Python 3.6

Workshop Details:

|  |  |
| --- | --- |
| Duration: | 6 Days |
| Description: | Developing Web and Windows applications using Python 3.6 |
| Objectives: | This session coverts the object oriented and procedural programming concepts of Python. This session helps the participants to understand various programming features of Python like File I/O, Threading, Networking, Database connectivity and integration with C language. |
| Participants’ Entry Profile: | Participants attending this course must have development experience on:   * Knowledge of basic Python programming concepts * OOPs and Procedure oriented programming * Knowledge of Scripts and services |
| Training Methodology: | The workshop will follow Synergetics methodology of   * Concept Visualization * Active Experimentation * Application Development   The workshop will be 100% Hands-On with each participant having access to system during the session |

Setup Requirements:

|  |  |
| --- | --- |
| Hardware and Software Requirements: | Participant’s as well as Trainer’s Machine are required to have:  Hardware   * Intel Pentium 4 [2+ GHz recommended] * 4 GB RAM * 50 GB HDD space * LAN connectivity * Good Internet connectivity and bandwidth   Software [Installed]   * Windows 8.1 or later * Python 3.6 * JetBrains PyCharm 2017 Community Edition |
| Training Lab Requirements: | Whiteboard 6 feet by 4 feet (minimum)  Whiteboard markers – Red, Blue, Green, Black  Video Projector (1024 X 768 resolutions) |
| Virtual Lab Requirements:  [Optional] | Virtual labs can be provided for participants, that provides completely configured platform to work with. |

Course Contents:

Day 1

1 Object-oriented programming with Python

* Creating classes and objects
* Encapsulating attributes and methods in classes
* Initializing objects with constructors
* Accessing and modifying attributes with methods
* Overloading operators
* Reusing functionality through inheritance
* Extending methods from base classes
* Overriding methods for dynamic behavior
* Tracing the scope in the namespace
* Enhancing functionality with class decorators

2 Functions in Python

* Functions and function parameters
* Positional, keyword and default arguments
* Implementing variable-length argument lists
* Global variables, scopes and return values
* Lambda functions
* Sorting collections of collections, dictionaries and Lists

3 Exception handling

* Error and exception handling
* Error types
* Gracefully handling exceptions
* Handling multiple exceptions
* Exception hierarchy
* Raising Exceptions
* User-Defined Exceptions
* Defining Clean-up Actions

Day 2

4 Reading and writing files

* Reading and writing text files
* Reading and writing binary files
* Importing OS module for directory management
* File objects
* Saving structured data with JSON

5 Regular expressions

* Compiling a pattern.
* Flags - ignorecase, dotall
* Working with multiple flags.
* Search vs match.
* Raw string notations.
* Special characters
  + Globbling characters
  + Anchors
  + Character sets
* Grouping

Day 3

6 Working with Databases

* Creating a Relational Database connection
* Instantiating cursors to access data
* CRUD Operations
* Retrieving desired datasets
* Updating with action statements

7 Developing web applications

* Request processing pipeline
* Mapping request to python scripts
* Developing MVC with python
  + Processing requests with Python controllers
  + Integrating the model with databases
* Producing HTML with Django templates

Day 4

8 Multi-threading in Python

* Understanding the Thread module
* Creating and starting new thread
* Synchronizing Threads
* Creating thread pool
* Performing Simple Parallel Programming
* Storing thread-state
* Implementing Publish/Subscribe messaging

9 Networking

* Interacting with HTTP Services
* Creating a TCP Server
* Creating a UDP Server
* Creating a Simple REST-Based Interface
* Understanding Event-Driven I/O
* Sending and receiving large arrays

Day 5

10 Unit Testing

* Introduction to Unit Testing
* Basic Example
* The unittest Module
  + *unittest*.TestCase
* Assertions
* Command-Line Interface
* Test Discovery
* Test Suites
* How to Skip Tests
* Integrating with doctest

11 Working with XML

* Parsing XML
* SAX vs DOM Parsers
* The xml.parsers.expat Module
* Handling Unicode
* The xml.sax and xml.sax.handler Modules
* Parsing with minidom
* Building XML with minidom
* Parsing XML with ElementTree
* Working with the Element Class
* The ElementTree Class
* Hands-on Lab Exercises

Day 6

12 Web Services

* Introduction to Web Services
* Calling SOAP Web Services
* A Restful Web Service Primer
* Consuming RESTful Web Services
* Python Web Frameworks

13 Mini Project

* Sample mini project