**Python Course**

**Original Requirement**

**1. Python debugging.**

**2. -**

**3. -**

**4. Object Oriented Programming and Python**

**5. Advanced Python (including these: network, file I/O, database, sockets, parallel, web programming)**

**6. IPython**

**7. -**

**8. Python Module/Library development**

**9. Object oriented programming**

**10. Algorithms and data structures programming**

**11. Test driven development (including mock testing)**

**Proposed Course Outline**

Session Timings:

Session 1: 9:30 am -10:45 am

Session 2: 11:00 am to 12:30 pm

Session 3: 1:15 pm to 2:45 pm

Session 4: 3:00 pm – 4:00 pm

Practice Programs: The course will be performed with live examples for each point mentioned in the course outline. The participants also can work on it at the same.

**Day 1: The Fundamentals**

Session 1:

* Introduction
* Use cases of python
* Comparison with other programming languages
* Python interpreter, console.
* Basic Data Structures
* Variables
* Dynamic typing
* Basic Programming constructs:
* Conditions : If, else, elif
* Loops: while, break, continue, for

Session 2:

* Interactive mode, file mode, dir, help, type
* Advanced Data Structures: Containers – list, tuple, dictionary, set
* Functions (Advanced):
  1. Syntax
  2. Invocation
  3. Parameter passing, Parameter passing with names
  4. Idea of pass by object
  5. Recursion and Nesting of functions

Session3:

* Object Oriented Concepts:
* Python objects
* Classes
* creation of object
* inheritance
* class attributes

Session 4:

* Modules
* concept
* .py and .pyc files
* identity, introspection
* Exception Handling:
* Significance
* Common exceptions
* Usage - try, except, raise, else, finally
* Custom Exceptions
* Best Practices

**Day 2: Productive Python 1**

Session 1:

* File handling:
* Open, read, write, delete files
* Shell utilities, os module
* Database connections: sqlite, mysql connector examples

Session 2:

* Database connections: SQL Alchemy ORM
* String Processing
* List Comprehensions, Dictionary Comprehensions
* Functional Programming Concepts: Map, Filter, Reduce.

Session 3:

* Python package
* 3rd party libraries, pip, easy\_install
* Using Debugger : pdb, stack trace.
* IPython
* Introduction
* IPython Console
* IPython Notebook

Session 4:

* Higher order Functions
* Decorators
* Exercises

**Day 3: Productive Python 2**

Session 1:

* Regular Expressions
* Testing: unittest, doctest, pytest, mock, nose, coverage

Session 2:

* Parsing
* ConfigParser
* HTML Parser
* XML Parser
* JSON Parser

Session 3:

* Networking
* SSH
* TELNET
* HTTP
* SERIAL PORT
* Socket Connections. Server – Client models.

Session 4:

* Web Development in Python:
  1. Various frameworks available in python World.
  2. Example: Flask Framework.
  3. Using Flask, Sqlalchemy, Jinja2 and WTForms to create a working web application.