TESTNG

TestNG is a testing framework inspired from ***JUnit*** and ***NUnit*** but introducing some new functionality that make it more powerful and easier to use. It is an open source automated testing framework; where ***NG** of Test**NG** means **N**ext **G**eneration*. TestNG is similar to JUnit but it is much more powerful than JUnit but still it's inspired by JUnit. It is designed to be better than JUnit, especially when testing integrated classes. Pay special thanks to *Cedric Beust who is the creator of TestNG*. TestNG eliminates most of the limitations of the older framework and gives the developer the ability to write more flexible and powerful tests with help of easy annotations, grouping, sequencing & parametrizing.

Benefits of TestNG:

There are number of benefits but from Selenium perspective, major advantages of TestNG are :

- It gives the ability to produce ***HTML Reports*** of execution
- ***Annotations*** made testers life easy
- Test cases can be ***Grouped & Prioritized*** more easily
- ***Parallel*** testing is possible
- Generates ***Logs***
- Data ***Parameterization*** is possible

Test Case Writing: Writing a test in TestNG is quite simple and basically involves following steps:

- Write the business logic of the test
- Insert TestNG annotations in the code
- Add the information about your test (e.g. the class names, methods names, groups names etc…) in a testng.xml file
- Run TestNG

Benefits of using Annotations:

- It identifies the methods it is interested in by looking up annotations. Hence method names are not restricted to anypattern or format.
- We can pass additional parameters to annotations.
- Annotations are strongly typed, so the compiler will flag any mistakes right away.
- Test classes no longer need to extend anything (such as TestCase, for JUnit 3).

Annotations in TestNG:

- **@BeforeSuite**: The annotated method will be run before all tests in this suite have run.
- **@AfterSuite**: The annotated method will be run after all tests in this suite have run.
- **@BeforeTest**: The annotated method will be run before any test method belonging to the classes inside the tag is run.
- **@AfterTest**: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.
- **@BeforeGroups**: The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.
- **@AfterGroups**: The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.
- **@BeforeClass**: The annotated method will be run before the first test method in the current class is invoked.
- **@AfterClass**: The annotated method will be run after all the test methods in the current class have been run. **@BeforeMethod**: The annotated method will be run before each test method.
- **@AfterMethod**: The annotated method will be run after each test method.
- **@Test**: The annotated method is a part of a test case.

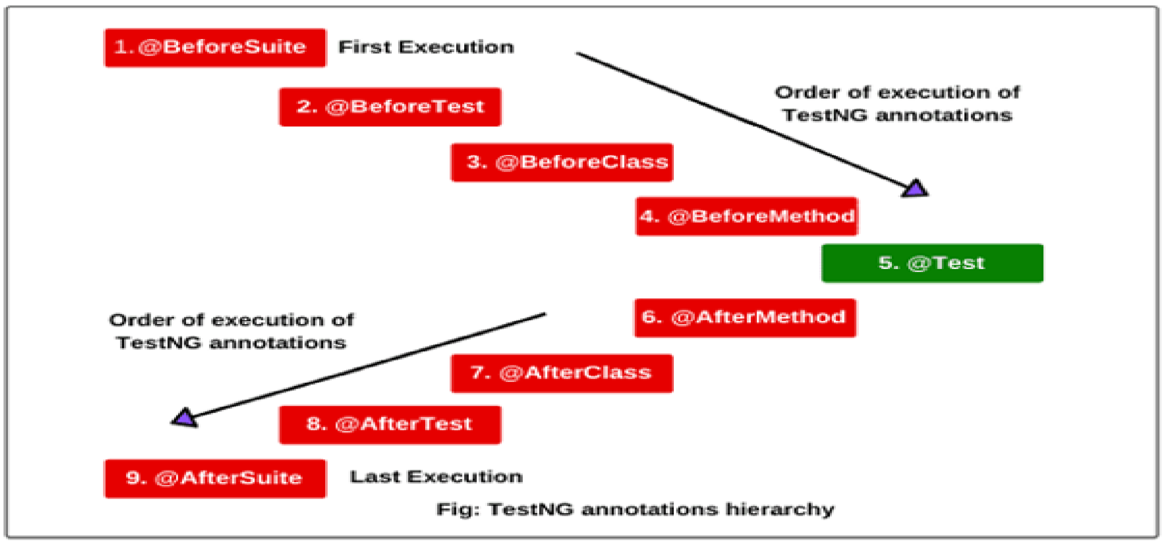


Fig: TestNG annotations hierarchy

Features Of TESTNG:

- Annotations
- Dependent Methods
- Data Driven Testing
- Better Reporting
- Parallel Execution
- Parameterization of test methods
- Groups
- Different Assertions

Commonly used TestNG Assert Methods:

- **Assert.assertEqual(String actual, String expected):**Pass the actual string value and the expected string value as parameters. Validates if the actual and expected values are the same or not.
- **Assert.assertEqual(String actual, String expected, String message):** Similar to the previous method just that when the assertion fails, the message displays along with the exception thrown.
- **Assert.assertEquals{boolean actual, boolean expected):** Takes two boolean values as input and validates if they are equal or not.
- **Assert.assertTrue(condition):**This method asserts if the condition is true or not. If not, then the exception error is thrown.
- **Assert.assertTrue(condition, message):** Similar to the previous method with an addition of message, which is shown on the console when the assertion fails along with the exception.
- **Assert.assertFalse(condition):** This method asserts if the condition is false or not. If not, the nit throws an exception error.
- **Assert.assertFalse(condition, message):** Similar to the previous method but with an addition of a message string whichis shown on the console when the assertion fails ,i.e., the condition is true*.
- **Assert.assertEquals(Object actual, Object expected, String message):** Asserts whether the two objects passed are equal or not. If not, the message and the exception error appears. The message parameter is optional.

- **Assert.assertEquals(String actual, String expected, String message):** Asserts whether two strings are equal or not. If not, the message along with the exception error displays. The message parameter is optional.

Note:- If TestNG class contains more than one test method then all the test method will be executed in alphabetical order. By Default each @Test has priority as 0 lowest priority will get executed first.
@BeforeTest/@AfterTest→behalf of all the @Test method executed only once either in the staring or at the end.
@BeforeMethod/@AfterMethod→it gets executed for each and every @Test method.
By default for each @Test, enabled is set to enabled=true
enabled=true---> consider for execution
enabled=false---> dont consider for execution

Difference Between TestNG & JUNIT: (Advantages of TestNG over JUNIT)

- Advance & easy annotations.
- Execution flow can be set
- Parallel execution is possible.
- Concurrent execution of test script is possible
- Test case dependencies can set.

