**Day-02**

**06-02-2025**

**==============**

**Programming Fundamentals**

**==================**

-> Language, we can consider for the interactions/communications.

-> Languages are classified into two types:

1) Natural Languages

2) Programming Languages

-> Natural languages can be used to make interactions between humans/users.

-> Programming languages are required for the interactions between user and computer.

-> Programming languages are classified into three types:

1) Binary language

2) Assembly language

3) High level language

-> Binary language has binary instructions

Binary instructions are with 1's and 0's only.

ex:

User =======> Computer

A 1000001

12 + 13

binary of 12 (1100) binary of +(1010101) binary of 13 (1101)

-> Handling of binary instructions to the user is very difficult. To overcome this issue, assembly language is introduced.

-> Assembly languages are user friendly.

-> Because any development with assembly language is possible with user understandable tokens.

Ex:

add ==> sum

sub ==> subtraction

mul ==> multiplication

mov ==> assigning of data to variable

div ==> division etc.

ex:

add 12,13;

+

12 ==> 1100

13 ==> 1101

-> The difficulty with assembly language is:

it is platform dependent.

-> The assembly language application need to be develop uniquely and independently from processor to processor.

**Platform Vs Processor**

**==============**

Platform ==> core software

which allows the other applications to run.

Software ==> collection of programs.

Application ==> is also the collection of programs

Software need not required platform to use

Application need platform to use.

Ex: WhatsApp ==> can run on IoS or Android

Notepad ==> can run on Windows.

-> Platforms are classified into two types:

1) Desktop based

==> Windows, Linux, Mac-os

2) Mobile based

==> IoS, android etc.

Processor ==> can manage the resources among multiple services.

-> To overcome the drawback of assembly language, the high-level languages are introduced.

-> The high-level languages are user-friendly.

-> Portable (Without any modification, we can extend the application to any platform).

-> Platform independent.

-> Processor independent.

-> Examples for high-level languages are:

Ex: C, C++, Python, Java etc.

**Working Domains:**

**============**

Banking Domain

Healthcare Domain

Insurance

Ecommerce

Stocks etc.

**Q: How the programmer can interact with computers?**

**==================================**

-> Using high-level programming languages

-> programmer can develop programs

-> using these programs, the programmer can communicate with computers.

**How computers can understand programs?**

**============================**

High-level programming languages ==> Translators

Program <=============> Device

Translator

-> Translator is a software, which can use to convert the program into machine understandable format.

-> Program ==> source code

The extension of the source code can differ from language to language

Ex: c source code ==> file.c

python ==> .py etc.

-> Machine code ==> in 1's and 0's format

==> Object code

==> has the extension of '.obj'

-> Two types of translators for high-level languages:

1) Compiler

2) Interpreter

**Q: Do we able to develop any application using natural languages?**

**======================================**

No

Because, Natural languages does not have any translators.

#icnlude<stdio.h>

main(){

printf("Hello");

}