Python Programming

Course Name	Python Programming
Duration	5 days
Training Outcomes	• Participants should be able to engage in technical interaction with clients • Participants should be able to deliver work on project assignments

DAY - 1

Theory Session:

- About Python and software installation
- Brief about IDEs like pycharm, notebook, vs code, anaconda distribution etc
- Numbers
  - float
  - numbers.Rational
- Arithmetic operations

Practical Session:

Write 2 to 3 programs which demonstrate below arithmetic operators with numbers

• +	• *

• -	• /
• //	• %
• **	

Theory Session:

- Built-ins
- Strings
  - Indexes
  - Slicing
  - Negative indexes
  - Unicode
  - Find() method
  - Index() method

Practical Session:

Program on strings using real time data using methods like split(), find(), index(), strip() etc

Problem Statement:

- Take real time data in string. Sample data example:

  ‘123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/mypics.gif HTTP/1.0" 200 6248 "http://www.abcxyz.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)"’

- Writing program to extract below information using string class methods
  - IP
  - DATE

  - PICS
  - URL

Theory Session:

- List type
- Tuple type
- Dictionary type
- Set type

DAY - 2

Theory Session:

- Conditional statement ‘if’

Practical Session:

Write a program which demonstrate comparison statements (> <) and Boolean operations coverage during 'if' conditional statements

Theory Session:

- Loops: ‘for-loop’

Practical Session:

Program on for-loop using real time data

Problem Statement:

- Increase the sample data used in program-1

- Using for-loop extract the information
- Produce below outputs
  - Output-1: list of lists

    [[ip, date, pics, url], [ip, date, pics, url], [ip, date, pics, url]]
  - Ouput-2: list of tuples

    [(ip, date, pics, url), (ip, date, pics, url), (ip, date, pics, url)]
  - Output-3: dictionary

    {‘0’: (ip, date, pics, url), ‘1’: (ip, date, pics, url)}

Theory Session:

- Loops ‘while’

Practical Session:

Program on while-loop using real time data

Problem Statement: Write 2nd program using while loop

Theory Session:

- Text file operations
  - Reading from text files
  - Writing to files

Practical Session:

Program on file operations

Sample file: in text file keep the data used in program-1 to program-4

Problem Statement:

- Extract the
    - IP
    - DATE
    - PICS
    - URL
- Produce .txt and .csv reports

Theory Session:

- Functions

  Theory Session:

    - Functions without arguments

  Practical Session:

  Write a program which demonstrate functions without arguments

  Theory Session:

    - Functions with return values

  Practical Session:

  Write a program which demonstrate functions with return values

  Theory Session:

    - Functions with positional arguments

Practical Session:

Write a program which demonstrate functions with positional arguments

Theory Session:

- Functions with keyword or named arguments

Practical Session:

Write a program which demonstrate functions with keyword arguments

Theory Session:

- Variable scopes
  - Local
  - Enclosed
  - Global
  - Built-in

Practical Session:

Write a program which demonstrate functions with local, enclosed and global variable

Practical Session:

Program on functions

Sample file: in text file keep the data used in program-1 to program-4

Problem Statement:

- Write a positional argument function which takes arguments as data file path, extract information, return extracted information in list of tuples
  - Expected return value format: [(ip, date, pics, url), (ip, date, pics, url), (ip, date, pics, url)]
- Write a keyword or named argument function which takes arguments as data file path, extract information, return extracted information in dictionary
  - Expected return value format: {'0': (ip, date, pics, url), '1': (ip, date, pics, url)}

DAY - 3

Theory Session:

- Classes and OOP
  - Class object and instance objects
  - Class variables and instance variables
  - Class methods and instance methods
    - Difference b/n class and instance methods
    - When to use this method
  - Static methods
    - Difference b/n class, instance and static methods
    - When to use this method
  - Multilevel inheritance
    - write program to demonstrate inheritance concepts
  - Multiple inheritance

- MRO Overview
- Operator overloading
- Brief about Abstract classes

Practical Session:

Program on classes

Sample file: in text file keep the data used in program-1 to program-4

Problem Statement:

