1. What is testing and Features of testing ?
2. What is annotation ?
3. What is @test annotation ?
4. If multiple @test annotations are present in testNg class then execution will happen in which order ?
5. Explain the terminologies we use for testing?
6. How to create maven project file in eclipse ?
7. What is group id and artifact id in maven project ?
8. What is pom in maven project ?
9. Advantages of maven project ?
10. Why we go for batch , group and parallel execution ?
11. What is chronological order for configuration annotation and Explain the order ?
12. How to create Suite file in testing and which format ?
13. How to make dependency between the test Cases ?
14. Explain all the helper methods in testNg?
15. How to generate report in testNg and types of report?
16. What is data provider annotation , use and Syntax to create Data provider ?
17. What is listeners Annotation ?
18. What is ItestListeners and IRetryAnalyser ?
19. How many ways are there to connect to listeners implemented class ?
20. What is data driven testing and its advantages ?
21. How to read the data from excel file ?
22. How to read data from properties file ?
23. How to read data from xml file ?
24. What is hardcoding in automation ?
25. What is assertion and why we go for assertion ?
26. Different between hard and soft assert ?
27. List out the methods of hard and soft assert ?
28. What is POM in Selenium ?
29. Why we go for Pom ?
30. Advantages of Pom ?

1)What is testng and its features ?

- Full form is test next generation
- It is a unit testing framework tool like Junit and Nunit and it is inspired from Junit
- And it having an advanced features , as well as it is used for unit , functional , integration ,end to end testing etc .
- Its is used by both developers and test engineers
- And it is plug in for eclipse so we need install the plug-in in eclipse

Features

- **Batch Execution**
- **Group Execution**
- **Parallel Execution**
- **Parameterization**
- **Prioritization**
- **Create dependencies between test cases**
- **Skip test cases**
- **Generate report and more**

2. What is annotation?

Annotation is a symbol or templates which is used to give information to the compiler , test engineer and developer etc

Which can be used on top of class, variable , method .

3. What is @test annotation?

@Test Annotation: it is one of the important annotation of testng and it acts like a main method in execution .

```
Syntax : public void anyName(){

             Statement ;

             }
```

4. If multiple @test annotations are present in TestNG class then execution will happen in which order?

Execution Order: By default, TestNG executes test methods in alphabetical order of their names to be exact based on ascii value. However, we can control the order of execution using the `priority` attribute of the `@Test` annotation.

5. Explain the terminologies we use for testing?

Common Testing Terminologies:

- **Testng Class :** A class which is having @test method , that class we call it as a testng class
- **Test Case :** A method which is prefix with @test annotation that is called asTestcase
- **Test Script :** A Steps which is present in @test method , that is called as Test Steps

6. How to create maven project file in Eclipse?

To create a Maven project in Eclipse:

1. Go to **File** > **New** > **Other**.
2. Select **Maven** > **Maven Project** and click **Next**.
3. Choose a workspace location and click **Next**.
4. Select an archetype (e.g., `maven-archetype-quickstart`) and click **Next**.
5. Fill in the **Group Id** and **Artifact Id**.
6. Click **Finish** to create the Maven project.

7. What is group id and artifact id in maven project?

- **Group Id:** It defines the unique identifier of the project group, typically based on the domain name (e.g., `com.example`).
- **Artifact Id:** It defines the unique base name of the primary artifact being generated by the project (e.g., `myapp`).

8. What is POM in maven project?

POM (Project Object Model): It is an XML file (`pom.xml`) that contains information about the project and configuration details used by Maven to build the project. It includes dependencies, plugins, goals, and other build configurations.

9. Advantages of Maven project?

Advantages of Maven:

- **Dependency Management:** Automatically handles project dependencies.
- **Build Automation:** Simplifies the build process using standard conventions.
- **Project Structure:** Enforces a standard project structure.
- **Consistent Builds:** Ensures consistent and reproducible builds.
- **Extensible:** Supports a wide range of plugins for various tasks.
- **Documentation:** Automatically generates project documentation.

10. Why we go for batch, group, and parallel execution?

- **Batch Execution:** Running a set of tests together, useful for nightly or regression testing.
- **Group Execution:** Organizing tests into groups for modular and organized testing.
- **Parallel Execution:** Running tests concurrently to reduce the overall execution time.

11. What is chronological order for configuration annotation and Explain the order?

Configuration Annotations in TestNG:

1. **@BeforeSuite:** Runs before all tests in the suite.
2. **@BeforeTest:** Runs before any test method in the `<test>` tag in the `testng.xml` file.

3. **@BeforeClass:** Runs before the first method in the current class.
4. **@BeforeMethod:** Runs before each test method.
5. **@Test:** Test method.
6. **@AfterMethod:** Runs after each test method.
7. **@AfterClass:** Runs after all the test methods in the current class.
8. **@AfterTest:** Runs after all test methods in the `<test>` tag in `testng.xml`.
9. **@AfterSuite:** Runs after all tests in the suite.

