Certified Associate Tester with Python (PCAT™)

Kickstart your career in software testing with this PCAT Python Associate Tester training. Learn how to write reliable, efficient tests using unittest, PyTest, and test-driven development (TDD). Master the fundamentals of Python test automation and quality assurance to ensure software runs as expected. Perfect for aspiring software testers, QA engineers, and developers looking to build a strong foundation in Python testing.

CBT Nuggets course material →

STUDY PLAN

WEEK 1

Introduction to Software Testing

The Four Traits of Effective Testing

Errors vs. Defects vs. Bugs vs. Failures

Challenge & Solution

Understand the Different Levels of Testing

Unit Testing

Integration Testing

System Testing and End-to-End Testing

Challenge & Solution

The Seven Principles of Software Testing

The Seven Principles

Testing Shows the Presence of Defects

Exhaustive Testing is Impossible

Early Testing

Defect Clustering

The Pesticide Paradox

The Absence-of-Errors Fallacy

Challenge & Solution

Basics of the PyTest Library

Getting Started with PyTest

PyTest Assertions vs. Unittest Assertions

PyTest Fixtures

WEEK 2

Challenge & Solution

Unittest Assertions In-Depth

The Importance of Custom Assertions

Creating Custom Assertions

Numeric Assertions

Asserting Types

Challenge & Solution

Python Class Decorators for Testing

Basics of Class Decorators

Class Decorator Use Cases

Using Class Decorators in Tests

Challenge & Solution

Basics of Assertions

What Are Assertions?

Basic Assertion Syntax

Different Types of Assertions

Challenge & Solution

Python Context Managers for Testing

Basics of Context Managers

WEEK 3

Creating Custom Context Managers

Creating Custom Context Managers

Using Context Managers for Testing

Challenge & Solution

Python Function Decorators for Testing

Basics of First-Class Functions

Creating Custom Decorators

Unittest's Testing Decorators

Challenge & Solution

xUnit Architecture and the Unittest Module

Introduction to xUnit Architecture

xUnit Test Cases

Running xUnit Tests and Making Better Assertions

xUnit Test Suites and Runners

Challenge & Solution

A Tour Of Unittest Assertions

The Main Assertion Types

Basic Assertions

Container Assertions

Exception Assertions

Challenge & Solution

WEEK 4

Working with Test Fixtures in unittest

The Importance of Test Fixtures

The setUp and tearDown Methods

Using Fixtures for File Operations

Challenge & Solution

Parameterized Testing Guide

What is Parameterized Testing?

Manual Parameterization

Using the Parameterized Library

Challenge & Solution

Advanced Test Parameterization

lParameterization in PyTest

Parameterizing Test Fixtures

Cross-Parameterization

Challenge & Solution

Basics of Test Doubles

The 5 Main Types of Test Doubles

Using Dummies in Tests

WEEK 5

Using Stubs in Tests

Using Spies in Tests

Challenge & Solution

Test Doubles In-Depth

Using Mocks in Tests

Using Fakes in Tests

Mock Assertions

Challenge & Solution

Advanced Mocking Techniques

Using patch for Mock Dependencies

Mocking Classes vs. Functions vs. Modules vs. Objects

Mocking Builtins, Files, and Time

Challenge & Solution

Writing Unit Tests with F.I.R.S.T. Principles

Introduction to F.I.R.S.T. Principles

Making Tests Fast

Making Tests Isolated/Independent

Making Tests Repeatable

Making Tests Self-Validating

Making Tests Timely

Challenge & Solution

Strategies for Testing Python Methods

The Different Method Types

Testing Instance Methods

Testing Class Methods

Testing Static Methods

Challenge & Solution

Monkeypatching in Pytest

The Basics of Monkeypatching

Patching Functions and Methods

WEEK 6

Patching Attributes

Patching Environment Variables

Patching User Input

Challenge & Solution

Basics of Test-Driven Development (TDD)

What is Test-Driven Development?

The Red-Green-Refactor Cycle

A Detailed Walkthrough

Challenge & Solution

Basics of Behavior-Driven Development (BDD)

What is Behavior-Driven Development?

Gherkin Syntax

Creating .feature Files

Writing and Linking Tests

Challenge & Solution

Intermediate Behavior-Driven Development Concepts

Using Parameterized Steps

Using Scenario Outlines

Providing Background in BDD

Using "And" and "But" in Steps

Challenge & Solution

Advanced Behavior-Driven Development

User Stories in BDD

The "Behave" Library as an Alternative to Pytest

WEEK 7

Writing, Running, and Troubleshooting Behave Tests

Challenge & Solution

Python Testing Best Practices

Five Signs You're Not Following Best Practices

Best Practices for File Structure

Best Practices for Naming Files & Tests

Unit Tests as Documentation

Challenge & Solution

Basics of Property-Based Testing

What is Property-Based Testing?

The Hypothesis Library

Working with Different Hypothesis Strategies

Challenge & Solution

Basics of Testing Interfaces

What is Interface Testing?

Basic Interface Info and Controls

Recording More Complicated Actions

Challenge & Solution

WEEK 8

Advanced Interface Testing Concepts

Finding and Interacting with Elements Dynamically

Integrating Interface Testing with PyTest

Testing and Using Hotkeys

Challenge & Solution

Testing Web Interfaces

What is Web Interface Testing?

Getting Selenium Set Up

Finding and Interacting with Elements

Making Assertions About Webpages

Challenge & Solution

Advanced Web Interface Testing

How to Wait For Things

Clicking On Elements

A Brief Review of CSS Selectors

Challenge & Solution

Python Code Refactoring Principles

Why is Refactoring So Important?

Five Signs You Need to Refactor

Refactoring Duplicated Code

Refactoring Long Functions, Large Classes, and Deep Nesting

WEEK 9

Refactoring Tightly Coupled Modules

Challenge & Solution

Last updated 6.25.2025