ONE PAGE SUMMARY

IT & Software › Software Testing › Unit Testing

Introduction to Unit Testing

Master the Basics of Unit Testing for Reliable Software Development

DEFINITION

> Unit testing is a software testing technique where individual components are tested in isolation. It ensures that each part of the code works correctly before integrating it into the larger system.

KEY CONCEPTS

Test Case

A test case is a specific condition or set of conditions under which a unit is tested.

Mock Objects

Mock objects simulate the behavior of real objects to test interactions without relying on external systems.

Code Coverage

Code coverage measures how much of the code is executed during tests, indicating test effectiveness.

Assertions

Assertions are statements that verify whether a condition is true, helping to validate the expected outcome.

Test-Driven Development

Test-driven development (TDD) is a practice where tests are written before the code itself, guiding development.

EXAMPLES

→ Testing a function that calculates the sum of two numbers.

→ Verifying that a login function returns the correct user object.

→ Checking if an input validation function correctly identifies invalid email formats.

MEMORY TIPS

★ Remember 'TDD' as 'Test First, Code Second' for Test-Driven Development.

★ Use 'MCA' for Mock, Code, Assert to recall the testing process.

★ Think of 'Coverage' as a blanket that covers your code to keep it safe from bugs.

COMMON MISTAKES

✗ Neglecting to test edge cases or unusual inputs.

✗ Writing tests after the code is already developed, losing the benefits of TDD.

✗ Overlooking the importance of maintaining and updating tests as code evolves.

QUICK RECAP

Unit testing focuses on verifying individual components of software for correctness. It involves creating test cases, using mock objects, and ensuring high code coverage to catch bugs early in the development process.