12. How to create Suite file in testing and which format?

Creating a Suite File:

- Create a new XML file (e.g., `testng.xml`).
- Define the suite, tests, classes, and methods within the XML structure.

```
xml
Copy code
<!DOCTYPE suite SYSTEM
"http://testng.org/testng-1.0.dtd" >
<suite name="SuiteName">
    <test name="TestName">
        <classes>
            <class
name="com.example.MyTestClass" />
        </classes>
    </test>
</suite>
```

13. How to make dependency between the test cases?

Test Dependency:

- Use the `dependsOnMethods` attribute in the `@Test` annotation to create dependencies between test methods

```
@Test
public void testMethod1() {
    // test code
}

@Test(dependsOnMethods = {"testMethod1"})
public void testMethod2() {
    // test code
}
```

14. Explain all the helper methods in TestNG?

Priority : it helps to give preference or customize the test cases to execute in our order and return type is int

innvocationCount : it helps to execute the test cases for multiple times and return type is int

threadPoolSize : it helps to execute the test cases in multiple platform and retuen type is int

enable : it helps to disable or enable the test cases and return type is Boolean

dependsOnMethod : it helps to create dependencies between the test cases

name : used to give identifier

group : used for group execution and group execution is occurs based on group helper method

timesOut : it is used to give execution time to the testcase

alwaysRun : it helps to execute the testcase without skip or fail

15. How to generate report in TestNG and types of report?

Generating Reports in TestNG:

- TestNG generates two types of reports:
    1. **HTML Reports:** Generated in the `test-output` folder, providing a graphical view.
    2. **XML Reports:** Generated in the `test-output` folder, providing detailed test execution data.

16. What is data provider annotation, use and Syntax to create Data provider?

DataProvider Annotation:

- It is one of the annotation which is present in testng
- Used to provide multiple data to the same test case .

Syntax:

```
@DataProvider(name = "dataProviderName")
public Object[][] dataProviderMethod() {

    return new Object[][] {{"data1"},
{"data2"}};
}

@Test(dataProvider = "dataProviderName")
public void testMethod(String data) {
    // test code
}
```

17. What is listeners Annotation?

Listeners Annotation:

- Used to listen to events in the TestNG lifecycle. Commonly implemented to capture logs, screenshots, etc.

18. What is ITestListeners and IRetryAnalyzer?

ITestListeners:

- An interface used to define methods that are called on various events in TestNG (e.g., onTestStart, onTestSuccess, onTestFailure).

IRetryAnalyzer:

- An interface used to retry failed test cases.

19. How many ways are there to connect to listeners implemented class?

Ways to Connect Listeners:

1. **Using @Listeners Annotation:** Add the annotation to the test class.
2. **In testng.xml File:** Define listeners in the XML configuration.

20. What is data-driven testing and its advantages?

Data-Driven Testing:

It is a testing methodology which is used to read the data from external files like properties , excel , json ,xml , database etc

Advantages:

- Reduces the number of test scripts.
- Avoid hardcoding
- Data reusability
- Enhances test coverage.
- Allows easy maintenance of test data.

21. How to read the data from excel file?

Reading Data from Excel:

- Use libraries like Apache POI to read data from Excel files.

```
FileInputStream fis = new FileInputStream(new
File("path"));

Workbook workbook = new WorkbookFactory(fis);

Sheet sheet = workbook.getSheet();

String
data=Sheet.getRow().getCell().toString();
```

22. what is properties File and How to read it?

A properties file is a simple key-value pair file often used to store configuration data in Java applications. It's commonly used to keep configuration parameters separate from the code, allowing for easy updates and changes without altering the codebase.

```
Properties properties = new Properties();

FileInputStream fis = new
FileInputStream("path");

properties.load(fis);

String value = properties.getProperty("key");
fis.close();
```

23. How to read data from xml file?

With the help of @parameter annotation prefixing in method and the data will be received from parameter tag in xml file

24. What is hardcoding in automation?

Hardcoding:

- Embedding fixed values directly into the source code. It’s considered a bad practice as it reduces code flexibility and maintainability.

25. What is assertion and why we go for assertion?

Assertion:

- It is a libraries or function in testNg which is used to verify or validate the expected output with the actual output is known as assertion
- And there are two types
- Hard assert and soft assert

26. Difference between hard and soft assert?

Hard assert	**Soft assert**
All methods are static	All methods are non-static
In case of failure ,the execution will be terminated and will not move to next steps	In case of failure ,the execution will not terminate and it will move to next steps
There is no need to create an object for accessing the methods	We should create object for accessing the methods
No need to use assertAll() for exception report	Need to use assertAll() for exception report

27. List out the methods of hard and soft assert?

Hard Assert Methods:

- `assertEquals()`, `assertTrue()`, `assertFalse()`, `assertNull()`, `assertNotNull()`, etc.

Soft Assert Methods:

- Similar methods as hard asserts but used with `SoftAssert` class.

28. What is POM in Selenium?

POM (Page Object Model):

- A design pattern in Selenium which make our test Script independent of the webElement identification
- We write a separate java class for each and every of the webpage
- It is also called as element and actions repository .

29. Why we go for POM?

Advantages of POM:

- Improves test maintenance.
- To avoid staleElementReference exception
- Enhances code readability and reusability.
- Reduces code duplication.
- Data Management etc

30. Explain the source folder in maven project ?

- **src/main/java** → used to store common data like base class , pom class , listeners class , javaUtility class , FileUtility class etc
- **src/main/resources** → used to store driver .exe & files related projects
- **src/test/java** →used to store testScripts
- **src/test/resources** →used to store external files (excel , properties etc)