**Advance Python Programming**

**Recommended duration: 25 hours**

**Course Description:**

This training program focuses on Advance Python programming Concepts

**Target Audience (who should attend):**

* Engineers who wish to learn advance python program to develop applications or to automate their applications/framework.
* Engineers who wish to prototype new applications.

**Pre-requisites:**

Participants should be comfortable with the following technologies:

* Basic programming background with good understanding of programming language ingredients that include variables and datatypes, flow control statements, and function/procedural programming paradigms.
* Knowledge of any scripting language would be beneficial.
* Basic python programming
* Knowledge of python list, tuples and dictionaries
* Knowledge of Python functions and modules

**Course Objectives:**

* Understand the OOPS features in python
* Understand class, objects and related concepts
* Learn how to connect to data and execute queries.
* Learn about exception handling and testing
* Learn about how to build multithreaded application using threading module
* Learn about pattern searching using re module
* Learn about web scrapping using xml and bs4 module.

**Course Agenda**

**Day1**

**Classes and Objects**

* Introduction to OOP using python
* Classes and class attributes
* Instances and instance attributes
* Binding and method invocation
* Composition, Subclassing and Derivation
* Inheritance
* Built-in functions for classes, instances and other objects
* An overview of built-in python classes and modules
* Magic functions in a class
* Introspection of objects using
* id(), type(), dir(),hasattr(), getattr(), setattr(), isinstance(), vars()
* Multiple inheritance
* Operator overloading
* \_\_lt\_\_(), \_\_gt\_\_(), \_\_le\_\_() and \_\_ge\_\_() methods
  + \_\_eq\_\_() and \_\_ne\_\_() methods.
  + \_\_add\_\_(), \_\_sub\_\_(), \_\_mul\_\_(), \_\_mod\_\_(), \_\_truediv\_\_() and \_\_pow\_\_() methods

**Day2**

**Errors and exception handling**

* Introduction to exceptions
* Detecting and handling exceptions
* Exceptions as Strings and Classes
* Raising exceptions
* Standard exceptions
* Python's Default Exception Handler
* Using Try/Except/Else/Finally Exceptions
* Generating User Defined Exceptions
* Exception Classes

**Testing, Debugging and Deployment**

* Using the unittest module for writing testcases
* Using doctest
* Documentation generation using pydoc
* Using pdb debugger framework and inspect
* Tracing python statement execution using
* Timing measurement using timeit module

**Day3**

**Implementing multi-tasking in Python programs**

* Introduction to threads and processes
* Creating and managing threads and processes
* Threading vs multiprocessing module
* Concurrency management using Lock and RLock
* Producer Consumer algorithm using Queue
* Implementing thread pool and process pool

**Connecting python to mysql database**

* Install mysql connector for python
* Python API’s for MySql
* Create sql table using Python
* Insert data into table using python
* Fetching data from table using python
* Updating data from table using python

**Day4**

**Regular Expression module**

* Importing re module
* Understanding Regular Expression syntax
  + Regex patterns notations
  + Metacharacters, metasymbols and escape sequences
  + Character classes and wild-cards
  + Quantifiers
  + Grouping and alternation
  + Extracting/Iterating pattern groups in match objects
* Splitting data using re module
* Matching in re module
* Findall in re module
* Searching in re module
* Special characters in re module
* Substitute in re module
* re.compile in re module

**Day5**

**Web Scrapping using xml and bs4 module**

* Urllib/requests modules
* Html, XML, JSON parsers
* Xml parsing using xml.Elementree
* Tags, attributes and texts in xml file
* Html parsing using Bs4 module
* Beautifulsoup in bs4
* Simple Task for WEB SCRAPPING

**Loggers**

* Different levels of verbosity
* Configuring loggers
* Formatting log messages

**Mini project discussion**