

Certified Associate Tester with Python (PCAT)

Duration: 4 Days

Course Description:

The Certified Associate Tester with Python (PCAT) course delves into fundamental software testing concepts, test automation, and advanced unit testing techniques using Python. The course is essential for anyone looking to solidify their software testing skills with Python.

Target Audience:

- Software Testers
- Quality Assurance Engineers
- Test Automation Engineers
- Software Developers
- Python Developers
- IT Professionals transitioning to software testing
- Quality Analysts
- DevOps Engineers
- Technical Leads
- Software Development Engineers in Test (SDET)
- Computer Science Graduates
- Software Quality Consultants
- IT Project Managers
- Professional testers seeking Python proficiency

Prerequisites:

To successfully undertake training in the Certified Associate Tester with Python (PCAT) course, students are recommended to have the following minimum required knowledge and skills:

- Basic Python Programming Skills: Equivalent to the knowledge gained from PCEP (Python Certified Entry-Level Programmer) and PCAP (Python Certified Associate Programmer) courses.
- Understanding of Software Development Concepts: Familiarity with basic software development and testing concepts.
- Foundational Knowledge of Algorithms and Data Structures: Basic understanding of standard algorithms and data structures to support test case writing and automation.

These prerequisites are essential to ensure that students can fully grasp and build upon the concepts covered in the course modules.

Course Objectives:

- Understand essential software testing concepts and terminology.
- Identify different levels and principles of software testing.
- Learn the initiation and completion phases of the testing process.
- Explore the significance of test automation and code refactoring.
- Utilize assertion statements, context managers, and decorators in Python.
- Gain a comprehensive understanding of unit test anatomy and xUnit architecture.
- Implement good practices and specialized assertions in unit testing.
- Master advanced unit testing techniques, including test parameterization and mocking.
- Apply principles of Test-Driven Development (TDD) through the Red-Green-Refactor cycle.
- Comprehend the fundamentals of Behavior-Driven Development (BDD).

Course Outlines:

Module 1: Software Testing Essentials

- Software Testing Concepts and Terminology
- Levels of Testing
- Software Testing Definitions
- Testing Principles
- Start and End of the Testing Process
- The Test Pyramid
- Code Coverage

Module 2: Test Automation and Code Refactoring

- The Goal and Importance of Test Automation
- The Code Refactoring Loop
- Code Refactoring
- Module 3: Assertions, Context Managers, Decorators and Methods
- The Assertion Statement
- Context Managers
- Function Decorators
- Class Decorators
- Different Types of Python Methods

Module 4: Foundations of Unit Testing

- Anatomy of a Unit Test
- xUnit Architecture
- Running Tests
- Good Practices
- Specialized Assertions
- Unit Tests as a source Documentation
- AAA Pattern used in Practice
- Using the Unit test Module

Module 5: Advanced Unit Testing Techniques

- Unit Test Preparation
- Unit Test Parameterization
- Marking Tests for Special Conditions
- Selective Test Execution
- Using Test Doubles
- Mocking with Patch
- Handling Exception in Unit Tests

Module 6: Test-Driven and Behaviour-Driven Development

- Test Driven Development Fundamentals
- Implementing the Red Green Refactor Phases
- Behaviour Driven Development Fundamentals

REGISTER NOW!

training@trends.com.ph
 (+632) 8863-2123
 www.trendsacademy.com.ph