**Day-02**

**20-02-2025**

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

**Programming Fundamentals:**

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

**Languages:**

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

-> Language is used to establish the communication between objects.

-> Languages are classified into two types:

1) Natural Languages

2) Programming Languages

-> When two persons want to communicate, natural languages can be used.

Ex: English, Telugu, Hindi, etc.,

-> When a person want to communicate with the computer (application), programming languages can be used.

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

-> Programming languages are classified into three types:

1) Binary Languages

2) Assembly Languages

3) High-level Languages

**1) Binary Languages:**

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

-> Binary ==> two

-> Binary languages can use only two letters for any communication.

those are: 1 and 0

Ex: When a person want to send 'A' to the computer

'A' , for this, user need to send binary

'A' ===> Binary

ASCII (American Standard Code Information Interchange)

Ex: A to Z ==> 65 to 90

a to z ==> 97 to 122

0 to 9 ==> 48 to 56

ASCII Values

'A' ==> ASCII = 65

Binary ==> 1000001

-> Drawback:

1) Too Complex procedure

2) Too difficulty to handle or to maintain binary languages

**2) Assembly Languages:**

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

-> Any development using assembly languages possible with some tokens like:

add, sub, mul, div, etc.,

Ex: adding of two numbers

add 90,80;

subtraction

sub 90,80;

multiplication

mul 9,8;

-> Comparing with the binary languages, the writing of code/program using assembly language is easy. And we can easily maintain and handle these codes.

-> Drawback:

Processor dependent

Ex: assume, we have 10's of processors

X-application ===> developed by the user on the processor of 'y'

Client ==> have instructed that the x-application want to run on another platform like 'z'

code at 'y' may or may not be work on 'z'

**3) High-level Programming languages:**

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

-> User friendly, because it consists of some tokens (English like words)

-> Processor independent

-> Platform independent

Ex: Java, Python, C# etc.,

**Day-03**

**21-02-2025**

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

**Python Introduction**

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

**How python is easy than other programming Languages**?

========================================

C program:

==========

#include<stdio.h>

void main(){

printf("Hello, Good morning!");

printf("Welcome To Ashok IT!");

}

Java program:

=============

class Greetings{

public static void main(String[] args){

System.out.println("Hello, Good morning!");

System.out.println("Welcome To Ashok IT!");

}

}

Python Program:

===========

print("Hello,Good Morning!")

print("Welcome To Ashok IT!")

**Scripting Vs Programming**

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

-> Programmer can be involved to develop end to end functionality of web application.

But the tester can involve to develop small amount of code to test the functionality of the web application in positive way (using positive test case) and in negative way (using negative test case).

-> Programming Languages are either compiler or interpreter or both compiler and interpreter dependent languages.

Ex: C ==> Compiler

Python ==> Interpreter

Java ==> Compiler & Interpreter

-> Scripting languages are always interpreter dependent only.

-> Compiler's performance is faster than the interpreter.

**How python program can program?**

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

Source code =======> application/computer

Source code ======> Object code (machine readable code)

Source code ============> compiled =====> Byte code ==> processors

Byte code =============> Object code ====> platform

PVM/Interpreter

Object code ===> in the format of 1's and 0's

Byte code ==> symbols (processor's symbols)

**Python Features:**

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

1) High level programming language

2) Easy programming languages

3) Open source

4) Free

5) Processor independent

6) Platform independent

7) General Purpose Programming language

Applications are three types:

1) Standard Applications

=========================

Ex: Notepad, Calculator, Paint etc.

2) Web applications

===================

Ex: IRCTC, Amazon etc.

3) Distributed applications

==================

Ex: Phonepe ================> Banking servers

person ==> ICICI

Check balance using Phonepe

logged in with Phonepe

Phonepe <=====================> ICIC Bank Server

UPI Id