

Practical-1: Introduction

Discuss the following points:

1. History of Python

Ans :-

The programming language Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to ABC capable of exception handling and interfacing with the Amoeba operating system. Python was named after the BBC TV show Monty Python's Flying Circus. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a simple, general purpose, high level, and object-oriented programming language. Python is an interpreted scripting language

2. Differentiate compiler and interpreter.

Ans :-

COMPILER	INTERPRETER	
- A compiler translates complete high-level programming code into machine code at once.	- An interpreter translates one statement of programming code at a time into machine code.	TRANSLATION TYPE
- As the source code is already converted into machine code, the code execution time becomes short.	- As the source code is interpreted line-by-line, error detection and correction become easy.	ADVANTAGES
- It stores the converted machine code from your source code program on the disk.	- It never stores the machine code at all on the disk.	MACHINE CODE
- If you want to change your program for any reason, either by error or logical changes, you can do it only by going back to your source code.	- Interpreted programs can run on only those computers which have the same interpreter.	DISADVANTAGES
- A compiler takes an enormous time to analyze source code. However, overall compiled programming code runs faster as compared to an interpreter.	- An interpreter takes less time to analyze source code as compared to a compiler. However, overall interpreted programming code runs slower as compared to the compiler.	RUNNING TIME

3. What types of applications can be developed by using Python?

Ans :-

- Develop Desktop based applications (eg :- calculator) using python Tkinter.
- Develop Web applications. Python provides Django framework and flask as well to develop web applications.
- Game development.
- In data science.
- For machine learning.
- Also useful in Artificial Intelligence.
- IOT based applications.
- Networking related applications.
- Chatbot.
- Database oriented applications using python is also possible.

4. List down Python versions.

Ans :-

Python 0.9.0 released on February 1991
Python 1.0 released on January 1994
Python 2.0 released on October 2000
Python 2.7.0 released on July 2010
Python 3 released on December 2008
Python 3.6 released on December 2016
Python 3.6.5 released on March 2018
Python 3.7.0 released on May 2018
Python 3.8 released on October 2019
Python 3.9 released on October 2020

5. List out Python IDE.

Ans :-

- IDLE :- Integrated Development and Learning Environment.

IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit.



- Visual studio :-

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.



- Anaconda :-

Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.



6. List out the features of Python.

Ans :-

- 1) Easy to Learn and Use. Python is easy to learn as compared to other programming languages.
- 2) Expressive Language.
- 3) Interpreted Language.
- 4) Cross-platform Language.
- 5) Free and Open Source.
- 6) Object-Oriented Language.
- 7) Extensible.
- 8) Large Standard Library.

- 9) GUI Programming Support.
- 10) Integrated.

7. Write advantages of Python language over other languages.

Ans :-

- Advantages of python over java :-

(i) Code of java

```
//create class HelloWorldExample class
public class HelloWorldExample
{
    //main() method start
    public static void main(String[] args)
    {
        //print hello world
        System.out.println("Hello World!");
    }
}
```

In Java, we have to declare the type of the variable explicitly because of the dynamic type nature it. In Java, the code will not compile even if a small irregularity present.

(ii) Code of python

```
print("Hello World")
```

As we already discussed above that Python is a dynamically typed language. It means that we don't need to define the type of the variable as it checks automatically by the interpreter at runtime. The Python code has become much more readable and simpler due to its design of "English-like".

- Advantages of python over C++ :-

(i) Code of C++

```
#include <iostream>
using namespace std;
int main()
{
    int a=20;
    std::cout << "value of a is : " <<a<< std::endl;
    return 0;
}
```

- It is not easy to learn because of its complex syntax.

(ii) Code of python

```
# python program
#integer assignment
a=20
print(a)
```

- It is easy to learn, as it does not follow any programming rules. It follows the indentation rules, which is very much similar to English.

In the above two programs, the output would be 20. The difference in the above two programs is that in C++, we need to declare the variable with its type, while in python, we do not need to declare the variables.

-Advantages of python over c# :-

(i) Code of C#

```
using System;
Namespace PrintNameApplication {
class PrintUserName {
    static void Main(string[] args) {
        /* Write user name to console */
        String userName;
        userName = Console.ReadLine();
        Console.WriteLine("Hello, " + username + ". How are you today?");
    }
}
}
```

- Statically typed. The compiler will give errors for wrong typecasting
- More organized and consistent syntax and format.

(ii) Code of python

```
# print
namename = input("Enter your name-")
print("Your name is ", name)
```

- Dynamic type-casting. No need for variable declarations.
- Simple, easy to read and code, doesn't contain too many symbols or formats.

What we achieved in about 10 lines in C#, we have done in just 2 lines in Python. The code is just like typing a sentence in English! Note that there are no ';' (semicolons) at the end of each line. The comments are added using '#' as compared to '/*' in C#. There are no type declarations. I haven't written 'String name;' anywhere in the code. There are no imports!

8. Differentiate dynamic and static types programming.

Ans :-

Static type programming	Dynamic type programming
- Static typed is a programming language in which a programmer must declare the variables clearly before using them. As a programmer, you don't have to define the variables before they are put into use. The most popular static typed programming languages are C, C++, and Java.	- Dynamic typed is a complete opposite of the statically typed programming language. In dynamic typed, all the variables must be defined before they are used. The good thing about dynamic typed is you don't have to declare these variables.
- Statically typed programming languages are compiled when executed. Each line of code is translated before the run-time.	- On the other hand, dynamically typed programming languages are interpreted when executed.
- In statically typed programming languages, types are checked before the run-time. If the code is found to be having an unsuitable type, an error will be thrown before the run-time.	- For dynamically typed programming languages, types are checked during the execution of the code. Even if the line of code has a mistake, an error won't be thrown.
- The performance of statically typed languages remains good at run-time as there is no need of type-checking.	- This is the opposite of dynamic languages as the codes have to be type-checked during run-time.
- static programming languages don't allow the changing of variable types.	- dynamic programming allows variables to change types.

9. Differentiate procedural and object oriented programming.

Ans :-

Procedural oriented programming	Object oriented programming
- In procedural programming, program is divided into small parts called functions.	- In object oriented programming, program is divided into small parts called objects.
- Procedural programming follows top down approach.	- Object oriented programming follows bottom up approach.
- There is no access specifier in procedural programming.	- Object oriented programming have access specifiers like private, public, protected etc.
- Procedural programming does not have any proper way for hiding data so it is less secure.	- Object oriented programming provides data hiding so it is more secure.
- In procedural programming, overloading is not possible.	- Overloading is possible in object oriented programming.
- Examples: C, FORTRAN, Pascal, Basic etc.	- Examples: C++, Java, Python, C# etc.