TestNG Annotation Attributes: While writing the test cases in the TestNG, you need to mention the @Test annotation before the testmethod.

```
@Test
public void testcase1() {
System.out.println("This is testcase1"); }
```

In the above code, we have specified the @Test annotation before the test method, i.e., testcase1(). We can also explicitly specify the attributes in a **@Test** annotation. Test attributes are the test specific,and they are specified at the right next to the **@Test** annotation.

```
@Test(attribute="value")
public void testcase2() {
System.out.println("This is testcase2"); }
```

Some of the common attributes are described below:

- description
- timeOut
- priority
- dependsOnMethods
- enabled
- groups

Other attributes are dataProvider, dataProvider class, dependsOnGroups, expectedExceptions, invocationCount.

→ **<u>description</u>:** It is a string which is attached to the @Test annotation that describes the information about the test.

```
@Test(description="This is testcase1")
public void testcase1() {
System.out.println("HR"); }
```

In the above code, we have added the description attribute. The "**description**" attributeprovides information about the test.

→ **<u>dependOnMethods</u>:** When the second test method wants to be dependent on the first test method, then this could be possible by the use of "**dependOnMethods**" attribute. If the first test method fails, then the dependentmethod on the first test method, i.e., the second test method will not run.

```
@Test
public void WebStudentLogin() {
System.out.println("Student login through web"); }
@Test
public void MobileStudentLogin() {
System.out.println("Student login through mobile"); }

@Test(dependsOnMethods= {"WebStudentLogin"})
public void APIStudentLogin() {
System.out.println("Student login through API"); }
```

We know that the TestNG executes the test methods in alphabetical order so, in the above program, APIStudentLogin() will execute first. However, we want

WebStudentLogin() method to be executed before the execution of the APIStudentLogin() method, so this would only be possible through the "dependsOnMethods" attribute. In the above program, we have specified "dependsOnMethods" attribute in an APIStudentLogin() test method and its value is "WebStudentLogin" which means that WebStudentLogin() method will be executed before the APIStudentLogin() method.

→**When multiple values are passed in a parameter:**

```
@Test(dependsOnMethods= {"testcase3","testcase2"})
public void testcase1() {
System.out.println("This is test case1"); }

@Test
public void testcase2() {
System.out.println("This is test case2"); }

@Test
public void testcase3() {
System.out.println("This is test case3"); }
```

In the above code, testcase1() is dependent on two methods, i.e., testcase2() and testcase3(), whichmeans that these two methods will be executed before the testcase1().

→ **priority:** When no 'priority' attribute is specified then the TestNG will run the test cases in alphabetical order. Priority determines the sequence of the execution of the test cases. The priority can hold the integer values between -5000 and 5000. When the priority is set, the lowest priority test case will run first and the highest priority test case will be executed last. Suppose we have three test cases and their priority values are -5000, 0, 15, then the order of the execution will be 0,15,5000. If priority is not specified, thenthe default priority will be 0.

```
@Test
public void mango() {
System.out.println("I am Mango"); }

@Test(priority=2)
public void apple() {
System.out.println("I am Apple"); }
```

```
@Test(priority=1)
public void watermelon() {
System.out.println("I am Watermelon"); }
```

In the above code, the default priority of mango() test method is 0, so it will be executed first. Thewatermelon() test method will run after mango() method as the priority of watermelon() test method is 2. The apple() test method has the highest priority, so it will be executed last.

→ **enabled:** The 'enabled' attribute contains the boolean value. By default, its value is true. If you want to skip sometest method, then you need to explicitly specify '**false**' value.

```
@Test
public void c_language() {
System.out.println("C language");  }

@Test(enabled=false)
public void jira() {
System.out.println("JIRA is a testing tool"); }

@Test
public void java() {
System.out.println("JAVA language"); }
```

In the above code, the value of the enabled attribute in jira() test method is false, so this method will not be invoked.

→ **groups:** The 'groups' attribute is used to group the different test cases that belong to the same functionality.

```
@Test(groups= {"software company"})
public void infosys() {
System.out.println("Infosys"); }

@Test
public void technip() {
System.out.println("Technip India Ltd"); }
@Test(groups= {"software company"})
public void wipro() {
System.out.println("Wipro"); }
```

testng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="test_suite">
<test name="Software Company">
<groups>
<run>
<include name="software company"/>
</run>
</groups>
<classes>
<class name="com.javatpoint.Software_Company"/>
</classes>
</test> <!-- Test -->
</suite> <!-- Suite -->
```

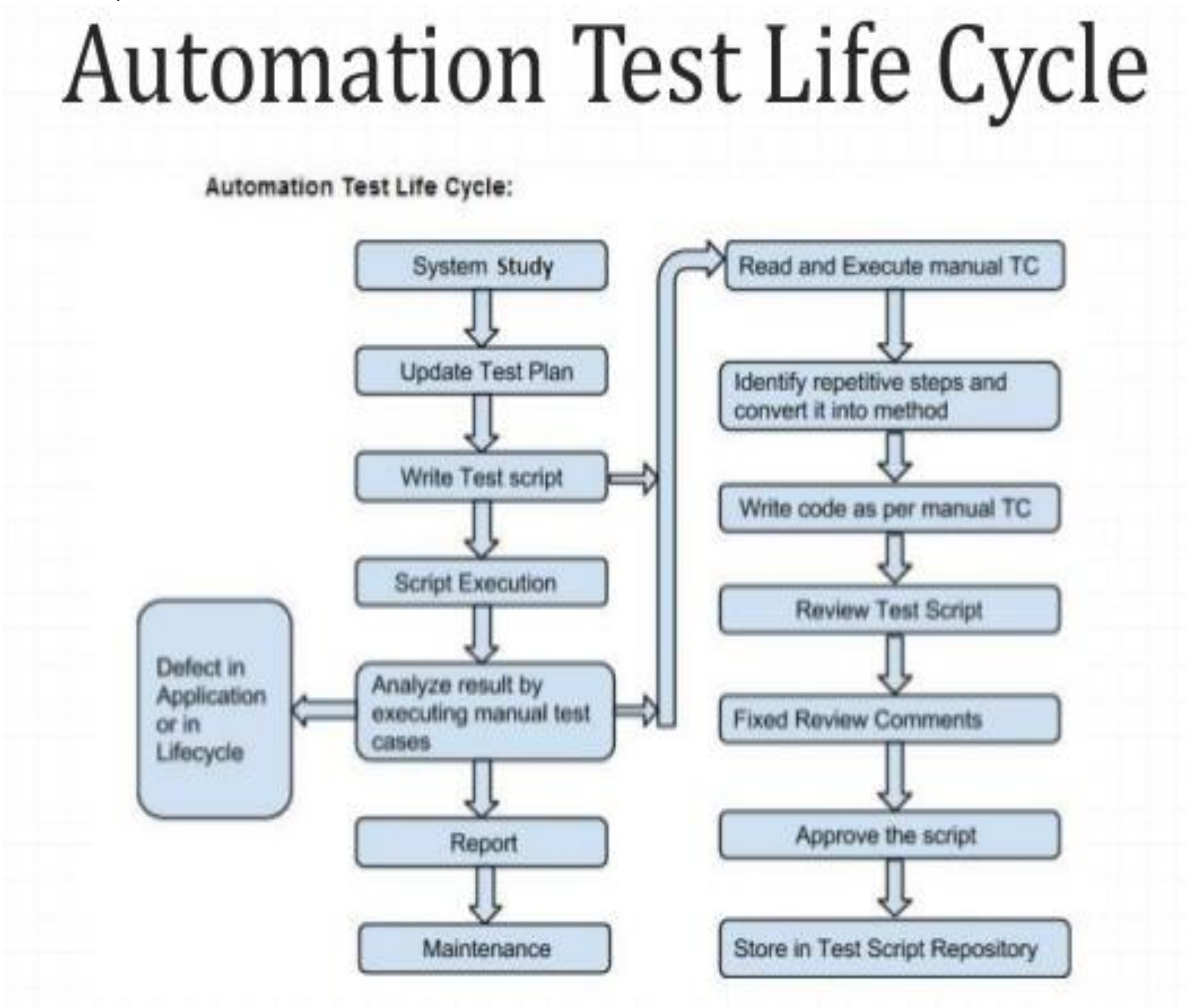
→ **timeOut:** If one of the test cases is taking a long time due to which other test cases are failing. To overcome suchsituation, you need to mark the test case as fail to avoid the failure of other test cases. The timeOut is a time period provided to the test case to completely execute its test case.

