Practical -1 : Introduction

Discuss the following points:

1) History of Python.

Ans :-

Python was **created by Guido van Rossum, and first released on February 20, 1991**. While you may know the python as a large snake, the name of the Python programming language comes from an old BBC television comedy sketch series called Monty Python's Flying Circus.

2) Differentiate compiler and interpreter.

Ans :-

S.No.	Compiler	Interpreter
1.	The compiler scans the whole program in one go.	Translates the program one statement at a time.
2.	As it scans the code in one go, the errors (if any) are shown at the end together.	Considering it scans code one line at a time, errors are shown line by line.
3.	The main advantage of compilers is its execution time.	Due to interpreters being slow in executing the object code, it is preferred less.
4.	It converts the source code into object code.	It does not convert source code into object code instead it scans it line by line
5	It does not require source code for later execution.	It requires source code for later execution.
6	Execution of the program takes place only after the whole program is compiled.	Execution of the program happens after every line is checked or evaluated.
7	The machine code is stored in the disk storage.	Machine code is nowhere stored.
8	Compilers more often take a large amount of time for analyzing the source code.	In comparison, Interpreters take less time for analyzing the source code.
9.	It is more efficient.	It is less efficient.
10.	CPU utilization is more.	CPU utilization is less.
Eg.	C, C++, C#, etc are programming languages that are compiler-based.	Python, Ruby, Perl, SNOBOL, MATLAB, etc are programming languages that are interpreter-based.

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

Ans :-

- Web Applications
- Desktop GUI Applications
- Console based Applications
- Software Development
- Scientific and Numeric
- Business Applications
- Audio or Video – based Applications

4) List down Python versions.

Ans :-

Python 1.0	January 1994
Python 1.5	December 31, 1997
Python 1.6	September 5, 2000
Python 2.0	October 16, 2000
Python 2.1	April 17, 2001
Python 2.2	December 21, 2001
Python 2.3	July 29, 2003
Python 2.4	November 30, 2004
Python 2.5	September 19, 2006
Python 2.6	October 1, 2008
Python 2.7	July 3, 2010
Python 3.0	December 3, 2008
Python 3.1	June 27, 2009
Python 3.2	February 20, 2011
Python 3.3	September 29, 2012
Python 3.4	March 16, 2014
Python 3.5	September 13, 2015
Python 3.6	December 23, 2016
Python 3.7	June 27, 2018
Python 3.8	October 14, 2019

5) List out Python IDE.

Ans :-

- IDLE (Integrated Devlopment and Learning Environment)
- Pycharm
- Visual Studio Code
- Sublime Text 3
- Atom
- Jupyter Notebook
- Spyder
- PyDev
- Thonny
- Wing

6) List out the features of Python.

Ans :-

- ✓ Easy to code
- ✓ Easy to Read
- ✓ Free and Open-Source
- ✓ Robust Standard Library
- ✓ Interpreted
- ✓ Portable
- ✓ Object-Oriented & Procedure-Oriented
- ✓ Extensible
- ✓ Expressive
- ✓ Support for GUI
- ✓ Dynamically Typed
- ✓ High-level Language
- ✓ Simplify Complex Software

7) Write advantages of Python language over other languages.

Ans :-

1. It is Free.
2. It needs Less Coding
3. All Kinds of Businesses Can afford it
4. Big Giants are using it
5. It is one of the most trending language
6. Extensive libraries
7. Large developer community

8) Differentiate dynamic and static types programming.

Ans :-

- **Static Type Programming :-**
  - It follows type checking during compilation (compile time). So, every detail about the variables and all the data types must be known before we do the compiling process.
  - e.g.:- *int a = 10;*
  - Java, C, C++, C#, Swift, etc are the example of static type programming.
- **Dynamic Type Programming :-**
  - In this type checking takes place while the program runs(run time). In this type of language, there is no need to specify the data type of each variable while writing code.
  - E.g.:- *a = 10*
  - Python, Javascript, Ruby, PHP, etc are the example of dynamic type programming.

9) Differentiate procedural and object-oriented programming.

Ans :-

Procedural Oriented Programming	Object-Oriented Programming
In procedural programming, the program is divided into small parts called ***functions***.	In object-oriented programming, the program is divided into small parts called ***objects***.
Procedural programming follows a ***top-down approach***.	Object-oriented programming follows a ***bottom-up approach***.
There is no access specifier in procedural programming.	Object-oriented programming has access specifiers like private, public, protected, etc.
Adding new data and functions is not easy.	Adding new data and function is easy.
Procedural programming does not have any proper way of 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.
In procedural programming, there is no concept of data hiding and inheritance.	In object-oriented programming, the concept of data hiding and inheritance is used.
In procedural programming, the function is more important than the data.	In object-oriented programming, data is more important than function.
Procedural programming is based on the ***unreal world***.	Object-oriented programming is based on the ***real world***.
Procedural programming is used for designing medium-sized programs.	Object-oriented programming is used for designing large and complex programs.
Procedural programming uses the concept of procedure abstraction.	Object-oriented programming uses the concept of data abstraction.
Code reusability absent in procedural programming,	Code reusability present in object-oriented programming.
Examples: C, FORTRAN, Pascal, Basic, etc.	**Examples:** C++, Java, Python, C#, etc.