Session 1: Python Programming

Python is a general purpose high level programming language. As Python is used for general use or in multiple domain areas including web applications, database applications, standalone applications (desktop applications), etc., it is considered to be a general purpose programming language. Python programming is carried out in human language, i.e. no machine level language or mnemonics need to be studied in order to work with Python, thus, it is considered a high level programming language.

Introduction to Python:

Python was developed by Guido van Rossum in 1989 at the Centrum Wiskunde & Informatica (CWI), which is the national research institute for mathematics and computer science in the Netherlands. However, Python was officially released on 20th of February, 1991. The name 'Python' comes from a famous BBC TV comedy sketch series called *Monty Python's Flying Circus*, which Rossum was a big fan of.

Many people believe that Python should be the first programming language to learn, as it is a beginner friendly language and is easier to learn than most other programming languages like C, C++, Java, etc.
For example, let us print "Hello World" in a few languages:

Using C:

```c
#include <stdio.h>
#include <conio.h>
int main(){
printf("Hello World");
return 1;
getch();
}
```

Using Java:

```java
public class HelloWorld{
public static void
HelloWorld(String[]
args){
System.out.println("Hello
World");
}
}
```

Using Python:

```python
print "Hello
World";
```

The output from all the different languages is: `Hello World`

Python is dynamically typed which means that there is no need to declare the type of a variable before using it. In contrast, C is statically typed, which means the type of a variable should be declared explicitly before using it. For example, let us add two variables in dynamically typed and statistically typed languages:

Using C:

```c
#include <stdio.h>
int main() {
    int a = 5;
    int b = 10;
    int sum = a + b;
printf("The sum is: %d\n", sum);
return 0;
}
```

Using Python:

```python
a = 5
b = 10
sum = a + b
print("The sum is:", sum)
```

In terms of C:

- The type of each variable is declared explicitly (int in this case).
- The C compiler ensures type safety by checking that variables are used consistently with their declared types.

In terms of Python:

- Variables a and b are created without specifying their type. Python automatically infers the type (int in this case).
- The type of a variable can change at runtime. You could later assign a string to a without any error, and Python would handle it accordingly.
  For example:

```python
a = 5
print(type(a))
#Output: <class 'int'>

 a = "Hello"
print(type(a))
# Output: <class 'str'>
```

In contrast, trying to change the type of `a` in C after its declaration would result in a compile-time error.

Rossum, the developer of Python, was also present during the development of C language at Bell Lab. But he realized that the C was a very complex language for the general public to use, thus, Rossum focused on creating a high level language which could cater to a wider group of people so they could learn to code using it easily.

Python language is a compilation of features from different languages such as the following feature/concept:

- Functional programming from C.
- OOP programming from C++
- Scripting Language from Perl & Shell Script.
- Modular programming from Modula-3.

Most of the syntax of Python is similar to C and ABC language.

Application areas of Python:

- Desktop Applications/ Standalone Applications (Frameworks such as tkinter, kiwi,etc.)
- Web Applications (Frameworks such as Django, Flask, Web2py,etc.)
- Database Applications
- Network Programming
- Game Development
- Data Analysis
- ML and AI
- IOT, etc.

Python is used in almost every national and multinational companies such as YouTube, Uber, NASA, IBM, Netflix, etc. Microservices are used mostly in large companies where an application is composed of small, independent services that work together where each service might be an implementation of a different language including python.

Features of Python:

- Simple and easy to learn: It only includes about 30+ keywords and less number of lines of code than other languages.
- Freeware and open source: It can be downloaded and used without any cost and also allows anyone to view, modify, and contribute to its source code.
- High level programming language: It is based on human language.
- Platform Independent: It is independent of the platform such as Windows, MacOS, Linux, etc.
- Portability: It is portable, i.e. Python code can run on different operating systems (e.g., Windows, macOS, Linux) without modification.

- Dynamically typed: It doesn’t declare the data type before using.
- Interpreted: It is an interpreted language, which means that its code is executed line-by-line by the Python Virtual Machine (interpreter) at runtime, rather than being compiled into machine code beforehand which is the case in languages such as C. This allows for easier debugging and platform independence but can result in slower execution compared to compiled languages.
- Extensible: It can extend its capabilities by writing parts of the code in other languages like C/ C++/Java.
- Embedded: It can be embedded within other programming languages, allowing you to include Python code in applications written in languages like C/ C++.
- Extensive Library: It comes with a vast standard library that provides modules and functions for a wide range of tasks which reduces the need to write code from scratch.
- Huge community: It consists of a large and active community of developers, which helps to find solutions and resources.

Limitations of Python:

- Python is considered slow compared to languages like C and C++ because it is an interpreted language and uses dynamic typing, which introduces overhead during execution. However, since around 2015, Python's performance has significantly improved with optimizations in the interpreter (like CPython, Jython,etc.)

Flavors of Python:

- Cpython: The reference implementation of Python, written in C. It is the most widely used version and is known for its stability and extensive library support.
- Jython/JPython: An implementation of Python that runs on the Java Virtual Machine (JVM). It allows seamless integration with Java libraries and applications.
- IronPython: A Python implementation designed to run on the .NET framework. It enables Python to interact with .NET libraries and applications.
- RubyPython: It is the integration or usage of Python with Ruby which is known for its elegant syntax and strong focus on simplicity and productivity.
- AnacondaPython: It is a distribution of Python designed for data science and machine learning. It includes the Python interpreter along with a comprehensive set of libraries and tools for scientific computing, data analysis, and visualization.

Versions of Python:

- Python 1.0: Introduced in 1994.
- Python 2.0: Introduced in 2000.

- Python 3.0: Introduced in 2008.

There will be no more new versions of Python, new releases will all be included in the version 3.x. The latest version of Python is the Python 3.13 (pre release).