SNS COLLEGE OF TECHNOLOGY

Coimbatore-36.
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : 19CST101 – PROGRAMMING FOR PROBLEM SOLVING

I YEAR/ I SEMESTER

UNIT – I INTRODUCTION TO PROBLEM SOLVING TECHNIQUES

Topic: Notation (Programming Language)

Mrs.Devi G

Assistant Professor

Department of Computer Science and Engineering

Programming Language

•A programming language is a set of symbols and rules for instructing a computer to perform specific tasks.

•The programmers have to follow all the specified rules before writing program using programming language.

•The user has to communicate with the computer using language which it can understand.
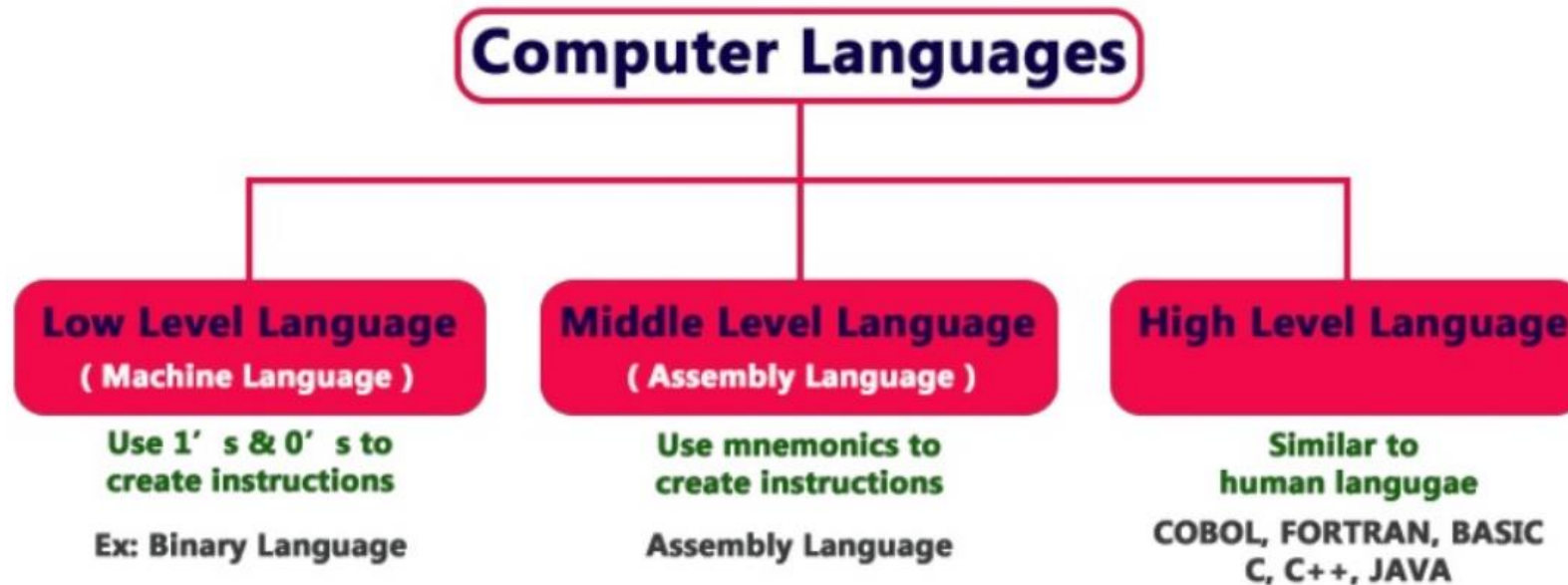
Program= Algorithm + Data

•Need for Programming Languages
- Used to organize the computation.
- We can solve different problems.
- To improve the efficiency of the programs.

Types of programming language

- Low – level   or Machine Language
- Intermediate or Assembly Language
- High – level or Programming language

Computer Languages

Low Level Language (Machine Language)	Middle Level Language (Assembly Language)	High Level Language
Use 1's & 0's to create instructions	Use mnemonics to create instructions	Similar to human langugae
Ex: Binary Language	Assembly Language	COBOL, FORTRAN, BASIC C, C++, JAVA

Machine language:

• Machine language is the lowest-level programming language.

• Machine languages are the only languages understood by computers.

• It is also called as low level language.

• In machine language the different instructions are formed by taking different combinations of 0's and 1's.

<p style="text-align:center">Example code:100110011</p>
<p style="text-align:center">111001100</p>

Advantages:

• **Translation free**
  • The program written in machine language can be executed directly on computer.
  • In this case any conversion process is not required.
• **High speed**
  • The conversion time is saved, the execution of machine language program is extremely fast.

Disadvantage:

• It is hard to find errors in a program written in the machine language.
• Writing program in machine language is a time consuming process.
• **Machine dependent:** According to architecture used, the computer differs from each other.

Assembly Language:

- To overcome the issues in programming language and make the programming process easier, an assembly language is developed which is logically equivalent to machine language but it is <u>easier for people to read, write and understand</u>.

- Assembly language is symbolic representation of machine language.

- Assembly languages are symbolic programming language that uses symbolic notation to represent machine language instructions.

- They are called low level language because they are so closely related to the machines.

- An assembly language contains the same instructions as a machine language, but the <u>instructions and variables have names</u> instead of being just numbers.