```
@Test(timeOut=200)
public void testcase1() throws InterruptedException  {
Thread.sleep(500);
System.out.println("This is testcase1"); }

@Test
public void testcaes2() {
System.out.println("This is testcase2"); }

@Test
public void testcase3() {
System.out.println("This is testcase3"); }
```

In the above code, inside the testcase1() method, we have Thread.sleep(500) which means that the testcase1() method will be executed after 500 milliseconds, but we have provided timeOUT attribute with the value 200 means that the testcase1() will be failed after 200 milliseconds.

testng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="test_suite">
<test name="TimeOut Program">
<classes>
<class name="com.javatpoint.Timeout_program"/>
</classes>
</test> <!-- Test -->
</suite> <!-- Suite -->
```


Automation Test Life Cycle
Automation Test Life Cycle:
System Study
Update Test Plan
Write Test script
Script Execution
Analyze result by executing manual test cases
Defect in Application or in Lifecycle
Report
Maintenance
Read and Execute manual TC
Identify repetitive steps and convert it into method
Write code as per manual TC
Review Test Script
Fixed Review Comments
Approve the script
Store in Test Script Repository

It is the standard process which is to be followed to automate testing of any software application.

System study:

It is a initial phase which is also called as feasibility study. In this phase, we will understand the application by directly referring the manual test cases (To save time). We also select test cases for the automation based on following criteria:

- It should be part of regression test suite.
- This information will be provided by manual testers &. we don't automate non-regression test cases.
- The test case should not contain any manual interventions. Example:
- getting product detail through barcode a scanning
- Paying amount the credit card shopping.
- Generating bills.
- Scanning the images
- Entering OTP which is received the SMS.
- Biomatic scanning for authentication
- Automation tool should recognize the obj problems properly. Examples for unrecognized obj:- Animation Captcha, graphical chart, virtual keyboard, game testing, hardware testing, verification of audio & video streaming.

Note: Because of the above reasons 100% automation is not possible, automation coverage, 60% - 80% is considered as automation coverage.

Automation plan: After identifying the test cases for automation we will capture the detail process of automation such as tool used for automation, framework used for automation, no. of automation engg required, responsibility of each automation engg, starting & ending date of automation & etc. In most of the company they don't create a separate automation test plan document because it will be the part of manual test plan itself.

Developing Test Scripts: This is the actual stage where we can convert manual test cases into automation script. To convert manual test cases into automation Script we follow the below mentioned steps:

- Take the manual test case, read it & execute it at least once so that we will get more clarity on the test case which is to be automated.
- Identify the repetitive steps & convert them into reusable function.
- Write a code as per manual test case steps.
- Execute it & ensure that it is working fine.
- send it for review & fix the review comment
    - **Note:** All the above activities will be done inside a local machine which contains automation tool as well as stable build. While

reviewing test script we should consider following points.

- Each step of manual test case is converted into automation script or not.
- Repetitive steps are converted into function or not
- Coding standard is followed or not
- Script executes without any error

Note – Performing review by other engg in the same team is known as internal review or peer review. If it is done by outside the team such as client review then it is called as external review.

Script Execution: After reviewing the script it will send be send for approval. Once it is approved, it will be stored into test script repository such as LAN machine which contains automation tool & all the approved test Script. As soon as we get a new build, we will install in local machine then we will execute the script. When the scripts are getting executed in the lab machine we will continue our work ie. automating the test cases in the local machine

Analysing the result & publishing the result : After executing the script we check the status of script. If it is failed then we take a responding test case & execute it manually. If the manual test case is failed then it indicates that there is defect in the application. In Such cases, we report the defect & follow the defect life cycle.

If manual test case is passed then it indicates that there is problem with the code in such cases we will debug our code send it for review, again send it for approval & store the updated script in the script repository. Then we prepare script execution report & publish it to the specified team such as development team, manual testing team, manager & client the email.

Maintenance: After converting all the test cases into automation Scripts, we just run the script whenever we get new build. In this phase we will not involved in too much of coding as compared to any of the previous stage. This phase is called as maintenance if required all the automation script will be handovered to the client. So that they can continue the maintenance.

Automation Framework: Automation framework is a standard guideline & the set of processes which is to be followed to convert manual test cases into automation script. We have following types of automation framework in selenium:

- Linear automation framework
- Method driven framework
- Module driven framework
- Keyboard driven framework
- Data driven framework
- Hybrid framework

Keyword Driven Framework: The process of converting keywords present in the excel sheet into respective webdriver code & executing them is called as keyword driven automation framework.

OR

In the keyword driven automation framework, we create various keywords & associate different action or function with each of the keywords, then we create a fun library that contains the logic to read the keywords & call the associated action

Coding Standards:

1. Use Proper Naming Conventions
2. Class Members must be accessed privately
3. Use Underscores in lengthy Numeric Literals
4. Never leave a Catch Blocks empty
5. Use StringBuilder or StringBuffer for String Concatenation
6. Avoid Redundant Initializations
7. Using enhanced for loops instead of for loops with counter
8. Proper handling of Null Pointer Exceptions
9. Float or Double: the right choice?
10. Use of single quotes and double quotes
11. Avoiding Memory leaks
12. Return Empty Collections instead of returning Null elements
13. Efficient use of Strings
14. Unnecessary Objects Creation
15. Proper Commenting

Naming conventions:

class/abstract class/interface → Every word first character should be in upper case i.e: Selenium Utility, ReusableComponent

variable/method → first word should be in lower case, 2nd word onwards 1st character of each word should be in upper case i.e: empld, empName, accountNumber, empResidentAddress, calculateEmpSalary()

constant→ every character should be in upper case i.e:
final int ACCOUNT_NUMBER=123566;

Package: domain.companyname.projectname.modulename
com.google.gmail.inbox
com.xyz.acititime.task

Comments in Java : Block Comments/Single-Line Comments

```
/*
* Here is a block comment.
*/
```

//or

/*Handle the condition. */

End-Of-Line Comments: // TODO: Explain why here.

