**Mastering Advanced Python Development**RectangleRectangle

Duration: 4 Days

Description

Dive deep into the intricacies of Python programming with our comprehensive course, "Mastering Advanced Python Development." This course is tailored for experienced Python developers looking to enhance their skills in object-oriented programming, Python features, unit testing, debugging, design patterns, web services, optimization, package management, and concurrency.

Audience

This course is tailored for intermediate to advanced Python developers seeking to enhance their skills in complex Python concepts and practical applications.

Objectives

In this Python course, you will learn:

* Advanced Object Oriented Programming
* Python features like generators and context managers
* Unit testing and debugging techniques
* Design pattern implementation
* REST API interaction
* Performance optimization strategies
* Concurrent programming in Python

Prerequisites

This course requires:

* An intermediate understanding of Python programming concepts
* A basic understanding of web technologies

Outline

* Module 1: Object Oriented Programming in Python
  + Basic Principles of Object Oriented Programming
    - Encapsulation
    - Inheritance
    - Polymorphism
    - Abstraction
  + Classes and Objects
    - Define a Class
    - Create Instances of a Class
    - Instance Methods, Class Methods, and Static Methods
  + Properties
    - Add Properties to a Class Using @property Decorator
    - Use Getter, Setter, and Deleter with @property Decorator
  + Inheritance and Subclasses
    - Define a Class that Inherits from Another Class
    - super() Function
    - Method Overriding
  + Abstract Base Classes(ABCs)
    - abc Module and Concept of Creating Abstract Classes
    - Abstract Methods with @abstractmethod Decorator
    - Purpose of Abstract Classes
* Module 2: Python’s Features
  + Best Practices
    - Pythonic Conventions
  + Monkey Patching
    - Dynamically Modify or Extend a Module, Class, or Method at Runtime
    - Useful Scenarios
  + Generators
    - Create Generators
    - The yield Keyword
    - Iterating over Generators
    - Advantages and Common Use Cases
  + Exceptions
    - Handling Exceptions
    - Create and Raise User-Defined Exceptions
  + The with Statement
    - Context Managers
    - Syntax of the with Statement
    - Exception Handling Simplification
* Module 3: Python Unit Testing
  + Python’s unittest Framework
    - unittest Structure and Writing Test Cases with unittest
    - Test Classes that inherit from unittest.TestCase
    - setUp and tearDown Methods
    - Assertions
  + Mocking and Patching
    - unittest.mock
    - mock Objects
* Module 4: Debugging and Logging
  + IDE Debugging Tools
    - PyCharm and Pylint
    - Breakpoints, Watch Windows, and Stepping through Code
  + Logging
    - Python’s Logging Module
    - Configure Log Levels
    - Write Log Messages
* Module 5: Design Patterns
  + Structural Patterns
    - Decorator Pattern
    - Proxy Pattern
  + Behavioral Patterns
    - Template Method Pattern
    - Observer Pattern
* Module 6: RESTful Web Services and Clients
  + Data Formats
    - JSON
    - XML
* Module 7: Python Optimization
  + timeit Module
  + cProfile
  + pstats Module
  + Appropriate Use of Data Structures
  + List Comprehensions
  + PyPy Python Interpreter
* Module 8: Managing Python Packages and Environments
  + pip Python Package Manager
  + Python Versions and Porting
  + Packaging and Distribution Package
  + Isolated Python Environments
    - virtualenv
  + Using a Local Repository
* Module 9: Concurrency in Python
  + Threads and Locks
  + Subprocesses
  + Processes and Queues
  + Parallelism