- An assembly language consists of mnemonics, mnemonics that corresponds unique machine instruction.

> Example code: start
> Add x,y
> Sub x,y

Assembler:
- Assembler is the program which translates assembly language instruction into a machine language.
  - Easy to understand and use.
  - It is easy to locate and correct errors.

Disadvantages:
- Machine dependent:
  - The assembly language program which can be executed on the machine depends on the architecture of that computer.
- Hard to learn:
  - It is machine dependent, so the programmer should have the hardware knowledge to create applications using assembly language.
- Less efficient :
  - Execution time of assembly language program is more than machine language program.
  - Because assembler is needed to convert from assembly language to machine language.

High Level Language

High – level Language:

- High level language contains English words and symbols.

- The specified rules are to be followed while writing program in high level language.

- The interpreter or compilers are used for converting these programs into a machine readable form.

- A high-level language (HLL) is a programming language such as C, FORTRAN, or Pascal that enables a programmer to write programs that are more or less independent of a particular type of computer.

- Such languages are considered high-level because they are closer to human languages and further from machine languages.

- Ultimately, programs written in a high-level language must be translated into machine language by a compiler or interpreter.

Example code: printf("Hello World!")

Translating high level language to machine language:

- The programs that translate high level language in to machine language are called interpreter or compiler.

Interpreter:

- Interpreter translates the high level language program in line by line manner.
- The interpreter translates a high level language statement in a source program to a machine code and executes it immediately before translating the next statement.
- When an error is found the execution of the program is halted and error message is displayed on the screen.
- Ex :Pascal, Python

SOURCE CODE → INTERPRETER → OUTPUT
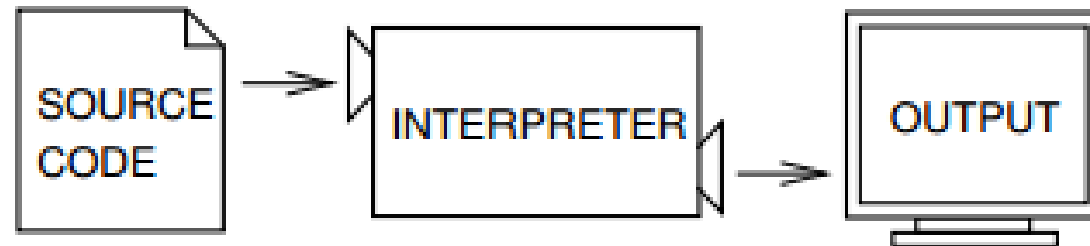
Compiler:

- A compiler is a program which translates the source code written in a high level language in to object code which is in machine language program.
- Compiler reads the whole program written in high level language and translates it to machine language.
- If any error is found it display error message on the screen.
- Ex: C, C++, JAVA
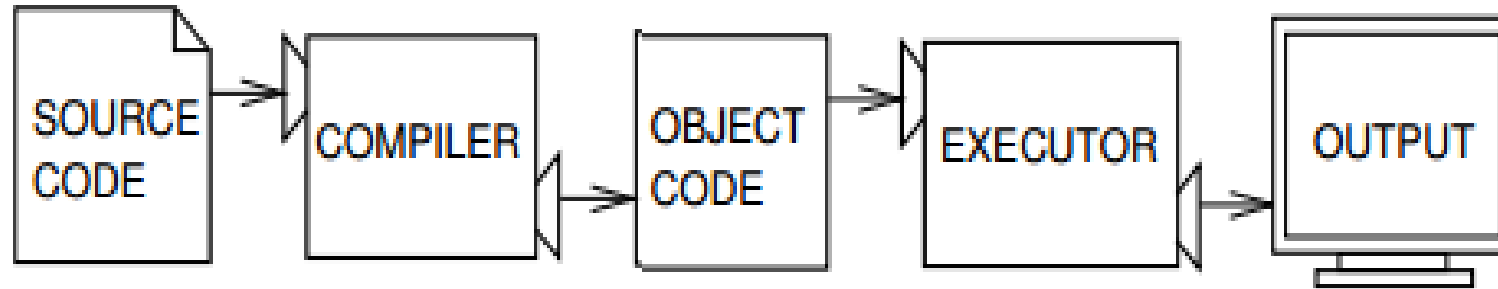
High Level Language

Advantages

- **Readability**
  - High level language is closer to natural language so they are easier to learn and understand

- **Machine independent**
  - High level language program have the advantage of being portable between machines.

- **Easy debugging**
  - Easy to find and correct error in high level language

Disadvantages

- **Less efficient**
  - The translation process increases the execution time of the program.

  - Programs in high level language require more memory and take more execution time to execute.

High Level Language

High Level Language are divided into following categories:

Language Type	Example
1. Interpreted programming languages	Pascal, Python
2. Functional programming languages	Clean, Haskell
3. Compiled programming languages	C, C++, C#, JAVA
4. Procedural programming languages	Hyper talk, MATLAB
5. Scripting programming language	Apple script, VB script
6. Mark-up programming language	HTML, XML
7. Concurrent programming language	Joule, Limbo
8. Object oriented programming language	Lava, Moto, C++, JAVA

Notations/19CSTt101 –Programming for Problem Solving/ Devi/CSE/SNSCT