Documentation Comments: used on the top of the class or method

```
/**
* The Example class provides ...
*/
```

APACHE MAVEN

Apache Maven, is an innovative software project management tool, provides new concept of a project object model (**POM**) file to manage project's build, dependency and documentation. The most powerful feature is ableto download the project dependency libraries automatically. It simplifies the build process like ANT. But it is too much advanced than ANT. Current version is 3. **Ant** and **Maven** both are build tools provided by Apache. The main purpose of these technologies is to ease thebuild process of a project.

Ant	Maven
Ant **doesn't has formal conventions**, so we need to provide information of the project structure in build.xml file.	Maven **has a convention** to place source code, compiled code etc. So we don't need to provide information about the projectstructure in pom.xml file.
Ant is **procedural**, you need to provide information about what to do and when to do through code. You need to provide order.	Maven is **declarative**, everything you define in the pom.xmlfile.
There is **no life cycle** in Ant.	There is **life cycle** in Maven.
It is **a tool** box.	It is **a framework**.
It is **mainly a build tool**.	It is **mainly a project management tool**.
The ant scripts are **not reusable**.	The maven plugins are **reusable**.
It is **less preferred** than Maven.	It is **more preferred** than Ant.

What is Maven?

Maven is a project management and comprehension tool. Maven provides developers a complete build lifecycleframework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

In case of multiple development teams environment, Maven can set-up the way to work as per standards in avery short time. As most of the project setups are simple and reusable, Maven makes life of developer easy while creating reports, checks, build and testing automation setups.

What it does?

Maven simplifies the above mentioned problems. It does mainly following tasks:
1. It makes a project easy to build
2. It provides uniform build process (maven project can be shared by all the maven projects)
3. It provides project information (log document, cross referenced sources, mailing list, dependency list, unittest reports etc.)
4. It is easy to migrate for new features of Maven

Apache Maven helps to manage: Builds, Documentation, Reporting, SCMs, Releases, Distribution.

What is Build Tool?

A build tool takes care of everything for building a process. It does following:

1.Generates source code (if auto-generated code is used)
2.Generates documentation from source code
3.Compiles source code
4.Packages compiled code into JAR of ZIP file
5.Installs the packaged code in local repository, server repository, or central repository

Maven Repository: A **maven repository** is a directory of packaged JAR file with pom.xml file. Maven searches for dependenciesin the repositories. There are 3 types of maven repository:

- Local Repository
- Central Repository
- Remote Repository

Maven searches for the dependencies in the following order:

→ **Local repository** then **Central repository** then **Remote repository**
If dependency is not found in these repositories, maven stops processing and throws an error. Maven **local repository** is located in your local system. It is created by the maven when you run any mavencommand. Maven **central repository** is located on the web. It has been created by the apache maven community itself. Maven **remote repository** is located on the web. Most of libraries can be missing from the central repositorysuch as JBoss library etc, so we need to define remote repository in pom.xml file.

Why Is It Important To Capture Screenshots In Selenium Testing?
The entire point of implementing Selenium test automation is to execute tests without interferences and get results instantaneously. Wouldn't be it good to have an image at hand for when you find a bug? You never know when a test might fail and all your effort will be in vain if you have no proof of the bug. As the saying goes- "A picture is worth a thousand words." That is why capturing a Selenium screenshot would be considered as an important step in testing as it helps to

- Understand the end to end flow of a web application.

- Analyze a bug.
- Track test execution.
- Analyze the behavior of the test in different browsers/environments.
- Trace and examine the reason behind test failure.

When To Capture Screenshots In Selenium Testing?

Selenium screenshots can help you analyze anything from a bug to a web element. Listed below are the possible scenarios when you might consider capturing a Selenium screenshot.

- Capture screenshot for each and every test case
- Capturing screenshots only when there is a failure.
- Capturing screenshots in a specific time interval.
- Capturing screenshots in different browsers.
- Capturing screenshots of specific elements in a webpage.
- Capturing screenshots in specific checkpoints.

RemoteWebDriver() class implements interfaces like WebDriver(), TakesScreenshot() etc and to access these methods we have to downcast the driver object.

```
WebDriver driver = new Chrome Driver();
TakesScreenshot ts = (TakesScreenshot)driver;
```

The method getScreenshotAs() in the interface TakesScreenshot helps in capturing the screenshots and storing it in a specified path. We have another Interface OutputType, which is used to specify the output format of the screenshot such as FILE, BYTES, etc.

```
File file = ts.getScreenshotAs(OutputType.FILE);
try { // save the screenshot taken in destination path
FileUtils.copyFile(file,new File ("./ScreenShot_Folder/Test1_Login.png"));
} catch (IOException e) {
e.printStackTrace();
}
```

//OR

```
TakesScreenshot ts = (TakesScreenshot) driver;
File Source = ts.getScreenshotAs(OutputType.FILE);
File dest = new File
("\\src\\test\\resources\\ScreenShot\\AmazonPage.jpg");
try {
FileUtils.copyFile(Source, dest);
} catch (IOException e) {
        e.printStackTrace();
}
```

In the above destination path "./" indicates the current directory and we can specify the subfolder to hold the screenshots taken and the name in which the screenshots have to be saved. As per the above line, the screenshot taken would be stored in ScreenShot Folder subfolder inside the current working directory in the name of Test1_Login.png file. If the given subfolder doesn't exist, then on first-time execution of the script the folder gets automatically created.

