**Duration: 6 Days**

**Course Outline**

**Day 1**

**Module 1: Introduction to Python**

* Introduction to Python
* Features of Python
* Why Python

**Module 2: Simple Program**

* Python shell
* Python command
* Prepare source File and execute the code
* Print “Hello World”
* Comments
* Help command

**Module 3: Basic Language Consructs**

* Data types and Variables
* String type
* Format method
* Operators and Expressions
* Indentation

**Module 4: Control Structure**

* If
* While loop
* For loop
* Break & Continue Statements

**Module 5: Functions**

* Simple Function declaration
* Function with parameter
* Variable Scope
* The “global” statement
* Function with default arguments
* Function with return type

**Module 6: Data Structures**

* List
* Tuple
* Dictionary
* Sequences
* Set

**Module 7: Modules**

* What is module?
* Use of modules
* Import statement
* Global and local module
* Standard library module
* User defined modules
* The dir() Function

**Day 2**

**Module 8: Object Oriented Programming Concepts**

* Introduction to object oriented concepts
* Classes and Objects
* The “self” keyword
* Methods and Attributes
* Constructor and Destructor
* Instance and static member
* Class Inheritance
* Super keyword

**Module 8.1: Database Connectivity**

* Database Connection:
* Creating Database Table:
* INSERT Operation:
* READ Operation
* Update Operation:
* Performing Transactions:
* Handling Errors:

**Module 9: Files**

* What is File Input output?
* How to open a file
* How to close a file
* Read and write data to a file
* Pickle Module

**Module 10: Exception Handling**

* What is an Exception?
* Run time Exceptions
* try … except statements
* Multiple except statements
* Clean up statement (finally)
* Raised exceptions
* User defined exceptions

**Module 11: File and Directory handling**

* File I/O operations
* Built-in file and directory handling libraries
* fileinput
* stat
* filecmp and dircmp
* glob, zipfile and tarfile
* pickle and shelve modules
* Serialization using json
* Pyxl for Excel, Pypdf and Pydoc

**Module 12: Regular Expression**

* Introduction to regexps
* Special symbols and characters for RE
* Metacharacters and Metasymbols
* Character classes
* Quantifiers and Positions
* Searching/Matching a string/pattern
* Search and replace operations
* Splitting and joining strings
* Practical examples

**Day 3**

**Module 13: Itertools and Collections framework**

* Iterators and generators
* Collections
* Deque
* defaultdict

**Module 14: Testing**

* Using the unittest module for writing testcases
* Using doctest
* Regressing testing using test module

**Module 15: XML Parsing**

* Introduction
* Element Tree Module
* Dom
* Minidom

**Module 16: Socket Programming**

* What is Socket Programming?
* What is protocol?
* Types of protocol
* What is “host” and “port”
* socket class
* methods of socket class
* type of socket
* Module 17: Threading
* What is process?
* What is thread?
* Thread class
* Life cycle of a thread
* Methods of thread class
* Time class
* Synchronization
* Lock class methods

**Module 18: Decorator**

* About Decorators
* Simple Function Decorators
* Decorator Example
* Classes as Decorators
* Decorator Arguments
* Exercises

**Module 19: Decorator**

* Working with REST APIs using Python
* How to use SSH API’s in python to connect with Python
* How to use SNMP in python
* Django module

**Day 4**

**Module 20**

**Execution steps**

* Environment setup(python, Pycharm, Selenium and etc.,)
* Introduction with Python and Pycharm
* First Script on Selenium

**Selenium- Web Driver**

* Introduction to Web driver and Remote vs. Local
* Guide to install Web driver
* Creating your first script on Web driver
* Accessing Forms in Web driver
* Accessing Links and Table content in Web driver
* Remote web driver

**Automation Framework**

* Advanced Web element access method- Contains, Sibling, Ancestor and etc.,
* Framework designing methods
* Framework adaptation
* Feature Testing Automation
* Report Generation out of Automation
* Real time Automation and the Challenges

**PyAutoGUI – Controlling Mouse and Keyboard**

* Introduction to pyautogui
* Accessing Flash content using pyautogui
* Controlling Keyboard and Mouse events on web driver

**Day 5**

**Image Handling**

* Introduction to `scipy`, `numpy` and `matplotlib`
* Using `pandas` for data manipulation
* Using Plotly
* Memory Management
* List Comprehensions
* Decorators
* Shallow and Deep Copy

CSV modules

* CSV format
* Tab delimiters instead of ‘,’
* Variants of CSV
* Reading csv files
* Dict references

 Writing CSV files

JSON

* JSON format
* Field data and formats
* JSON data parsed to dict’s
* Modifying dict data
* Writing out dict data to JSON
* BSON (Binary JSON to save space, security)

**Day 6**

We will be working on micro-projects dealing

a) CSV - Import CSV files with statistics.

b) Excel files - Import Excel files with data

c) Python for Report Writing - Word / Power-point etc

d) Manipulating large data. - Creation of charts and reports from data

e) Creation of Power-point presentation from data provided.

Case Study (E-commerce Application)

Setup a webserver that takes our inputs through http for below listed functions. The webserver will need to  provide a 2-factor authentication option for the users before taking their inputs,  The inputs may vary based on each of below listed tasks.

* Software tools for a Django development environment
* Database abstraction (Object Relational Mapper)
  + Creation of a model
  + Model validation
  + Export and import of data (via JSON)
  + Working with an ORM on the commandline
  + Using Django's admin
  + Defining URLs at one place (Routing)
  + In-depth explanation of function based views
  + How to use class-based views
  + Creating templates using Django's template engine
  + How to handle static files in Django
  + Managing Django's settings
  + Different debugging techniques
  + Logging
  + Forms and their validation
  + Sending email using a form
  + Pagination
  + Authentication
  + Sessions and messages
  + Security
  + Write your own middleware
  + Create your own tags and filters