TestNG important question

1. ***What is TestNG?***
   - TestNG is open a source testing framework were NG for Next Generation
   - TestNG is testing framework designed to simplify a broad range from unit testing to System Testing.
   - initially TestNG developed for unit testing but now we are using for all kinds for testing.
   - TestNG inspired tool from JUnit and NUnit but introducing a new feature that make more powerful tool in market.
   - TestNG annotation are easier to use.

2. ***what are the advantages of TestNG***
   - we can create the test cases.
   - we can prioritize the test cases
   - we can group the test cases, and we can execute or skip the specific group.
   - We can achieve parameterization with the help of @Parameter annotation
   - We can achieve parallel testing and cross browser testing
   - We can achieve data driven testing
   - It generates the HTML reports

3. ***Diff between TestNG and Junit testing framework?***
   - With the help of TestNG we can achieve parallel test execution but with the help of unit it is not possible.
   - By using we can run the test cases as per the own order but by using Junit is not possible.
   - TestNG create automatically HTML Report but in Junit we have add external dependency.
   - By using TestNG we can group the test cases but the by using Junit it is not possible.
   - TestNG support listeners using annotation but Junit is support the listeners using Launcher API, listeners can not support using annotations.

4. ***What are the different annotation present inside the TestNG?***
   - There are 3 types of TestNG annotation 1st is precondition annotation, 2nd @test annotation and 3rd post condition annotation. Precondition annotation are execute before each and every test case, test annotation are used to create the test cases and

post condition annotations are used to execute after the test cases, there are 4 different types of precondition annotation present inside the TestNG, 1st as before suite it will execute before the test cases, 2nd before test it will execute before the classes tags in xml file, 3rd before classes it will execute before all the test cases in class, and 4th before method it will execute before each and every test cases, then test annotation is used to create the test cases and post annotation contains again 5 types 1st after method it will execute after each and every test cases ,2nd after class it will execute after all test cases in class, 3rd before test it will execute after classes tags in xml file and 4th after suite it will execute after the suite.

5. ***TestNG annotation execution sequence?***
   - *@BeforeSuite*
   - *@BeforeTest*
   - *@BeforeClass*
   - *@BeforeMethod*
   - *@Test*
   - *@AfterMethod*
   - *@Afterclass*
   - *@AfterTest*
   - *@AfterSuite*

6. ***What is default priority for test cases?***
   - If we declare any test cases with the help of @Test annotation then default priority is zero.

7. ***Can we mention negative priority for test cases?***
   - Yes we can mention positive as well negative priority for test cases, so first it will execute the negative priority test cases then positive test cases.

8. ***Can we mention same priority for multiple test cases?***
   - Yes we can mention same priority for multiple then it will execute test cases as per the alphabetical order.

9. ***If we don't declare the priority for test cases what will happen?***
   - If we don't declare the priority for test cases then it will execute test cases as per the alphabetical order.

10. Diff between before method and before class?

- BeforeMethod is precondition for each and every test cases in class and before class I precondition for all the test cases in class, just imagine inside the class we declare the before method , before class pre condition and 100 test cases, so before method it will execute 100 times, and before class it will execute only one time.

11. Diff between before class and before test?

- Before Class is precondition for all the test cases in class and before test is precondition for classes tags in xml file, just imagine we have 2 classes Admin page class and contact page class and inside the Admin page class we declare the before class and before test pre condition and there we create xml file for both class, and if we execute the xml file so 1st it will execute the before test then before class as pre condition for only admin page class then it will execute all the test cases from admin page class and then it will execute all the test cases from contact page class.

12. Diff between hard assert and soft assert?

- Hard assert is also known as Assert and soft assert is also known as verify.
- Hard assert is throws exception immediately after assert statement fail and soft assert it collects the error if assert statement fail.
- Hard assert does not execute remaining life of code after hard assert statement fail and soft assert it executes remaining life of code after assert fail.
- If u want throws exception in soft assert, then we use assert all method.
- In hard assert all methods are static, and inside the soft assert all methods are non-static.
- We call hard assert all with the help of Assert class name and we call soft assert all methods with the help of soft assert object name.
- In hard and soft assert can compare string value, boolean ,int , float , double , all datatypes.

13. Diff between assert and verify?

- Assert is also known as hard Assert and verify is also known as soft assert.
- Hard assert is throws exception immediately after assert statement fail and soft assert it collects the error if assert statement fail.

- Hard assert does not execute remaining life of code after hard assert statement fail and soft assert it executes remaining life of code after assert fail.
- If u want throws exception in soft assert, then we use assert all method.
- In hard assert all methods are static, and inside the soft assert all methods are non-static.
- We call hard assert all with the help of Assert class name and we call soft assert all methods with the help of soft assert object name.
- In hard and soft assert can compare string value, boolean ,int , float , double , all datatypes.

14. What are the different tags present inside the XML File?

- XML means Extensive markup language file, inside this file we declare the different tags, 1st tags should be suite tag and inside suite tag mention suite name, then we mention parallel attribute, thread count attribute at the time achieving parallel testing we use this above attribute then we mention listeners tags and inside the listeners tag we mention listener tag, there we pass the implemented listener package name and class name, then parameter tags and inside the tag we pass parameter name and value attribute and then test tags inside the tag we have groups tags, then run tag and then we can declare include or execute tag to run the specific group, then we close include tags, run tag and groups tag and then we have classes tags and inside the classes tag we can declare multiple class tag here we mention different class name with the help of package name then we have to close the classes , test and suite tags.

