ONE PAGE SUMMARY

IT & Software › Software Testing › Unit Testing

Introduction to Unit Testing

Mastering Unit Testing: The Foundation of Reliable Software

DEFINITION

> Unit testing is a software testing method where individual components are tested in isolation. It ensures that each part of the code functions correctly before integration.

KEY CONCEPTS

Test Case

A test case is a specific scenario used to validate that a unit of code behaves as expected.

Mocking

Mocking involves creating simulated objects to mimic the behavior of real components in tests.

Code Coverage

Code coverage measures the percentage of code that is tested by unit tests, indicating test effectiveness.

Assertions

Assertions are statements in tests that verify whether a condition is true, helping to confirm expected outcomes.

Test-Driven Development

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

EXAMPLES

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

→ Verifying that a user login function returns the correct error message for invalid credentials.

→ Ensuring a sorting algorithm correctly sorts an array of integers.

MEMORY TIPS

★ Remember 'T' in TDD stands for 'Test' to emphasize writing tests first.

★ Use the acronym 'MATS' for key concepts: Mocking, Assertions, Test cases, and Software coverage.

★ Think of unit tests as safety nets that catch bugs before they fall into production.

COMMON MISTAKES

✗ Neglecting to test edge cases, leading to unhandled scenarios.

✗ Writing tests that are too complex, making them hard to maintain.

✗ Failing to run tests regularly, resulting in undetected issues.

QUICK RECAP

Unit testing is essential for ensuring code quality by testing individual components. Key concepts include test cases, mocking, and assertions. Regular testing prevents bugs and improves software reliability.