Excel Reading and Update Program:

```
@Test
public void getSheetCount() throws IOException {
//FileLocation
FileInputStream fis=new FileInputStream
(".\\src\\test\\resources\\testData\\AppData.xlsx");
//Create an instance of respective workbook class and provide file location to
its constructor
//XSSFWorkbook workBook=new XSSFWorkbook(fis);
//or more generic way
Workbook workbook=new XSSFWorkbook(fis);
//**** Get Sheet info
int sheetCount=workbook.getNumberOfSheets();
System.out.println("Number of sheets present in excel file: "+sheetCount);
//get all the sheet names
//String sheetName=workbook.getSheetName(0);
for(int i=0;i<sheetCount;i++) {
System.out.println(workbook.getSheetName(i));
}
}

@Test
public void getRowCount() throws IOException {
//FileLocation
FileInputStream fis=new FileInputStream
(".\\src\\test\\resources\\testData\\AppData.xlsx");
//Create an instance of respective workbook class and provide file location to
its constructor
        //XSSFWorkbook workBook=new XSSFWorkbook(fis);
                        //or more generic way
Workbook workbook=new XSSFWorkbook(fis);
//**** Get Sheet info
        //XSSFSheet sheet=workbook.getSheet("Sheet1");
                        //or more generic way
Sheet sheet=workbook.getSheet("Sheet1");
// Row info
                int rowCount=sheet.getLastRowNum(); //actual row count -1
                System.out.println("Number rows in sheet1: "+rowCount);
//for particular raw
                //XSSFRow row=sheet.getRow(1);
                                //or
                 Row row=sheet.getRow(1);
}
@Test
public void getCellCount() throws IOException {
//FileLocation
FileInputStream fis=new FileInputStream
(".\\src\\test\\resources\\testData\\AppData.xlsx");
//Create an instance of respective workbook class and provide file location to
its constructor
        //XSSFWorkbook workBook=new XSSFWorkbook(fis);
                        //or more generic way
        Workbook workbook=new XSSFWorkbook(fis);
// Get Sheet info
        //XSSFSheet sheet=workbook.getSheet("Sheet1");
                        //or more generic way
        Sheet sheet=workbook.getSheet("Sheet1");
// for particular raw
```

```
                    //XSSFRow row=sheet.getRow(1);
                                    //or
                   Row row=sheet.getRow(1);
// Cell info
          int cellCount= row.getLastCellNum(); //actual count
          System.out.println("Cell count in row1 of sheet1: "+cellCount);
//get cell
          //XSSFCell cell=row.getCell(0);
                          //or
          Cell cell=row.getCell(2);
//get value form cell
          String cellValue=cell.getStringCellValue();
          System.out.println("cell value form row1 sheet1: "+cellValue);
          System.out.println("************************");
//get all cell value
          for(int i=0;i<cellCount;i++) {
          System.out.println(row.getCell(i).getStringCellValue());
                    }
          }
@Test
public void updateExcel() throws IOException {
//FileLocation
FileInputStream fis=new FileInputStream
(".\\src\\test\\resources\\testData\\AppData.xlsx");
//Create an instance of respective workbook class and provide file location to
its constructor
          //XSSFWorkbook workBook=new XSSFWorkbook(fis);
                          //or more generic way
          Workbook workbook=new XSSFWorkbook(fis);
// Get Sheet info
          //XSSFSheet sheet=workbook.getSheet("Sheet1");
                          //or more generic way
          Sheet sheet=workbook.getSheet("Sheet1");
// for particular raw
          //XSSFRow row=sheet.getRow(1);
                          //or
          Row row=sheet.getRow(1);
          row.createCell(4).setCellValue("Pass");
 //update excel file content
 //Identify the location where you want to store your new/update data
```

```
FileOutputStream fos=new FileOutputStream
(".\\src\\test\\resources\\testData\\AppData.xlsx");
//write you content into your excel file
   workbook.write(fos);
//flush the content from stream to excel file
   fos.flush();
//close the stream to save the data
   fos.close();
   System.out.println("Updated excel");
}
```

TestNG Test Suite

In any project, you will end up to a place where you need to execute so many test cases on a run. Running a set of test cases together is call executing a Test Suite. Those test cases can be dependent to each other or may have to be executed in a certain order. TestNG gives us the capability to manage our test execution. In TestNG framework, we need to create testng.xml file to create and handle multiple test classes. This is the xml file where you will configure your test run, set test dependency, include or exclude any test, method, class or package and set priority etc.

```
import org.testng.Assert;
import org.testng.annotations.Test;
public class Test1 {
String message = "Vaishali";
MessageUtil messageUtil = new MessageUtil(message);

@Test
public void testPrintMessage() {
System.out.println("Inside testPrintMessage()"); Assert.assertEquals(message,
messageUtil. printMessage()); }
}
```

Now right click on your project and select TestNG then Click on “Convert to TestNG”. Select only those classes which you want to include in “Suite” and then click on “finish”. After click on finish you will get a new “testing.xml” under your project. Now right click on “testing.xml” and “Run AS“ as “TestNG

Suite ”. After doing this you can see the output on “console window ” as well as “TestNG Window”.

TestNG also produce HTML reports. To access those reports go to the Project directory and open test-output folder. TestNG also produce „index.html„ report and it resides in the same test-output folder. This reports gives the link to all the differentcomponent of the TestNG reports like Groups & Reporter Output. Very first tag is the Suite tag, under that it is the Test tag and then the Class tag. You can give any name to the suite and the test butyou need to provide the correct name to the tag which is a combination of your Package name and Test Case name.

<u>Test Case Grouping:</u> Groups is one more annotation of TestNG which can be used in the execution of multiple tests. Let’s say you have hundred tests of class vehicle and in it ten method of car, ten method of scooter and so on. You probably like to run all the scooter tests together in a batch. And you want all to be in a single test suite. With the help of grouping you can easily overcome this situation.

How to do it…

Create two methods for Car, two methods for Scooter and one method in conjunction with Car & Sedan Car. Group them separately with using (groups = { ” Group Name” })

```
import org.testng.annotations.Test;
public class Grouping {

@Test (groups = { "Car" })

public void Car1()
System.out.println("Batch Car - Test car 1"); }

@Test (groups = { "Car" })

public void Car2() {

System.out.println("Batch Car - Test car 2");

}

@Test (groups = { "Scooter" })
```

```
public void Scooter1() {
System.out.println("Batch Scooter - Test scooter 1"); }

@Test (groups = { "Scooter" })

public void Scooter2() {
System.out.println("Batch Scooter - Test scooter 2"); }

@Test (groups = { "Car", "Sedan Car" })

public void Sedan1() {
System.out.println("Batch Sedan Car - Test sedan 1"); }
}
```

<u>Create a testng xml</u>

```
<suite name="Suite">
<test name="Practice Grouping">
<groups>
<run>
<include name="Car" />
</run>
</groups>
<classes>
<class name="automationFramework.Grouping" />
</classes>
</test>
</suite>
```

<u>Clubbing of groups is also possible</u>

```
<suite name="Suite">
<test name="Practice Grouping">
<groups>
<define name="All">
<include name="Car"/>
<include name="Scooter"/>
</define>
<run>
<include name="All"/>
```

```
</run>
</groups>
<classes>
<class name="automationFramework.Grouping" />
</classes>
</test>
</suite>
```

<u>TestNG Reporters:</u> TestNG is a Framework and so far we have already seen the many different powerful features of TestNG. It almost gives you all the important things you are required to complete the Framework.