- Write a class with below methods
    - __init__ method to read the data and keep in instance variable
    - Write an instance method to extract IP
    - Write an instance method to extract DATE
    - Write an instance method to extract PICS
    - Write an instance method to extract URL
    - Write an instance method to extract ALL
        - Create instance variable ‘all_data’ to store extracted data
    - __add__ special method to use + to add port number
    - Write class method to set ‘location’
        - class variable ‘location’
    - Write class method to get ‘location’
- Inheritance hands-on program writing
    - Extend above class and add below 2 new methods
        - To_csv
        - To_txt method to write extracted data txt and csv files

- MRO of Python 3
    - Demonstrate MRO using above class

Theory Session:

- Exceptions handling
    - try and except block
    - try-except with exception classes
    - try-except-else blocks
    - try-except-finally block
    - ‘assert’ statement
    - ‘raise’ statement
    - User defined exception classes

Practical Session:

Program on exceptions handling

Problem Statement: Write program-4 using exceptions handling which handles the exceptions like FileNotFoundError etc

DAY - 4

Theory Session:

- Modules and packages
    - Creating modules
    - Creating packages
    - write program on importing one module into another module, importing module from package
    - About pypi and installing libraries

Practical Session:

Create module and packages for the functions and classes defined during practical session on functions and classes. Import created module and packages in another program using ‘import’ and ‘from-import’

Theory Session:

- Virtual Environment
    - Creating virtual environment
    - Activate
    - Deactivate
    - Delete

Practical Session:

Demonstration of Virtual environments create, activate, deactivate and delete using ‘pyenv’

Theory Session:

About Beautifulsoup library and installation

Library installation from pypi using pip

Practical Session:

Web scraping using Beautifulsoup

Problem Statement: Get any freely available website or create html file

Using above sample data used earlier and pull some of the tags data, tags attribute, find_all elements etc

Theory Session:

About Regular expression and its library ‘re’

- re.match
- re.search

Practical Session:

Practical session on ‘re’ meta characters

- [ ]
- \
- .
- ^
- $
- *
- +
- ?
- {}
- |
- ()

Practical Session:

Practical session of re.match

Use the data present in text file provided above, extract

- IP
- DATE
- PICS
- URL

Using re.match

Practical Session:

Practical session on re.search

Use the data provided above to search based on the pattern for the data provided above.

Using re.search()

DAY - 5

Theory Session:

- SQLite Databases
  - Creating database
  - Executing queries on the database

Practical Session:

Program on SQL Databases

Problem Statement: Use the data present in text file provided above, extract

Using regular expression, send extracted data to SQLite database table

- Creating the database

- Creating the tables
- Executing the queries

Theory Session:

Introduction to pandas library, DataFrame.

Practical Session:

Program on Data Analysis and Data Preprocessing using pandas

Problem Statement: Get data from above database, create pandas DataFrame

- Produce different report like .txt, csv, xlsx, xml, json etc
- Try methods like count, value_counts, groupby, dropduplicates, fillna etc

Theory Session:

Introduction to seaborn

Practical Session:

Program on plotting graph on above DataFrame data

Problem Statement: Plot the graph on DataFrame created above

Theory Session:

Introduction to flask framework

Practical Session:

Create REST-API using flask

Problem Statement: Create REST-API to which supports CRUD operations

  - GET
  - POST
  - PUT
  - PATCH
  - DELETE

Theory Session:

- GitHub
  - GitHub code push
  - clone repository,
  - comparing local vs remote,
  - create branch,
  - conflict resolution

Practical Session:

Create a GitHub repository and perform the above operations

Theory Session:

- Logging module

Practical Session:

Write any of the above program to use logging module for logging to file and also output stream. Use levels INFO, DEBUG, ERROR, WARNING, CRITICAL wherever it is required

Theory Session:

- Generators

Practical Session:

Rewrite a function which is developed earlier to make use of the generator

Theory Session:

- Decorators

Practical Session:

Write a decorator which has a common functionality which can be attach to other functions.

Theory Session:

- Multithreading

Practical Session:

Create 2 or more threads using Thread class present in threading module and use methods like start, join, run etc

Brief introduction about:

- Numpy
- Scikit
- GUI desktop app

- Tkinter
- bash command
- PostgreSQL
- Authentication
- asynchronous programming