15. How to achieve parallel testing or cross browser testing?

- We achieve the parallel testing with the help of @ parameters annotation in TestNG, we use these parameter annotation in test level and we pass the parameter value from xml file, inside the xml file there we use parameter tag in test level and there we pass same key name for all the parameter tags in test level and we use different browser value as like chrome ,edge and Firefox and in test level there we use parallel attribute and we pass parallel attribute value as tests keyword and there we use one more attribute as thread-count and we pass thread value and we have to use same parameter name inside the parameters annotation in test level and inside the that method we use same method arguments , and inside these method there we declare the condition to check the browser passing value from xml is same or not if it is same then execute the block of code, and we run these code from XML file.

16. How to skip specific test cases in TestNG?

- There 3 ways we can skip the specific test cases in TestNG, 1st way we can use enabled equal to false attribute in test annotation, 2nd way we can use @ignore annotation in test level and 3rd way we can use groups attribute for test cases and use group name for test cases and then we create xml file and inside the xml file we use exclude tags and inside the exclude tag we can mention group name.

17. How to ignore specific test cases in TestNG?

- There 3 ways we can skip the specific test cases in TestNG, 1st way we can use enabled equal to false attribute in test annotation, 2nd way we can use @ignore annotation in test level and 3rd way we can use groups attribute for test cases and use group name for test cases and then we create xml file and inside the xml file we use exclude tags and inside the exclude tag we can mention group name.

18. How to execute 10 test cases out of 100 test cases?

- If u want execute 10 test cases out of 100 test cases we use groups attribute and we declare group for all the 10 test cases, and then we can run this 10 test cases by using 2 ways, right click on class, then go to the run as option , then run configuration then inside the group radio button we select the mentioned group name checkbox then click on run, 2nd way we can create xml file for same class and inside the xml file we have different tags as like suite tag then test tag and inside the test tag we mention groups tag then run tag and then include tag here we mention group name ,then we close run then close groups tag , then start the classes tags inside the classes tag we have class tag there we mention class name with package name then we close classes tag, test tag and suite tag once we create this file then right click on same xml file and go to run option and click on test suite.

19. How to run specific test cases multiple times

- We execute specific test cases multiple time with the help of invocation count attribute, inside the test annotation we use invocation-count attribute then we pass int number.

20. How to provide time out for test cases

- We provide test cases time out with the help of timeout attribute inside the test annotation we use timeout attribute by passing time in milliseconds and if this test cases execute before time, then it will mark test cases is pass and if test cases takes more time then it will mark test cases is fail.

21. Diff between invocation count and thread Pool Size?

- Invocation count is used to run the same test cases multiple times and thread pool size it allocates multiple thread for run the invocation count test cases parallelly.

22. What is thread- count in TestNG?

- We use Thread count attribute in XML file, we can declare thread-count in suite level or test level. Thread count attribute we pass int type value, just imagine if we pass thread count as 5 then it will allocate 5 instance to run the test cases simultaneously or in parallel

23. How to achieve data driven testing in TestNG?

- Data driven framework it means we separate the test data in excel sheet or database and test scripts in different class. In my current framework we store data in excel sheet and if you want to work with excel sheet then we need Apache POI Dependency and we have to add these dependency inside the in pom dot xml file and we use @DataProvider annotation from TestNG to achieve the data drive testing.
- Basically, in my current framework inside the Utility Layer package there we have created reusable Excel Reader class for read excel sheet file and inside this class we have created 1 constructor with string argument and inside this constructor we have written code for check the if file is present or not, read the file and load the whole workbook code. We are checking file is present or not with the help of File class then we read the File with the help of File Input Steam class and before loading the excel sheet first we have to check excel sheet type, basically there are 2 types of excel sheet present 1st is dot xls excel sheet and dot xlsx excel sheet, if excel sheet type is dot xls then we have to create object of HSSFWorkbook class by passing file input stream instance and if excel sheet type is dot xlsx then we have to create object of XSSFWorkbook class by passing file input stream instance name. and once the workbook is loaded then we focus on specific sheet with the help of get sheet at method by passing sheet index or sheet name, and then we count how many rows

present inside the sheet with the help of get Last Row Num method and also we count how many columns in sheet with the help of get Last Cell Num method.

- Then we capture the sheet data, we have created one more method, Basically inside the sheet, some time we have string value, boolean values, integer value or decimal value, null value so first we have verify which type value present inside the each and every sheet cell, so first we focus on specific sheet with the help of XSSF Sheet object name, with the of XSSF sheet object name we have to call get row and get cell method by passing row number and column int value, once we used this method it will return the XSSFCell object, once we get the Cell object then we use get Cell Type method and if cell type value is String then we capture these value by using get String cell value method, and if cell type value is int then we capture these value by using get raw value method and if cell type value is boolean then we capture these value by using get boolean cell value method and if cell type value is some formula then we capture these value by using get cell formula method and then we store these value in Object class, because object class is super parent class of all the classes in java.it accept the all the datatype values and we return all values one by one.
- Again, we have created one method to read the data from excel sheet and store all data in object in 2-dimensional array and we return this 2-dimensional array.
- And as per the requirement wise we are calling these class in test layer package to achieve the data driven framework and at the time of achieving data driven framework we use @ data provider annotation.