Intermediate Python Programming

Course Duration: 3 Days

Course Objectives:

- To deepen participants' understanding of Python programming concepts.
- To provide in-depth knowledge of Object-Oriented Programming (OOP).
- To introduce networking and multithreading in Python.

Software Setup:

- Participants should have Python 3.x installed on their laptops.
- PyCharm Integrated Development Environment (IDE).

Day 1: Foundations of Intermediate Python Programming

Session 1: Review of Python Basics

- **Module 1: Python Basics**
  - Recap of variables, data types, and basic operations.
  - Control flow: if statements, loops (for and while).
  - Functions and their importance.
- **Module 2: Working with Data Structures**
  - Lists, tuples, and dictionaries.
  - List comprehensions.
  - Iterating through data structures.

Session 2: File Handling and Modules

- **Module 3: File Handling**
  - Reading and writing files.
  - CSV and JSON handling.
  - Error handling with try...except.
- **Module 4: Functions and Modules**
  - Defining and using functions.
  - Creating and importing modules.
  - Best practices for function design.

Day 2: Object-Oriented Programming in Python

Session 1: Introduction to OOP

- **Module 5: Introduction to OOP**
  - Understanding classes and objects.
  - Constructors and destructors.
  - Class attributes and methods.
- **Module 6: Inheritance and Polymorphism**
  - Creating and using subclasses.

- Method overriding and method overloading.
- Achieving polymorphism in Python.

Session 2: Encapsulation and Data Abstraction

- **Module 7: Encapsulation and Data Abstraction**
  - Encapsulation principles in Python.
  - Data hiding.
  - Abstract classes and interfaces.

Day 3: Additional Topics in Python

Session 1: Networking in Python

- **Module 8: Networking in Python**
  - Introduction to network protocols (TCP, UDP).
  - Socket programming in Python.
  - Creating client-server applications.

Session 2: Multithreading in Python

- **Module 9: Multithreading in Python**
  - Understanding threads and concurrency.
  - Creating and managing threads.
  - Thread synchronization and communication.