**PYTHON ASSIGNMENT – 1**

**Q1. What is Cpython and Jpython?**

1. **CPython**

The **default implementation of the Python programming language is Cpython**. As the name suggests **Cpython is written in C language**. Cpython compiles the python source code into intermediate bytecode, which is executed by the Cpython virtual machine. CPython is distributed with a large standard library written in a mixture of C and Python. CPython provides the highest level of compatibility with Python packages and C extension modules. All versions of the Python language are implemented in C because CPython is the reference implementation.

The Cython language is a superset of the **Python** language that additionally supports calling **C functions** and declaring **C types** on variables and class attributes. This allows the compiler to generate very **efficient C code** from Cython code. The C code is **generated once** and then compiles with all major C/C++ compilers in CPython 2.6, 2.7 (2.4+ with Cython 0.20.x) as well as 3.3 and all later versions

Some of the implementations which are based on CPython runtime core but with extended behavior or features in some aspects are Stackless Python, wpython, MicroPython.

Stackless Python – CPython with an emphasis on concurrency using tasklets and channels (used by dspython for the Nintendo DS)

Cython is an **optimising static compiler** for both the **Python** programming language and the extended Cython programming language (based on **Pyrex**). It makes writing C extensions for Python as easy as Python itself.

CPython has an ability to provide combined features of C and Python like:

* Python code can easily be written that calls back and forth from and to C or C++ code natively at any point
* easily tune readable Python code into plain C performance by adding static type declarations, also in Python syntax.
* use combined source code level debugging to find bugs in your Python, Cython and C code.
* interact efficiently with large data sets, e.g. using multi-dimensional NumPy arrays.
* quickly build your applications within the large, mature and widely used CPython ecosystem.
* integrate natively with existing code and data from legacy, low-level or high-performance libraries and applications.

1. **JPython**

Jython is an implementation of the Python programming language that can run on the Java platform. Jython programs use Java classes instead of Python modules. Jython compiles into Java byte code, which can then be run by **Java virtual machine**. Jython enables the use of Java class library functions from the Python program. Jython is slow as compared to Cpython and lacks compatibility with CPython libraries.

JPython or Jython is successor of CPython and has its pillars build on the Java Platform. It had its stable release for Python 2.7.1 in July 2017 and had its first release in January 2001.

Jython is especially suited for the following tasks:

* **Embedded scripting -** Java programmers can add the Jython libraries to their system to allow end users to write simple or complicated scripts that add functionality to the application.
* **Interactive experimentation** - Jython provides an interactive interpreter that can be used to interact with Java packages or with running Java applications. This allows programmers to experiment and debug any Java system using Jython.
* **Rapid application development** - Python programs are typically 2-10x shorter than the equivalent Java program. This translates directly to increased programmer productivity. The seamless interaction between Python and Java allows developers to freely mix the two languages both during development and in shipping products.

**Q2. Difference between Python 2 and Python 3 ?**

|  |  |
| --- | --- |
| **Python 2** | **Python 3** |
| * 1. It is more stable and transparent version of python programming language. | It is the future of Python designed to address the design flaws in the previous versions. |
| * 1. The print-syntax is treated as a statement rather than a function which requires text to be wrapped in parenthesis. | The print is explicitly treated as a function and replaces by the print() function in python3 which requires an extra pair of parenthesis. |
| * 1. ASCII string type is used by default to store strings. | Unicode is the implicit string type by default |
| * 1. It simply returns an integer to the nearest whole number when dividing two integers. | It makes integer division more intuitive using true division for integers and floats. |
| * 1. xrange() function reconstructs the sequence every time. | Xrange is replaced by rnage() function in python 3. |
| * 1. In python2 there were two methods to take input: * Raw\_input() * Input() | In python3 raw\_input() was removed and there is only one input() method left. |
| * 1. It had next() function and .next() method for iterate next element. | Have only next() function to iterate. |

**Q3. Difference between ASCII and Unicode?**

|  |  |
| --- | --- |
| **ASCII** | **UNICODE** |
| A character coding standard for electronic communication | A computer industry standard for consistent encoding, representation and handling of text in the world’s writing system. |
| Stands for American Standard Code for Information Interchange | Stands for universal Character set. |
| ASCII defines 128 characters, which map to the numbers 0–127. | Unicode defines (less than) 221 characters, which, similarly, map to numbers 0–221 (though not all numbers are currently assigned, and some are reserved). |
| ASCII is stored as 8-byte. | There are numerous ways of storing Unicode characters in byte sequences, such as UTF-32 and UTF-8. |
| ASCII is not standardized. | UNICODE is standardized. |
| It requires less space. | It requires more space. |
| It is used by C. | It is used by Java. |