PURPLE GRIFFON
The IT Service Management Training & Consultancy Specialists

Course Title: Advanced Python: Best Practices And Design Patterns	**Course Duration**: 4.0 Days
Exam: Included	**Exam Type**: Proctored Exam
Qualification: Advanced Python: Best Practices And Design Patterns Certificate	

Course Syllabus

Our Advanced Python: Best Practices And Design Patterns training course covers the following Modules:

Module 1: Object-Oriented Programming in Python

In this module, you will learn how to:

- Extend classes to define subclasses
- Add properties to a class
- Define abstract base classes

Module 2: Exploring Python Features

In this module, you will learn how to:

- Write "Pythonic" code
- Modify code dynamically with monkey patching
- Process large data structures efficiently with generators
- Handle exceptions
- Raise user-defined exceptions
- Reduce code complexity with context managers and the "with" statement

Module 3: Verifying Code and Unit Testing

In this module, you will learn how to:

- Develop and run Python unit tests
- Simplify automated testing with the Pytest package
- Verify code behaviour
- Mock dependent objects with the Mock package
- Use mock objects to verify code behaviour when exceptions occur

Module 4: Detecting Errors and Debugging Techniques

In this module, you will learn how to:

- Log messages for auditing and debugging
- Check your code for potential bugs with Pylint
- Debug your Python code
- Extract error information from exceptions

- Trace programme execution with the PyCharm IDE

Module 5: Implementing Python Design Patterns

In this module, you will learn how to:

- Implement the Decorator pattern using @decorator
- Control access to an object with the Proxy pattern
- Lay out a skeleton algorithm in the Template Method pattern
- Enable loose coupling between classes with the Observer pattern

Module 6: Interfacing with REST Web Services and Clients

In this module, you will learn how to:

- Build a REST service
- Generate JSON responses to support Ajax clients
- Send REST requests from a Python client
- Consume JSON and XML response data

Module 7: Measuring and Improving Application Performance

In this module, you will learn how to:

- Time execution of functions with the "timeit" module
- Profile programme execution using "cProfile"
- Manipulate an execution profile interactively with "pstats"
- Efficiently apply data structures, including lists, dictionaries, and tuples
- Map and filter data sets using comprehensions
- Replace the standard Python interpreter with PyPy

Module 8: Installing and Distributing Modules

In this module, you will learn how to:

- Install modules from the PyPi repository using "pip"
- Port code between Python versions
- Package Python modules and applications
- Establish isolated Python environments with "virtualenv"
- Build a distribution package with "setuptools"
- Upload your Python modules to a local repository

Module 9: Concurrent Execution

In this module, you will learn how to:

- Create and manage multiple threads of control with the Thread class
- Synchronise threads using locks
- Launch operating system commands as subprocesses
- Synchronise processes with queues
- Parallelise execution using process pools and Executors

Course Overview

Our four-day Advanced Python: Best Practices And Design Patterns training course will teach you the importance of Advanced Python programming and the accompanying concepts and methods of Best Practices and Design Patterns. Understand the methods and following the principles of this course will help you write efficient, maintainable, and scalable code.

- Best practices in advanced Python programming refer to a set of guidelines and recommendations that experienced Python developers follow to write high-quality code. These practices encompass various aspects of coding style, organisation, error handling, testing, documentation, and performance optimisation.
- Design patterns, on the other hand, are reusable solutions to common software design problems. They provide a blueprint or template for solving recurring issues in software architecture and design.

Course Learning Outcomes

Our Advanced Python: Best Practices And Design Patterns training course will teach you to become proficient in the following:

- Employ design patterns and best practices in Python applications
- Exploit the object-oriented programming features in Python for stable, reliable programs
- Create and manage concurrent threads of control
- Generate and consume REST web service requests and responses
- Implement Gang of Four (GoF) design patterns to solve commonly recurring software design problems

Course benefits:

- Unit test, debug, and install Python programmes and modules
- Profile programme execution and improve performance
- Apply advanced Python programming features for efficient, reliable, and maintainable programmes
- Gain knowledge and skills applicable to all Python environments, including Microsoft Windows, macOS, and all Linux and UNIX distributions
- Test your knowledge in the included end-of-course exam
- Continue learning and face new challenges with after-course one-on-one instructor coaching

Audience

Our Advanced Python: Best Practices And Design Patterns training course will benefit several individuals and organisations including but not limited to:

- Experienced Python Developers
- Intermediate Python Developers
- Software Architects
- Technical Leads and Team Leads
- Open-Source Contributors
- Data Scientists and Data Engineers
- Students and Python Enthusiasts

Entry-Level Requirements

Our Advanced Python: Best Practices And Design Patterns training course requires attendees to have experience at the level of our Introduction To Python training course.

Recommended Reading

There is no recommended reading for our Advanced Python: Best Practices And Design Patterns training course.

What's Included

Our Advanced Python: Best Practices And Design Patterns training course contains the following:

- 4-day instructor-led training course
- After-course instructor coaching benefit
- End-of-course exam included
- Pre-reading
- Course Manual
- Quizzes
- Exercises

Exam Information

Advanced Python: Best Practices And Design Patterns Exam:

- Format: Multiple Choice
- Questions: 40
- Pass Mark: 70%

What's Next

Attendees may enjoy our three-day Introduction To Python training course as well as our one-day Introduction To Python For Data Analytics training course.

Our three-day Introduction To Python training course will provide a foundation for learning and understanding the Python programming language. Python is a versatile and widely used programming language known for its simplicity, readability, and a large number of libraries and frameworks that make it suitable for various applications. We will teach you how to use Python's features, standard library modules, and third-party software packages.

Our one-day Introduction To Python For Data Analytics training course will help you gain the skills you need to analyse and visualise data with Python. Aiding in helping you learn the fundamentals of Python programming with a focus on data analytics, and work with popular statistical computing libraries — like numPy, Pandas, sciPy, and Scikit-learn — that allow you to begin analysing data to answer key business questions.

- The eBooks and on-demand courses provided with this offering are a great way to explore your interest in the topics covered in the instructor-led course.

Additional Information

Our Advanced Python: Best Practices And Design Patterns training course offers several benefits to individuals and organisations including but not limited to:

- **Code Quality and Readability**: Learning best practices ensures that your code follows established standards, making it more readable and maintainable.
- **Scalability and Maintainability**: Applying design patterns helps in building scalable and extensible software systems.
- **Efficiency and Performance**: Advanced Python techniques, such as optimising algorithms and leveraging data structures, can significantly improve the efficiency and performance of your code.



- **Error Handling and Robustness**: Understanding error handling strategies and exception management techniques helps you build robust applications.
- **Reusability and Productivity**: Learning design patterns enables you to create reusable components that can be applied across projects.
- **Collaboration and Teamwork**: Following best practices and design patterns improves collaboration within development teams.
- **Career Advancement**: Proficiency in advanced Python best practices and design patterns distinguishes you as a skilled and experienced developer.
- **Adaptability and Future-Proofing**: Learning advanced techniques and patterns equips you with a broader skill set and the ability to adapt to new technologies and frameworks.