1. **Why we do parallel testing in TestNg?**

**A.**   In every project it is required to perform multi-browser testing to make sure that the functionality is working as expected with every browser to give equal user experience to all of the wide range of audience. It takes a considerable time to test everything on every browser and when we have used automation to reduce the testing efforts then why don’t we perform the multi-browser testing using automation. TestNG gives us functionality to perform same test on different browsers in a simple and easy way.

STEPS:

1)Create your Script to test a LogIn application using TestNG class.

2) Pass ‘Browser Type’ as parameters using TestNG annotations to the before method of the TestNG class. This method will launch only the browser, which will be provided as parameter.

3) Create a TestNG XML for running your test. Configure the TestNG XML for passing parameters i.e. to tell which browser should be used for Running the Test.

4) Now it’s time to run the xml. Run the test by right click on the testng.xml file and select Run As > TestNG Suite.

**2. Difference between TestNG and juint?(gomathy-wipro/visa)(puja Jha-F2F-Apple)**

1. In TestNG, Parameterized test configuration is very easy while It is very hard to configure Parameterized test in JUnit.
2. TestNG support group test but it is not supported in JUnit.
3. TestNG has a feature to configure dependency test. Dependency test configuration for software web application is not possible in JUnit.
4. TestNG support @BeforeTest, @AfterTest, @BeforeSuite, @AfterSuite, @BeforeGroups, @AfterGroups which are not supported in JUnit.
5. Test prioritization, Parallel testing is possible in TestNG. It is not supported by JUnit

**3. Have you created custom annotations in TestNG? How did you create?**

Well, you can write your own Annotation for that and have your Annotation processed at runtime like other TestNG Annotations.

Here is how you do it -

* Create a basic maven project and add TestNG as a dependency.
* Create your Annotation
* Create your Listener
* Create a folder called META-INF\services in src\main\resources
* Create a file called org.testng.ITestNGListener in src\main\resources\META-INF\services
* Add the following line to this file - com.amareshp.annotations.MyTestNGAnnotationListener -

**4.What are the annotations used in TestNG?**

@Test, @BeforeSuite, @AfterSuite, @BeforeTest, @AfterTest, @BeforeClass, @AfterClass, @BeforeMethod, @AfterMethod,@dataprovider

**5. How do you group the tests in TestNG?**

**Groups are specified in your testng.xml file using the <groups> tag. It can be found either under the <test> or <suite> tag. Groups specified in the <suite> tag apply to all the <test> tags underneath.**

**STEPS:-------**

* **Create a java test class, say, GroupTestExample.java.**
* **Add test methods, testPrintMessage() and testSalutationMessage(), to your test class.**

**Group the test method in two categories −**

* **Check-in tests (checkintest) − These tests should be run before you submit new code. They should typically be fast and just make sure no basic functionality is broken.**
* **Functional tests (functest) − These tests should cover all the functionalities of your software and be run at least once a day, although ideally you would want to run them continuously.**

**Create testng.xml in C:\> TestNG\_WORKSPACE, to execute test case(s). Here, we would be executing only those tests, that belong to the group *functest*.**

**6. What are the benefits of using TestNG ?**

**Ans-**

a) TestNG allows us to execute of test cases based on group.

b) In TestNG Annotations are easy to understand.

c) Parallel execution of Selenium test cases is possible in TestNG.

d) Three kinds of report generated

e) Order of execution can be changed

f) Failed test cases can be executed

g) Without having main function we can execute the test method.

h) An xml file can be generated to execute the entire test suite. In that xml file we can   rearrange our execution order and we can also skip the execution of particular test case.

**-------------------------------------F2F-Manasa-Apple----------------------------------**

**why testng?**