Basic to Advance Python Programming

Total Course Duration – 40 hours

WEEK 1:

1) Python Introduction

- ➢ Unique Features & Advantages of Python
- ➢ Python Installation – Windows and Linux
- ➢ Python Identifier, Keywords and Indentation
  Keywords – 33 [import keyword, keyword.kwlist]
- ➢ Data Types Supported by Python
  - Integer - immutable
  - String - immutable
  - List - mutable
  - Tuple - immutable
  - Set - mutable
  - Dictionary - mutable
  - File
- ➢ Getting user input(input and raw_input(obsolete in 2.7))
- ➢ Loop and control flow
  - for loop
  - while loop
  - break, continue & pass
  - if else and elif
- ➢ Python Build-ins – 65 build ins(will cover important ones)

ASSIGNMENT

WEEK 2:

2) Important Python Modules:

- os
- sys
- subprocess
- copy
  1. Shallow Copy
  2. Deep Copy
- glob
- collections
  1. defaultdict
  2. OrderedDict
  3. Counter
  4. Namedtuple
  5. deque
- itertools:
  1. Infinite Iterators:
     i. count,
     ii. cycle
     iii.repeat
  2. Combinatoric generators:
     - i. product(p, q, ... [repeat=1]) --> cartesian product
     - ii. permutations(p[, r])
     - iii. combinations(p, r)
     - iv. combinations_with_replacement(p, r)

3) Functions: defining and calling functions.

- Anonymous Functions a.k.a. lambda functions
- Functions as FCO(First Class Object)
- Higher Order Functions
- Closures Functions
- Function Decorators

4) Exceptional Handling in Python

- What are Exceptions?
- Handling an Exception
- try…except…else. Finally
- Python Standard Exceptions
- Raising an Exceptions

ASSIGNEMENT

WEEK 3:

5) Generators & Coroutines

6) Python logging module

7) Python Package Manager (pip installer):

- ➢ How to install third party packages
- ➢ How to list all the installed packages.
- ➢ How to install package on multiple systems
- ➢ How to see outdated packages and update them

8) Argparse module for parsing arguments and command line interface:

- ➢ Positional and Optional Arguments
- ➢ Creating Parser
- ➢ Customizing the parser
  Argument Parser Object:
  - i. prog
  - ii. usage
  - iii. description
  - iv. epilog
  - v. prefix_chars
- ➢ Adding arguments
  Add argument object:
  - i. name or flags
  - ii. action
  - iii. nargs
  - iv. const
  - v. default
  - vi. type
  - vii. choices
  - viii. required
  - ix. metavar
  - x. dest
- ➢ Parsing arguments

ASSIGNEMENT

WEEK 4:

9) Object Oriented Python Programming (OOPs)

- Creating Classes and Objects
- Accessing Attributes
- Magic Methods
    - i. __init__
    - ii. __repr__
    - iii. __str__
    - iv. __add__
    - v. __len__
    - vi. __new__
    - vii. __call__

10) OOPs continued…

- Class inheritance
    - Multiple Inheritance
    - Multi Level Inheritance
- Overriding methods
- Class Method Resolution Order
- Super Keyword
- Special Class Methods
    - Iterators(__iter__ and __next__)
    - Context Managers(__enter__ and __exit__)
- SLOTS

11) Python Descriptors: Data & non-data descriptors

- property
- classmethod
- staticmethod

12) Class as Function Decorator

ASSIGNEMENT

WEEK 5:

13) Regular Expression (re module)

- The Match Function
- The search Function
- Regular Expression characters:
  - i. Caret (^)
  - ii. Doller ($)
  - iii. Dot (.)
  - iv. Asterisk (*)
  - v. Plus (+)
  - vi. Question Mark (?)
  - vii. Curly braces ({ })
  - viii. Forward slash (\ )
  - ix. Square brackets ([ ])
  - x. Pipe (|)
  - xi. Rounded Brackets (())
- Special Sequence:
  - i. \number
  - ii. \A
  - iii. \Z
  - iv. \b
  - v. \B
  - vi. \d
  - vii. \D
  - viii. \w
  - ix. \W
- Grouping
- Search and Replace (re.sub)
- Find All (re.findall)
- Split (re.split)

14) Python Multithreading (threading module)

- What is multithreading?
- Starting a new thread
- The Threading Module
- Synchronizing Threads

ASSIGNEMENT

WEEK 6:

15) Applications of Python

- Python Spreedsheet Interface(openpyxl module)
    1. Creating xlsx workbook and sheet
    2. Naming sheet and inserting data
    3. Inserting chart
- Python XML interface(lxml module)
    1. Parsing XML File
    2. Iterating all elements
    3. find and findall methods
    4. Modifying XML file
        1. Adding new attribute
        2. Adding new children
- Python Json Interface(json module)
    1. Loading json object from string
    2. Loading json object from file
    3. Dumping json object to string
    4. Dumping json object to file
- Python ssh module for remote command execution(paramiko module)
    1. Connecting to remote host
    2. Executing command on remote host and getting output
    3. Transferring file from local to remote - vice versa
    4. Using sftp
- Socket Programming Using Python(socket module)
    1. Creating server and client application
    2. Creating Echo server
    3. Creating Chat applications
- Python Database Access(pymysql module)
    - Creating Database Connection
    - CREATE, INSERT, READ, UPDATE and DELETE Operation
    - Performing Transactions
    - Handling Database Errors
    - Disconnecting Database
- Python GUI Programming (tkinter module)
- Python Web Development(Server side) (flask module)
- Python Web Scraping using requests, urllib and beautifulsoup
- Introduction to Pandas

ASSIGNEMENT