<u>TestNG Reporter Logs:</u> TestNG also gives us the logging facility for the test. For example during the running of test case user wants some information to be logged in the console. Information could be any detail depends upon the purpose. Keeping this in mind that we are using Selenium for testing, we need the information which helps the User to understand the test steps or any failure during the test case execution. With the help of TestNG Logs it is possible to enable logging during the Selenium test case execution.

In selenium there are two types of logging. High level logging and Low level logging. In low level logging you try to produce logs for the every step you take or every action you make in your automation script. In high level logging you just try to capture main events of your test. I perform low level logging with log4j and high level logging with testng reporter logs.

How to do it…

Write a test case for Sign In application and implement Log4j logging on every step. Insert Reporter logs on the main events of the test. Log4j logging will help you to report a bug or steps taken during the test, on the other hand reporters log will help you to share the test status with leadership. As leadership is just interested in the test results, not the test steps.

<u>TestNG Parameters:</u> Everybody knows the importance of Parametrization in testing and in automation testing. It allows us to automatically run a test case multiple times with different input and validation values. As Selenium Webdriver is more an automated testing framework than a ready-to-use tool, you will have to put in some effort to support data driven testing in your automated tests. TestNG again gives us another interesting feature called TestNG Parameters. TestNG lets you pass parameters directly to your test methods with your testng.xml.

How to do it…

Let me take a very simple example of LogIn application, where the username and password is required to clear the authentication. Create a test on my demo OnlineStore application to perform LogIn which takes the two string argument as username & password. Provide Username & Password as parameter using TestNG Annotation. The parameter would be passed values from testng.xml

```
<suite name="Suite">
<test name="ToolsQA">
<parameter name="sUsername" value="yourusername"/>
<parameter name="sPassword" value="yourpassword"/>
<classes>
<class name="automationFramework.TestngParameters" />
</classes>
</test>
</suite>
```

TestNG DataProviders: When you need to pass complex parameters or parameters that need to be created from Java (complex objects, objects read from a property file or a database, etc…), in such cases parameters can be passed using Dataproviders. A Data Provider is a methodannotated with @DataProvider. A Data Provider returns an array of objects.

How to do it…

Define the method credentials() which is defined as a Dataprovider using the annotation. This method returns array of object array. Add a method test() to your DataProviderTest class. This method takes two strings as input parameters. Add the annotation @Test(dataProvider = “Authentication”) to this method. The attribute dataProvider is mapped to“Authentication”. As the test data is provided two times, the above test executed two times completely.

Testing your website with multiplecombinations of browsers is known as **Cross Browser testing.**

Your site will look different in different browsers. That's because browsers understand some code slightly differently. Your designer should be testing to make sure that your site works well in all modern browsers. But as a tester we need to make sure thatfunctionality should at least tested on Internet Explorer, Firefox, Safari & Google Chrome browser.

Multi Browser Testing using Selenium TestNG:

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.

How to do it…

Create your Script to test a LogIn application using TestNG class. Pass ,Browser Type as parameters using TestNG annotations to the before method of the TestNG class. This method will launchonly the browser, which will be provided as parameter. You can set any number of Browsers here and just for the example purpose I have set up only two main browsers.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite" parallel="none">
<test name="FirefoxTest">
<parameter name="browser" value="firefox" />
<classes>
<class name="automationFramework.MultiBrowser" />
</classes>
</test>
<test name="IETest">
<parameter name="browser" value="ie" />
```

```
<classes>
<class name="automationFramework.MultiBrowser" />
</classes>
</test>
</suite>
```

Re-run Failed Test Cases:

You must have seen random failure during an automated test run. These failures might not necessarily be because of product bugs. These failure can be because of following reasons:

- Random browser issues or browser becoming unresponsive
- Random machine issues
- Server issues like unexpected delay in the response from server

These failure are genuine and should be investigated but these failures are not necessarily because of a bug in the product. TestNG provides a wonderful feature using which you can retry a test case multiple times before declaring it as Failed. In order achieve this we have to first understand the org.testng.IRetryAnalyzer interface.

```
public interface IRetryAnalyzer {
int counter = 0;
int retryLimit = 2;
/** Returns true if the test method has to be retried, false otherwise.
* @param result The result of the test method that just ran.
* @return true if the test method has to be retried, false otherwise.
* TestNg will call this method every time a test fails. So we can put some code
in here to decide when to rerun the test.
*/
@Override
public boolean retry(ITestResult result) {
if(counter < retryLimit) {
counter++;
return true;
```

}
return false; }

- "This interface has only one method **public boolean retry(ITestResult result);**
- This method will be called once a test method fails. You can get the details of the test from ITestResult input argument to this method. This method implementation should return true if you want to re-execute your failed test and false if you don't want to re-execute your test.
- Usually the implementation of this interface decides on how many times to retry a failed tests based on a fixed counter or a complex logic based on your requirements.

→ Specifying retryAnalyzer attribute in the @Test annotation, We can do this by simply using following syntax to

@Test(retryAnalyzer="IRetryAnalyzer Implementing class")

→ Specifying retryAnalyzer during runtime:

- In this case we need to implement ITestAnnotationTransformer interface.
- ITestAnnotationTransformer interface falls under a broad category of interfaces called TestNG Listeners. This interface definition looks like this:

public class Annotation Transformer implements IAnnotationTransformer {
@Override
public void transform(ITestAnnotation annotation, Class testClass,
Constructor testConstructor, Method testMethod) {
}

- Once we have the implementation of IAnnotationTransformer, we just need to add it as a listener in the testng run xml. Like this:

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="RetryFailed Tests" verbose="1">
<listeners>
<listener class-name="packagename.AnnotationTransformer"/>
</listeners>
<test name="RetryMulitple">
<classes>
```

```
<class name=" packagename.Test001"/>
</classes>
</test>
</suite>
```

What are Extent Reports?

ExtentReports is an open-source reporting library useful for test automation. It can be easily integrated with major testing frameworks like JUnit, NUnit, TestNG, etc. These reports are HTML documents that depict results as pie charts. They also allow the generation of custom logs, snapshots, and other customized details. Once an automated test script runs successfully, testers need to generate a test execution report. While TestNG does provide a default report, they do not provide the details. Extent Reports in Selenium contain two major, frequently used classes:

- ExtentReports class
- ExtentTest class

Syntax :

ExtentReports reports = new ExtentReports("Path of directory to store the resultant HTML file", true/false);
ExtentTest test = reports.startTest("TestName");

The ExtentReports class generates HTML reports based on a path specified by the tester. Based on the Boolean flag, the existing report has to be over written or a new report must be generated. 'True' is the default value, meaning that all existing data will be overwritten. The ExtentTest class logs test steps onto the previously generated HTML report. Both classes can be used with the following built-in methods:

- **startTest:** Executes preconditions of a test case
- **endTest:** Executes postconditions of a test case
- **Log:** Logs the status of each test step onto the HTML report being generated
- **Flush:** Erases any previous data on a relevant report and creates a whole new report

A Test Status can be indicated by the following values:

- PASS

- FAIL
- SKIP
- INFO

Syntax:

reports.endTest();

test.log(LogStatus.PASS, "Test Passed");

test.log(LogStatus.FAIL, "Test Failed");

test.log(LogStatus.SKIP, "Test Skipped");

test.log(LogStatus.INFO, "Test Info");

The Log method takes into account two parameters, the first being the test status and the second being the message to be printed onto the generated report.

