



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Python Unit Testing with Unittest

Duration

2 days

Description

This comprehensive course is designed for computer programming professionals to master the art of Test-Driven Development (TDD) using Python. The course begins with an overview of TDD, its benefits, and challenges, and then dives deep into the principles of TDD inspired by “Uncle” Bob Martin. Participants will learn about different types of testing, components of testing, and get hands-on experience with Python unit testing frameworks. The course also covers assertions, mocking, organizing tests, code coverage, and the role of testing in CI/CD. The course concludes with a summary of key concepts, Q&A, and resources for further learning. This course is a must for professionals looking to enhance their skills in TDD and deliver high-quality, reliable software.

Objectives

- Understand the principles and benefits of Test-Driven Development (TDD) and its application in Python.
- Learn about different types of testing including unit tests, integration tests, and end-to-end testing.
- Familiarize with the components of testing such as tests, test suites, assertions, setup/teardown, and mocking.
- Gain hands-on experience with Python unit testing frameworks like Unittest and PyTest.
- Master the art of creating test cases, parametrized tests, and setting up and tearing down tests.
- Understand the concept of assertions and learn to create custom assertions and assert exceptions.
- Learn how to organize tests for easy discovery and efficient execution.
- Understand the importance of code coverage and learn how to measure it using various tools.

Prerequisites

Python programming experience is required. Experience with unit testing in other programming languages is beneficial but not mandatory.



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Student will need the latest stable release of Python, the ability to install Python packages, a modern web browser, Visual Studio Code, and the ability to install Visual Studio Code Extensions.

Outline

- Overview of Unit Testing
 - What is Test-Driven Development (TDD)?
 - Benefits of TDD
 - Challenges of TDD
 - Test-Driven Development with Python
- Principles of TDD (inspired by “Uncle” Bob Martin)
 - Three Laws of TDD
 - Clean Tests
 - One Assert Per Test
 - Five Rules: FIRST
 - Fast
 - Independent
 - Repeatable
 - Self-Validating
 - Timely
 - Red, Green, Refactor Technique
- Kinds of Testing
 - Unit Tests
 - Integration Tests
 - E2E Testing
 - Automated vs. Manual Testing
 - Testing & DevOps
- Testing Parts
 - Tests



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Test Suites
- Assertions
- Setup/Teardown
- Mocks, Fakes, Stubs
- Arrange, Act, Assert
- Test Frameworks
- Test Runners
- Code Coverage
- Python Unit Testing Frameworks
 - Unittest
 - PyTest
- Unit Testing with Unittest
 - Running tests
 - Console
 - Visual Studio Code
 - Creating Test Cases
 - Parametrized Tests
 - Setup and Teardown
- Assertions
 - What is an Assertion?
 - Types of Assertions
 - Unittest Assertions
 - Custom Assertions
 - Assert Exceptions
- Mocking
 - What is Mocking?
 - Why Mock?
 - Mocking with unittest.mock
 - Mocking and Patching
 - Mocking Environment Variables
 - Mocking File I/O
 - Mocking Network Calls
- Organizing Tests
 - Test Discovery
 - Test Suites



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Test Case Classes
- Test Case Methods
- Test Case Inheritance
- Code Coverage
 - What is Code Coverage?
 - Why Code Coverage?
 - Measuring Code Coverage
 - Code Coverage Tools
 - Unittest Coverage
 - Coverage Reports
- CI/CD and Testing
 - Continuous Integration, Delivery, and Deployment
 - Unit Testing Tasks
 - Pull Requests Pipeline
 - Build Pipeline
 - CI/CD Platforms
 - Azure DevOps
 - GitHub Actions
 - GitLab CI
 - Jenkins
 - Code Analysis
 - Code Quality
 - MyPy
 - Linters
 - Static Analysis
- Conclusion
 - Summary of Key Concepts
 - Q&A
 - Further Resources and Next Steps