**Python Programming (Advanced)**

**Course Overview**

The **Advanced Python Programming** course is designed for learners who have a strong foundation in Python basics and are ready to explore its more powerful, complex, and real-world application features. This course delves into object-oriented programming, advanced data structures, functional programming, modules and packages, file operations, error handling, decorators, generators, multithreading, and working with external libraries.

Additionally, learners will gain hands-on experience with popular modules used in real-world development, such as NumPy, Pandas, Requests, and Regex, as well as concepts like APIs, working with JSON, and integrating databases using SQLite.

**Course Objectives**

By the end of this course, learners will be able to:

* Apply object-oriented programming concepts in Python projects.
* Work with advanced data structures like sets, dictionaries, and nested collections.
* Use lambda functions, list comprehensions, and other functional programming tools.
* Implement exception handling, context managers, and custom error classes.
* Use decorators and generators for efficient and modular code.
* Interact with databases and external files (CSV, JSON, APIs).
* Perform multithreading and understand concurrency in Python.
* Build modular, scalable, and production-ready Python applications.

**Syllabus Breakdown**

**Module 1: Object-Oriented Programming in Python**

* Classes and objects
* Constructors (\_\_init\_\_)
* Instance and class variables
* Methods: instance, class, and static
* Inheritance and method overriding
* Encapsulation and abstraction
* Special (magic/dunder) methods like \_\_str\_\_, \_\_repr\_\_

**Module 2: Functional Programming and Comprehensions**

* Lambda expressions and anonymous functions
* map(), filter(), and reduce() functions
* List, dictionary, and set comprehensions
* Nested comprehensions
* Use of any() and all() in data validation

**Module 3: Exception Handling & File I/O**

* Advanced error handling with multiple except blocks
* Creating custom exception classes
* The finally block and resource cleanup
* Context managers and the with statement
* File operations with CSV, JSON, and text
* Working with file directories using os and shutil modules

**Module 4: Decorators and Generators**

* Understanding closures and nested functions
* Writing and applying decorators
* Chaining decorators
* Generators and yield keyword
* Use cases of generators in large data handling

**Module 5: Modules, Packages & Virtual Environments**

* Creating and importing modules
* Organizing code into packages
* \_\_init\_\_.py and namespace management
* Installing and managing packages via pip
* Setting up and using virtual environments with venv or virtualenv

**Module 6: Working with APIs, JSON, and Web Data**

* Fetching data using the requests library
* Parsing and writing JSON files
* Consuming RESTful APIs (GET, POST, etc.)
* Handling authentication and headers
* Creating simple CLI tools using API data

**Module 7: Database Programming (SQLite)**

* Introduction to databases and SQLite
* Connecting Python with SQLite
* Executing SQL queries from Python
* Creating tables and inserting, updating, deleting records
* Handling exceptions and transactions in DB code

**Module 8: Multithreading and Concurrency**

* Threads and the threading module
* Creating and running threads
* Thread synchronization using locks
* Thread-safe operations
* Introduction to multiprocessing vs threading

**Module 9: Popular Libraries & Mini Projects**

* Using NumPy and Pandas for data manipulation
* Regular Expressions (re module)
* Working with dates and times (datetime and calendar)
* Mini Project 1: Contact Book with File Storage
* Mini Project 2: API-based Weather Report App
* Mini Project 3: Multithreaded File Downloader

**Career Opportunities**

Advanced Python skills are highly sought-after in diverse tech domains. After completing this course, learners will be prepared for:

* **Python Developer (Advanced/Backend)**
* **Full Stack Developer (Python-based)**
* **API Developer / Automation Engineer**
* **Data Analyst / Data Engineer (entry-level)**
* **Machine Learning Engineer (with additional specialization)**
* **DevOps Scripting Specialist**
* **Software Engineer / Application Developer**

Industries including **banking, finance, education tech, healthcare, AI startups, and government IT projects** all seek professionals proficient in Python at an advanced level.