COURSE NAME

Python

DURATION

3 Days

PRE-REQUISITES OF THE PARTICIPANTS

- None – Even the Participant without prior knowledge in programming can attend this training program.

COURSE OBJECTIVE

- Think intuitively in Python
- Gain in-depth understanding of Python's architecture, memory models, object-oriented features, input, and output details: stdio and file io
- Use interesting and extra-useful control of flow statement
- Master Python's simple and robust error handling methods
- Implement flexible function protocols
- Leverage built-in data types: Using them and inheriting from them in classes you design
- Deploy Python's list comprehensions, decorators, iterators, and generators
- Be industry ready to rapidly develop and deploy enterprise-grade Python applications using off-the-shelf libraries for all general-purpose usage

LAB REQUIREMENTS DETAILS

LAB SETUP DETAILS

OPERATING SYSTEM: Windows 7/10 or Any flavours of UNIX (Linux, HP-UX, Ubuntu, Solaris or MAC OS). PYTHON Supports Multiplatform.

RAM: 4GB RAM

HARD DISK SPACE: 100GB

SOFTWARES:

NOTE: In Flavours of UNIX, Python v2 will be installed by default and the packages will be installed during the training session.

a) Python v3.7.x – http://www.python.org/downloads

b) MySQL Database (If any other database is being used, this should be informed prior to the training)

c) PyCharm Community Edition – https://www.jetbrains.com/pycharm/download/

COURSE CONTENT

DAY – 1

PYTHON Introduction and Basics

- The Python Interpreter
- Working with Command Line/IDLE
- Python Data Types
- Built in Operators, Functions and Methods
- The 'type()' and 'dir()' functions
- Blocks and Indentation
- Scope of Variables

Conditional Statements and Iterators

- Simple 'if' and Simple 'if … else' statements
- Multilevel 'if … elif … else' statements
- Nested 'if … else' statements
- The 'for' and 'while' Iterators
- Loop Control Statements – break and continue
- The 'pass' statement

Data Structures

- Lists – Definition and Operations
- Tuples – Definition and Operations
- Dictionary – Definition and Operations
- Sets – Definition and Operations
- Membership and Identity Operators

Exercises

DAY – 2

Functions

- Defining the Function
- Function without Arguments
- Function with Fixed number of arguments
- Function with keyword arguments
- Function with Default arguments
- Function with variable length arguments
- Returning single/multiple values from the function
- Lambda Expressions
- Scope of the Variables

Modules and Packages

- What are modules?
- Import user defined modules
- Priority in reading the modules from the directories
- Creating '.pyc' file using '-m' option
- Advantage of '.pyc' file
- Controlling imports with __all__

- Pre-installed Modules
- Installing new Modules
- Python Repository
- Creating Packages
- Accessing the Packages

File I/O

- Creating the File
- Writing the contents to the file
- Using 'writelines()' method
- Explicit reading with read(), readline() and readlines()
- Reading Binary Files
- Use of 'with' statement

Regular Expressions

- The 're' module
- Patterns
- Anchors
- Range of Characters
- Metacharacters
- Quantifiers
- Choices
- Alternatives
- Groupings
- Dot (.) Character
- Character Class Escape Sequences
- Methods – compile, search, match, findall, split and sub

Exercises

DAY – 3

Classes and Objects

- Defining the Class
- The '__init__' and '__str__' methods
- Creating the Object
- The 'self' parameter
- Private and Public Attributes
- Concept of Encapsulation
- Concept of Polymorphism
- Concept of Inheritance
- Types of Inheritance – Single, Multiple, Multilevel and Hierarchial
- Magic Methods
- Method and Operator Overloading

Exception Handling

- Standard Exception Hierarchy
- Handling the Exceptions
- Generic Exceptions
- Handling Multiple Exceptions
- User Defined Exceptions

Standard Python modules

- Using the sys module
- Using sys.argv, sys.path, sys.version
- An overview on __builtin__ and __future__ modules
- Using the os module
- Filesystem/directory functions
- Basic process management functions
- Recursive directory iteration using os.walk
- Using the os.path module
- Determining basename, dirname, path manipulation
- File type/size/timestamp and other stat determination
- Using the time and datetime modules

Relational Database Interaction

- Use of Database in Real Time
- Modules to access the databases in Python
- Installation of DB Module, preferably for MySQL
- Connecting to the database
- CRUD Operations
- Transaction Management
- Connection and Cursor Objects
- Use of Query String Parameters

Decorators

- Usage of Decorators
- Implementing Decorator patterns
- Using Decorators
- Functions Decorators

Exercises