**Unit Testing & JUnit**

1. What is unit testing?

Answer:

Unit testing is a level of testing done by the developers on the unit of work which they have created. Unit is a small part of a software which have a functionality. This is to done to validate code is working fine as a single unit. Unit testing will be done at development phase.

1. What is the difference between manual testing and automated testing?

Answer:

|  |  |
| --- | --- |
| Manual Testing | Automated testing |
| Test cases are executed manually by humans without any tools | Test cases are executed with the help of tools and software |
| Less reliable due human error | More reliable |
| Time consuming | Testing will be fast with the help of tools. |
| Lot of human investment is required | Investment of testing tools is required |
| Suitable for Exploratory, Usability and Adhoc testing | Suitable for Regression testing, Performance testing, Load testing |
| Less expensive | Can be expensive depends on the tools used. |
| Testing is practical when test cases are run once or twice | Used when test cases are required to run repeatedly |

1. Is it necessary to write the test case for every logic? If yes, why

Answer:

Yes, it is necessary to write test cases for every logic. The main objective of unit testing is to make it sure that all the business logic is working as expected in a software. So it is necessary to test each logic in a program

1. What are the features of JUnit?

Answer:

* Junit is an open source unit testing tool and it is used to test unit of code in Java.
* Junit tests can run automatically, check their own results and provide immediate feedback.
* Test cases can be organised into test suites and it can be run together
* Provide test runner for running tests.
* Provide annotations to identify test methods.
* Allows to write test cases faster and with increased quality
* Junit is less complex and takes less time

1. What are the important JUnit annotations? And its usage in coding

Answer:

Annotation are added in Junit to identify the test methods.

Important Junit annotations are:

@Test: It tells the Junit that the method to which it is attached can be run as a test case

@Before: This annotation is used if we need to execute some statement before each test

@BeforeClass: This annotation is used when we need to execute some statements before all the test cases

@After: It is used when we need to execute some statement after each test case

@AfterClass: It is used when we need to execute some statement after all test cases

@Ignore: This annotation is used when we need to ignore some statements during the execution of a test

@Mock: used to mock object

1. What does Assert class?

Answer:

Assert is useful in determining the status of the test case. It will decide whether the case pass or fail. Assert class has a group of assertion methods useful in writing test cases and to detect test failure.

AssertEqualsI () is a functionality which hold the expected output value and actual output value. If both values are same then test is success otherwise it’s a failure

1. What is Code Coverage?

Answer:

It is the measure of the degree to which source code of a program has been test in automated testing. In other words, it measures the effiency of testing. Different code coverage levels are there

* Statement Coverage: Number of statements that have been successfully validated
* Decision Coverage: Boolean expressions are validated.
* Branch Coverage: In branch coverage every outcome from a code module is tested.
* Toggle Coverage: Decision control structures like loops that have executed successfully.
* FSM Coverage: Finite State Machine coverage works on the behaviour of the design

1. What are the best practices to perform Unit Testing?

Answer:

* Test unit should be written during the time of development.
* Unit test must be easy to read and understand, not only for whoever wrote it but for others as well.
* Unit test should be isolated with no dependencies on environmental factors
* Should be automated
* Each test should validate only one thing. Then test case will be more simple and understandable.
* Unit test should be trustable, which means test should fail if the code is broken

1. What is Mocking?

Answer:

Mocking is a unit testing technique in which a real code is replaced by a mock code or dummy code. It is used to avoid external dependencies but still developers will be able to test it.