CLOUD CONTRAPTIONS
COMPUTER PROGRAMMING & TRAINING

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

Python Unit Testing with PyTest

Duration

2 days

Description

This comprehensive course on Test-Driven Development (TDD) with Python is designed for programming professionals seeking to enhance their skills in software testing. The course covers a wide range of topics, from the fundamentals of TDD and its principles to different types of testing and the components involved in testing. Participants will gain hands-on experience with Python unit testing frameworks, particularly Unittest and PyTest, and learn about assertions, fixtures, mocking, and code coverage. The course also delves into the role of testing in Continuous Integration/Continuous Deployment (CI/CD) and provides an overview of various CI/CD platforms. With a blend of theoretical knowledge and practical exercises, this course aims to equip participants with the skills to write clean, efficient, and reliable code.

Objectives

- Understand the concept of Test-Driven Development (TDD) and its benefits and challenges.
- Learn the principles of TDD, including the Three Laws, Clean Tests, One Assert Per Test, and the FIRST rules.
- Distinguish between different types of testing, including unit tests, integration tests, and E2E testing.
- Understand the components of testing, including tests, test suites, assertions, setup/teardown, and test frameworks.
- Learn how to use Python unit testing frameworks, particularly Unittest and PyTest.
- Understand the concept of assertions and how to create custom assertions and assert exceptions.
- Learn about fixtures, mocking, and code coverage in testing.
- Understand the role of testing in CI/CD and learn how to use various CI/CD platforms.

Prerequisites

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

CLOUD CONTRAPTIONS
COMPUTER PROGRAMMING & TRAINING

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

  - 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 PyTest
  - Install PyTest
  - Running tests
    - Console
    - Visual Studio Code
  - Creating Test Cases
  - Parametrized Tests
- Assertions
  - What is an Assertion?
  - Types of Assertions
  - PyTest Assertions
  - Custom Assertions
  - Assert Exceptions
- Fixtures
  - What is a Fixture?
  - PyTest Fixtures
  - Scope of Fixtures
  - Fixture Finalization
  - Fixture Factories
  - Parametrized Fixtures
  - Setup and Teardown
- Mocking
  - What is Mocking?
  - Why Mock?

  - Mocking with PyTest-Mock
  - Mocking with unittest.mock
  - Mocking Environment Variables
  - Mocking File I/O
  - Mocking Network Calls
- Code Coverage
  - What is Code Coverage?
  - Why Code Coverage?
  - Measuring Code Coverage
  - Code Coverage Tools
  - PyTest Coverage
  - Coverage Reports
- CI/CD and Testing
  - Continuous Integration, Delivery, and Deployment
  - Unit Testing Tasks
    - Pull Requests Pipeline
    - Build Pipeline
  - CI/CD Platforms
    - Azue 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