In Java, classes and methods play a vital role, and therefore we need to test the classes and methods, which together known as a Unit

So, for performing unit testing in Java, there are several testing frameworks. One such famous framework is JUnit which is a framework for performing unit testing.

\* when doing a big project, it may consist of numerous classes with numerous methods. To check and test all those manually might not be flexible, and change in one class will definitely affect the other classes. Thus, there comes the role of automation testing, i.e., Junit testing.

## **What are Test Cases**

Test Cases are the set of conditions that are tested by a software tester for the developed application. The tester checks and evaluates if the built software is fulfilling all the requirements of the customer. If any bug or error is found, the tester informs the development team. The need to build test cases and perform testing is to verify that all the customer's needs are fulfilled, and no bugs are present in the software.

## **JUnit**

JUnit is a unit testing framework for java.

It is used by java developers to write and execute tests.

Every time a new code is added, all the tests cause java to be re-executed.

Unit testing refers to the testing of small chunks of codes that are logically isolated.

It helps in early detection of defects.

The Developer tends to spend more time on reading the code.

Successful cond increase the confidence of the developers.

**@Test** annotation specifies that method is the test method.

**@Test(timeout=1000)** annotation specifies that method will be failed if it takes longer than 1000 milliseconds (1 second).

**@BeforeClass** annotation specifies that method will be invoked only once, before starting all the tests.

**@Before** annotation specifies that method will be invoked before each test.

**@After** annotation specifies that method will be invoked after each test.

**@AfterClass** annotation specifies that method will be invoked only once, after finishing all the tests.

Assert Class

The org.junit.Assert class provides methods to assert the program logic.

1. **void assertEquals(boolean expected,boolean actual)**: checks that two primitives/objects are equal. It is overloaded.
2. **void assertTrue(boolean condition)**: checks that a condition is true.
3. **void assertFalse(boolean condition)**: checks that a condition is false.
4. **void assertNull(Object obj)**: checks that object is null.
5. **void assertNotNull(Object obj)**: checks that object is not null.