Benefits of using Extent Reports:

- They can be integrated with TestNG and JUnit
- If required, screenshots can be captured and displayed for each step in a test
- They allow testers to track multiple test case runs in a single test suite
- They show the time needed for test execution
- They can be customized to graphically represent each step in a test

How to generate Extent Reports:

- Add extentreports dependency into pom.xml
```
<dependency>
<groupid>com.aventstack</groupId>
<artifactId>extentreports</artifactId>
<version>2.41.2<version>
</dependency>
```
- Test execution commences with the **startTest** method. It also initializes the Extent Reports object. Any valid user-defined path can be the parameter passed onto the Extent Reports object.
  @Test: This class automates the following actions:

- Open Chrome browser with the url https://www.google.com
- After the page opens, validate page title with the expected
- Log the test case status as PASS/FAIL using the log method explained before

@AfterClass: Executes post conditions of the test case-ending the test (using **endTest** method) and flushing the report.

Note: Don't forget to use the flush() method, since the report will not be generated otherwise.

```
public class ExtentDemo {
static ExtentTest test;
static ExtentReports report;
@BeforeClass
public static void startTest() {
report = new ExtentReports(".\\ExtentReport\\ExtentReportResults.html");
test = report.startTest("GooglePageReport"); }
@Test
public void extentReportsDemo() throws IOException {
System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
driver.get("https://www.google.com");
if (driver.getTitle().equals("Google1")) {
test.log(LogStatus.PASS, "Navigated to the specified URL successfully and page
tile is also validated");
//System.out.println("Navigated to the specified URL successfully and page tile
is also validated"); }
else {
test.log(LogStatus.FAIL,test.addScreenCapture(screenShot(driver)), "Google
page validation got failed");
//System.out.println("Google page validation got failed"); }
}
@AfterClass
public static void endTest() {
report.endTest(test);
report.flush(); }

public static String screenShot(WebDriver driver) throws IOException {
File scrFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);
File Dest = new File("src/../BStackImages/" + System.currentTimeMillis()+
".png");
String errflpath = Dest.getAbsolutePath();
FileUtils.copyFile(scrFile, Dest);
return errflpath; }
}
```

//OR

```
public static String screenShot(WebDriver driver) throws IOException {
TakesScreenshot ts = (TakesScreenshot) driver;
File source = ts.getScreenshotAs(OutputType.FILE);
File Dest = new File(".\\src\\test\\resources\\ScreenShot\\flipKart.jpg");
FileUtils.copyFile(source, Dest );
String file = Dest.getAbsolutePath();
return file;
}
```

By capturing screenshots, testers can better identify what went wrong when the software acted erroneously during a test. Capture screenshots only when a test fails, since they consume a lot of memory. Use the log method because it uses the **addScreenCapture** method of **Extent Test** class to get a screenshot and add it to the **Extent Report.**
test.log(LogStatus.FAIL,test.addScreenCapture(capture(driver))+ "Test Failed");

Page Object Model

What is POM?
The easiest way to describe a POM in a maven project is, it is nothing but the core element of any mavenproject. Basically any maven project consists of one configurable file called pom.xml, which stands for the abbreviation **Project Object Model**. This pom.xml will always be located in the root directory of any maven project. This file represents the very basic and fundamental unit in maven. The **pom.xml** basically contains the information related to the project which is built or to be built in. It contains all the necessary information about the configuration details, dependencies included and plug-insincluded in the project. In simple, it contains the details of the build life cycle of a project.

Below are some of the configurations that can be handled in the pom.xml file

- Dependencies used in the projects (Jar files)
- Plugins used (report plugin)
- Project version
- Developers involved in the project
- Mailing list
- Reporting
- Build profiles

Page Object Pattern/Model

- Page Object model is an object design pattern in Selenium, where web pages are represented as classes, and the various elements on the page are defined as variables on the class. All possible user interactions can then be implemented as methods on the class
- Page Object Model is a design pattern to create Object Repository for web UI elements.
- Under this model, for each web page in the application there should be corresponding page class.
- This Page class will find the WebElements of that web page and also contains Page methods which perform operations on those WebElements.
- In order to support Page Object model, we use **Page Factory**. Page Factory in Selenium is an extension to Page Object and can be used in various ways. In this case we will use Page Factory to initialize web elements that are defined in web page classes or Page Objects.
- Web page classes or Page Objects containing web elements need to be initialized using Page Factory before the web element variables can be used. This can be done simply through the use of **initElements** function on PageFactory.
- Name of these methods should be given as per the task they are performing i.e., if a loader is waiting for payment gateway to be appear, POM method name can be **waitForPaymentScreenDisplay().**

Advantages of using Page Object Pattern:

- Easy to Maintain
- Easy Readability of scripts
- Reduce or Eliminate duplicacy
- Re-usability of code

- ➢ Reliability

What is Page Factory?

Page Factory is an inbuilt page object model concept for Selenium WebDriver but it is very optimized. Here as well we follow the concept of separation of Page Object repository and Test methods. Additionally with the help of PageFactory class we use annotations **@FindBy** to find WebElement. We use **intiElements** method(static) of PageFactory class to initialize web elements.

@FindBy can accept tagName, partial LinkText, name, linkText, id, css, className, xpath as attributes.

```
WebDriver driver;              // inside the webpage class constructor
public HomePage(WebDriver driver) {
this.driver=driver;
PageFactory.initElements(driver, this); }
//OR
HomePage page = PageFactory.intElements (driver, HomePage.class)
//OR
HomePage page = new HomePage (driver);
PageFactory.initElements (driver, page);
```

```
 @FindBy(id="logoutLink")
private WebElement logoutLink;
public WebElement getLogoutLink() {     //getter method
return logoutLink; }
```

→ findElements can be written as:

```
WebElement username = driver. find Element(By.id("username"));
```

//or

```
 By usr = By.id("username");
WebElement username = driver.findElement(usr);
```

//or

```
@FindBy(id="username")
WebElement username;
```

Examples:

```
@FindBy(xpath="//input[@id= 'username']")
WebElement username;

@FindBy (css="#username")
WebElement username;
```

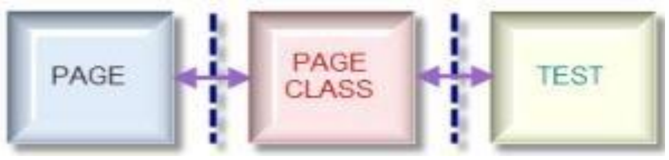
→Moving on, you will often come across situations where you need to find a list of elements on a page, and that iswhen ***@FindBys*** comes in handy:

@FindBys(@FindBy(css=”div[class=’yt-lockup-vido’]”)))
private List<WebElement> videoElements;

→ Additionally, you can use ***@FindAll*** with multiple ***@FindBy*** annotations to look for elements that match any ofthe given locators.

