

An average recruiter has 6 seconds to check your resume... will you be noticed?

[Check Premium Resumes](#)



Top 11 JUnit interview questions

1) Explain what is JUnit?

JUnit is a testing framework for unit testing. It uses Java as a programming platform, and it is an Open Source Software managed by the JUnit.org community.

2) Explain what is Unit Test Case?

Unit Test Case is a part of the code that ensures that the another part of code (method) behaves as expected. For each requirement, there must be at least two test cases one negative test and one positive test.

3) Explain how you can write a simple JUnit test case?

- Determine a subclass of TestCase
- To initialize object(s) under test, override the setup() method
- To release object(s) under test override the teardown() method

Determine one or more public test XYZ() methods that exercise the objects under test and assert expected results.

4) Mention what are parameterized tests?

Parameterized tests enable developer to perform the same test over and over again using different values.

5) Mention what is the difference between JUnit and TestNG?

JUnit

- In JUnit the naming convention for annotation is a bit complicated for, e.g., “Before”, “After” and “Expected”
- In JUnit, for a method declaration you have to follow a specific style like using “@BeforeClass” and

TestNG

- In TestNG it is easier to understand annotations like “BeforeMethod”, “AfterMethod” and “ExpectedException”
- In TestNG, there is no restriction like you have to declare methods in a specific format

- “@AfterClass”.
- In JUnit method name constraint is present
- JUnit framework does not have “Parameterized Test” or “Dependency Test” feature
- In JUnit grouping of test cases are not available
- JUnit does not support parallel execution on Selenium test cases
- It cannot re-run the failed cases
- In TestNG method name constraint is not present, and you can determine any test method names
- TestNG use “dependOnMethods” to implement the dependency testing
- In TestNG, grouping of test cases is available
- In TestNG Parallel execution of Selenium test cases are possible
- It can rerun the failed tests



6) Mention different methods of exception handling in JUnit?

There are different methods of exception handling in JUnit

- Try catch idiom
- With JUnit rule
- With @Test annotation
- With catch exception library
- With customs annotation

7) Explain what is ignore test in JUnit?

When your code is not ready, and it would fail if executed then you can use **@Ignore** annotation.

- It will not execute a test method annotated with **@Ignore**
- It will not execute any of the test methods of test class if it is annotated with **@Ignore**

8) List out some useful JUnit extensions?

JUnit extensions include

- Cactus
- JWebUnit
- XMLUnit
- MockObject

9) Explain who should use JUnit - a developer or tester? Why you use JUnit to test your code?

JUnit is more often used by developers to implement unit tests in JAVA. It is designed for unit testing that is more a coding process and not a testing process. However, many testers and QA engineers use JUnit for unit testing.

JUnit is used because

- It test early and does automate testing
- JUnit tests can be compiled with the build so that at unit level, regression testing can be done
- It allows test code re-usage
- JUnit tests behave as a document for the unit tests when there is a transfer

10) Explain what is JUnitCore Class?

JUnitCore class is an inbuilt class in JUnit package; it is based on Façade design pattern, this class is used to run only definite test classes only.

11) Explain how you can run JUnit from the command window?

To run JUnit from the command window, you have to follow the steps

- Set the CLASSPATH
- Invoke the runner:

Java org.junit.runner.JUnitCore

[Guru99](#) provides [FREE ONLINE TUTORIAL](#) on Various courses like

[PHP](#)

[Java](#)

[Linux](#)

[Apache](#)

[Perl](#)

[SQL](#)

[VB Script](#)

[JavaScript](#)

[Accounting](#)[Ethical Hacking](#)[Cloud Computing](#)[Jmeter](#)[Manual Testing](#)[QTP](#)[Selenium](#)[Test Management](#)[Load Runner](#)[Quality Center](#)[Mobile Testing](#)[Live Selenium
Project](#)[Enterprise Testing](#)[Live Testing Project](#)[Sap & All Modules](#)

Copyrighted Material