance". While this sounds impressive, it also means that the algorithms in the previous versions of the software were not written as well as they could have been.

A main benefit of the use of an algorithm comes from the improvement it makes possible. If the problem solver does not know what was done, he or she will not know what was done wrong. As time goes by and results are compared with goals, the existence of a specified solution process allows identification of weaknesses and errors in the process. Reduction of a task to a specified set of steps or algorithm is an important part of analysis, control, and evaluation.

EXAMPLE:

Write an algorithm and draw a flow chart to calculate 24.

Algorithm power of number.

Step 1: Input Base (2), Power (4)

Step 2: Product= Base

Step 3: Product = Product * Base

Step 4: Product = Product * Base

Step 5: Product = Product * Base

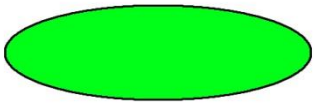
Step 6: Print Product

6. Flowchart Definition

Flowchart is a graphical representation of an algorithm. Programmers often use it as 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".

Basic Symbols used in Flowchart Designs

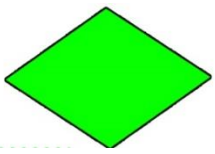
1. **Terminal:** The oval symbol indicates Start, Stop and Halt in a program's logic flow. A pause/halt is generally used in program logic under some error conditions. Terminal is the first and last symbols in the flowchart.



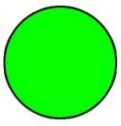
2. **Input/Output:** A parallelogram denotes any function of input/output type. Program instructions that take input from input devices and display output on output devices are indicated with parallelogram in a flowchart.

3. **Processing:** A box represents arithmetic instructions. All arithmetic processes such as adding, subtracting, multiplication and division are indicated by action or process symbol.

4. **Decision** Diamond symbol represents a decision point. Decision based operations such as yes/no question or true/false are indicated by diamond in flowchart.



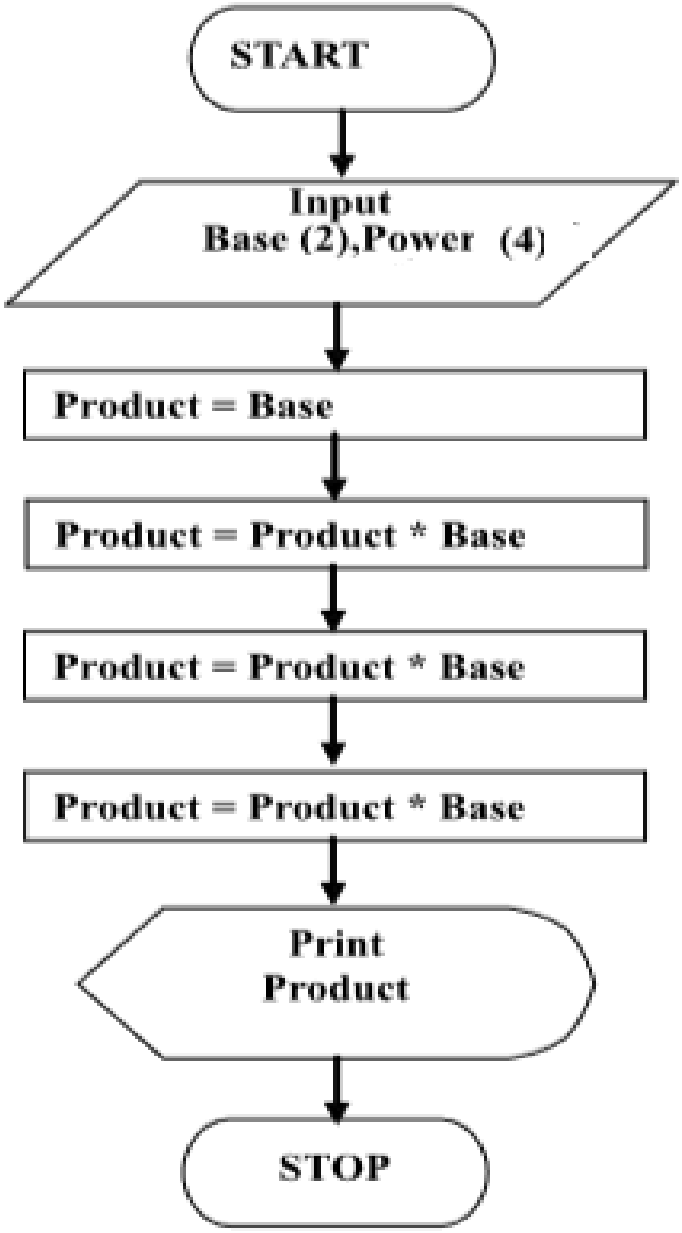
5. **Connectors:** Whenever flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusions. It is represented by a circle.



6. **Flow lines:** Flow lines indicate the exact sequence in which instructions are executed. Arrows represent the direction of flow of control and relationship among different symbols of flowchart.

EXAMPLE1:

Draw a flow chart to calculate 24.



EXAMPLE2:

Draw a flowchart to input two numbers from user and display the largest of two numbers.