PYTHON - SDLC

**Que.1) What is the definition of software?**

Ans: Software is **a set of instructions, data or programs used to operate computers and execute specific tasks**. It is the opposite of hardware, which describes the physical aspects of a computer. Software is a generic term used to refer to applications, scripts and programs that run on a device.

**Que.2)What are the type of applications?**

Ans: An application, also referred to as an application program or application software, is a computer software package that performs a specific function directly for an end user or, in some cases, for another application. An application can be self-contained or a group of programs. The program is a set of operations that runs the application for the user.

**Application examples**

Some types of applications include:

* Word processors
* Database programs
* Web browsers
* Deployment tools
* Image editors
* Communication platforms

**Que.3)What is programming?**

Ans: Programming is the process of creating a set of instructions that tell a computer how to perform a task. Programming can be done using a variety of computer programming languages, such as JavaScript, Python, and C++.

**Que.4)What is Python?**

Ans: Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.