Python 3.6-Programming-Fundamentals

Workshop Details:

|  |  |
| --- | --- |
| Duration: | 3 Days |
| Description: | Python 3.6-Programming-Fundamentals |
| Objectives: | This session coverts the object oriented and procedural programming concepts of Python. This session helps the participants to understand various basic programming features of Python like variables, loops, functions, events, File I/O, Exception handling, python scripts, Object-oriented programming with Python, Regular Expression and much more. |
| Participants’ Entry Profile: | Participants attending this course must have development experience on:   * Knowledge of basic 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 Introduction to Python 3.6

* What is Python?
* Benefits and feature of Python
* Where to use Python
* Interpretation process
* Installing and configuring Python

2 Fundamentals of Python

* Data types and operators
* Immutable and mutable types
  + Numbers, strings and Tuples
  + Lists, Dictionary and Sets
* Loops in Python
* Type conversions
* String operations
  + Formatting, encoding and decoding
  + Secret codes
* Python Scripts on UNIX/Windows
* Command Line Parameters and Flow Control
* Built-in Functions, Math Operators and Expressions
* Conditional branching
  + If, elif, then, otherwise
  + Combining conditions

3 Repeating events

* Turtle
* Loops, nested loops and loop variables
* Loop an unknown number of times
* Looping issues

4 Sequences and operations

* Lists and tuples
  + Slicing, indexing, searching
  + Iterating sequences
* Operators in sequences
* X range() function
* List comprehensions
* Generator expressions

5 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

Day 2

6 Modules and packages

* Using modules
* Import statement for modules
* Controlling the Import of Everything
* Splitting a Module into Multiple Files
* Module searching path
* Package installation
* Referencing functions from modules by qualification
* Accessing the Standard Library

7 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

Day 3

8 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

9 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

10 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