# ONLINE
# SUPPROT SERVICES

# CERTIFICATE IN
# INFORMATION TECHNOLOGY

Run & Managed by NASO

# IGNOU SC-2281

## Jakhepal-Ghasiwala Road, Sunam

**For more information visit us at: nirmancampus.co.in**

**Call us at: 9815098210, 9256278000**

## COMPUTER LANGUAGES

Computer languages are used to develop programs or software. There are many types of computer languages. But a computer can understand only binary language. So, the programs written in other languages must be translated into the binary form for execution. Computer Languages can be classified as:

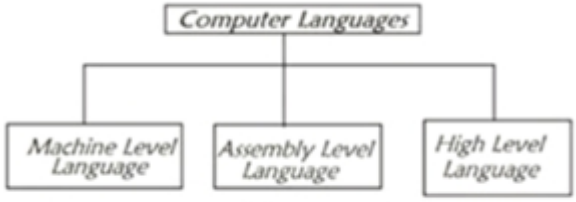


**Figure: Types of Computer Languages**

## MACHINE LANGUAGE (BINARY LANGUAGE)

This is also called as the **First Generation Computer Language**. It is a fundamental language of computer. It is directly understood by the computer. It means it does not require any translation. It consists of 0's and 1's. It is a low level programming language because low level technical hardware knowledge is required for the programming. Therefore, it is a machine dependent language. Normally, a machine language instruction consists of two parts.

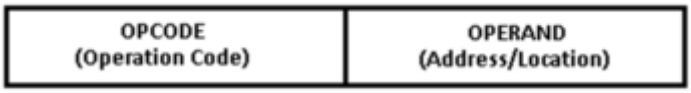


**Fig: Machine Instruction Format**

**OPCODE:** The first part is the operation code. This part tells the computer what function is to be performed.

**OPERAND:** The second part is the operation address. It tells the address of the data on which the operation will be performed.

**Advantages of Machine language:**

- It is faster in execution. It is because the computer understands it directly.
- There is no need of translation.

**Disadvantages of machine languages:**

- **Machine Dependent**: Programs written in Machine Language are not portable, i.e. they are machine-dependent.
- **Difficult to Program**: It is difficult to develop a program using machine language.
- **Difficult to modify**: It is difficult to modify a machine language program.
- Technical Low level hardware knowledge is required for the programming in Machine Language.
- Error Prone

## ASSEMBLY LANGUAGE

This is also called as the **Second Generation Computer Language**. Assembly language uses symbols and mnemonic codes to represent instructions and address. Therefore this language is also known as **Symbolic Language**. It is a low level programming language because low level technical hardware knowledge is required for the programming. Therefore, it is a machine dependent language.

Mnemonic codes are English like codes. These codes are used to perform many types of operations. For example:

**ADD** mnemonic code is used for addition,
**SUB** is used for subtraction,
**MUL** for multiplication etc

Assembly language is not directly understood by computer. So, a translator is required to translate it into machine format. This translator is called **Assembler**. An assembler takes an assembly language program (source Code) as input. It produces machine language code (Object Code) as its output.

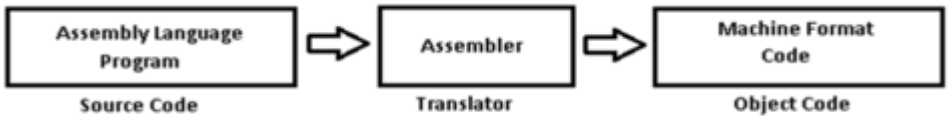


Fig: Assembler

Following example shows the assembly language program. This program shows the addition of two numbers A and B:

| | |
|---|---|
| LD A, 7 | Load register A with 7 |
| LD B, 10 | Load register B with 10 |
| ADD A, B | A ←A+ B |
| LD (100), A | Save the result in the Memory location 100 |
| HALT | Halt process |

**Advantages of Assembly Language:**

- Easier to Understand and Use
- Easier to locate and correct errors
- Easier to modify
- Efficiency of Machine Language
- No need to keep track of storage locations of the data and instructions

**Disadvantages of Assembly language:**

- Low level technical hardware knowledge is required for the programming in Assembly language.
- Assembly language programs are machine dependent.

**HIGH LEVEL LANGUAGES**

These languages are also called as the **Third Generation Computer Languages.** These languages use **alphanumeric codes** to write programs. These languages are very close to English like languages. No low level technical hardware knowledge is required for the programming in these Languages. Therefore, these languages are **machine independent**. FORTRAN, COBOL, BASIC, PASCAL, C etc are the examples of High Level Languages.

High level language is **not directly understood** by computer. So, a **translator** is required to translate it into machine format. This translator is called **Compiler or Interpreter**. A Compiler takes a high level language program (source Code) as input. It produces machine language code (Object Code) as its output.

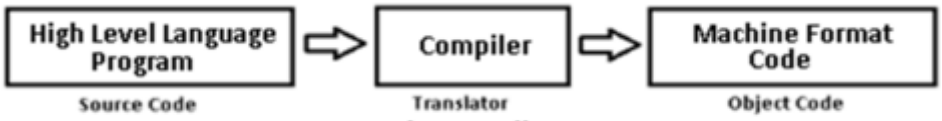


Fig: Compiler

Following example shows the high level language program written in C. This program shows the addition of two numbers A and B:

```
#include<stdio.h>
void main()
{
        int A,B,C;
        A=56;
        B=67;
        C=A+B;
        printf("%d",C);
}
```

**Advantages of High Level Programming Language:**

- Machine Independent programs
- Easy to learn and use
- Programs are more readable.
- Programs could be run on different machines with little or no modification. It means programs are portable.
- Finding and correcting errors is easy (Easy Debugging).
- High level languages provide better documentation (comments).
- Programs are easier to maintain.

**Disadvantages of High Level Programming Languages:**

- High level languages has lower efficiency than machine and assembly languages
- They are less flexible because they do not have the instructions to control computer hardware (CPU, memory, registers etc)

**THE FOURTH GENERATION LANGUAGES (4GL) OR PROBLEM ORIENTED LANGUAGES**

Fourth Generation Language has not been developed yet. The fourth generation languages are non-procedural languages. These languages are also known as Problem Oriented Languages. They are highly user friendly. The user has to define the problem, provide data and the output required. These languages call appropriate procedures to solve the problem and provide the required output.

Most of the 4GL's are menu driven languages. A menu is a list of facilities to solve a problem. Examples of these languages are: LOTUS 1-2-3, dBASE III plus, dBase IV, Fox Pro, Ingres, Oracle etc.