What is Junit?

JUnit is an open source Unit Testing Framework for JAVA. It is useful for Java Developers to write and run repeatable tests. Erich Gamma and Kent Beck initially develop it. It is an instance of xUnit architecture. As the name implies, it is used for Unit Testing of a small chunk of code.

Developers who are following test-driven methodology must write and execute unit test first before any code.

Once you are done with code, you should execute all tests, and it should pass. Every time any code is added, you need to re-execute all test cases and makes sure nothing is broken.

What is Unit Testing?

Before discussing Junit testing in detail, it is imperative to understand what is Unit Testing?

Unit Testing is used to verify a small chunk of code by creating a path, function or a method. The term "unit" exist earlier than the object-oriented era. It is basically a natural abstraction of an object oriented system i.e. a Java class or object (its instantiated form).

Unit Testing is used to identify defects early in software development cycle.

Unit Testing will compel to read our own code. i.e. a developer starts spending more time in reading than writing.

Defects in the design of code affect the development system. A successful code breeds the confidence of developer.

Why you need JUnit testing

It finds bugs early in the code, which makes our code more reliable.

JUnit is useful for developers, who work in a test-driven environment.

Unit testing forces a developer to read code more than writing.

You develop more readable, reliable and bug-free code which builds confidence during development.

Annotations are discussed below-

Setup

@Before annotation is used on a method containing Java code to run before each test case. i.e it runs before each test execution.

Teardown (regardless of the verdict)

@After annotation is used on a method containing java code to run after each test case. These methods will run even if any exceptions are thrown in the test case or in the case of assertion failures.

Once-only setup

It is possible to run a method only once for the entire test class before any of the tests are executed, and prior to any **@Before** method(s).

"Once only setup" are useful for starting servers, opening communications, etc. It's time-consuming to close and re-open resources for each test.

This can be done using the **@BeforeClass** annotation

Once-only tear down

Similar to once only setup , a once-only cleanup method is also available. It runs after all test case methods and **@After** annotations have been executed.

It is useful for stopping servers, closing communication links, etc.

This can be done using the **@AfterClass** annotation

JUnit is a framework which supports several annotations to identify a method which contains a test.

JUnit provides an annotation called **@Test,** which tells the JUnit that the public void method in which it is used can run as a test case.

A test fixture is a context where a test case runs

To execute multiple tests in a specified order, it can be done by combining all the tests in one place. This place is called as the test suites.

JUnit provides a tool for execution of the tests where we can run our test cases referred as Test Runner.