How does it work?

Page Object Pattern is a pattern that displays user interface as a class. In addition to user interface, functionalityof the page is also described in this class. This provides a bridge between page and test.



Why should it be used?

- Simple and clear tests.
- Good support of tests, because everything is stored in one place.
- Easy creation of new tests. In fact, tests can be created by a person not knowing the features ofautomation tools.
- Simple example of using Page Object Pattern and PageFactory in Selenium WebDriver.

Why use PageFactory?

When we initialize the page using PageFactory, then, if annotation FindBy is not specified, PageFactory searches for elements on the page which name or id attributes match the name of WebElement. As a result, we actually do not have to worry about searching for elements on the page. If the element will appear on thepage, so when we turn to it, it will be initialized. Example of webpage and webpage script:

WEBPAGE

```
public class VtigerLoginPage extends SeleniumUtility {
public VtigerLoginPage(WebDriver driver) {
PageFactory.initElements(driver, this); }

@FindBy(name="username")
private WebElement userName;

@FindBy(name="password")
private WebElement password;

@FindBy(xpath="//button[text()='Sign in']")
private WebElement signInButton;

public WebElement getUserName() {
return userName; }

public WebElement getPassword() {
return password; }

public WebElement getSignInButton() {
return signInButton; }

public void loginIntoVtiger(String userId,String pwd) {
typeInput(userName, userId);
typeInput(password, pwd);
clickOnElement(signInButton); }

public String getHomePageTitle() {
return getCurrentTitleOfApplication(); }
}
```

WEBPAGE SCRIPT:

```
public class TestVtigerLogin extends SeleniumUtility{
WebDriver driver;
VtigerLoginPage getVtigerLoginPage;

@BeforeTest
public void openBrower() {
driver=setUp("chrome", "https://demo.vtiger.com/vtigercrm/index.php");
getVtigerLoginPage=new VtigerLoginPage(driver); }

@Test
public void testLogin() {
```

```
getVtigerLoginPage.loginIntoVtiger("admin", "Test@123");
String actualTtile=getVtigerLoginPage.getHomePageTitle();
String expectedTitle="Dashboard";
Assert.assertEquals(actualTtile, expectedTitle); }

@AfterTest
public void closeBrowser() {
cleanUp(); }
}
```

POM OVERLOOK

1. **Language**: In our Selenium Project we are using **Java language.**
2. **Type of Framework**: In our project, we are Using **Behavioural-driven Framework** by using Page Object Model design pattern with page Factory.
3. **POM:** As per the Page Object Model, we have **maintained a class for every web page.** Each web page has a separate class and that class holds the **functionality and members of that web page**. Separate classes for every individual test.
4. **Packages**: We have separate packages for **pages and Tests**. All the web page related classes come under **Pages** package and all the tests related classes come under **Tests** package.
5. **Test Base Class**: Test Base class (TestBase.java) deals with all **the common functions** used by all the pages. This class is responsible for loading the configurations from properties files, Initializing the **WebDriver, Implicit Waits, Extent Reports and also to create the object of FileInputStream** which is responsible for pointing towards the file from which the data should be read.
6. **Utility class(AKA Functions Class)**: Utility class(TestUtil.java) stores and handles the functions(**The code which is repetitive in nature** such as **waits, actions, capturing screenshorts, accessing excels, sending email** etc.,) which can be commonly used across the entire framework. The reason behind creating utility class is to achieve reusability. This class extends the TestBase class to inherit the properties of TestBase in TestUtil.
7. **Properties file**: This file (**config.properties**) stores the information that remains static throughout the framework such as browser specific information, **application URL screenshots path** etc, all the

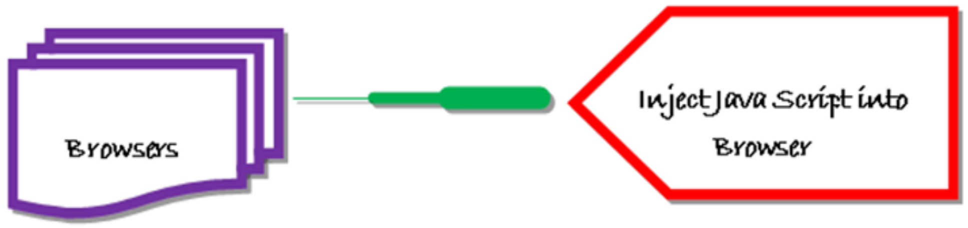
details which change as per the environment and authorization such **as URL, Login Credentials are kept in the config.properties file**. Keeping these details in a separate file makes easy to maintain.

8. **Screenshots**: Screenshots will be captured and stored in a separated folder and the screenshots of a failed test cases will be added in the extend reports.
9. **Test Data**: All the historical **test data will be kept in excel sheet** (controller.xlsx). By using 'controller.xlsx', we pass test data and handled data driven testing. We use Apache POI to handle excel sheets.
10. **TestNG**: Using TestNG for **Assertions, Grouping and parallel execution**.
11. **BDD framework i**.e. Behaviour driven Development is a software development approach that allows the tester/business analyst to create test cases in simple text language (English). The simple language used in the scenarios helps even non-technical team members to understand what is going on in the software project.
12. **Maven**: Using Maven for **build, execution, and dependency purpose.** Integrating the TestNG dependency in POM.xml file and running this POM.xml file using Jankins.
13. **Version Control Tool**: We use **1** as a repository to store our test scripts.
14. **Jenkins**: By using Jenkins CI (Continuous Integration) Tool, we execute test cases on daily basis and for nightly execution based on the schedule. Test Result will be sent to the peers using Jenkins.
15. **Extend Reports**: For the reporting purpose, we are using Extent Reports. It generates beautiful HTML reports. We use the extent reports for maintaining logs and to include the screenshots of failed test cases in the Extent Report.

JavaScriptExecutor

What is JavaScriptExecutor?

JavaScriptExecutor is an Interface that helps to execute JavaScript through Selenium Webdriver. JavaScriptExecutor provides two methods "executescript" & "executeAsyncScript" to run javascript on the selected window or current page.



Why do we need JavaScriptExecutor?

In Selenium Webdriver, locators like XPath, CSS, etc. are used to identify and perform operations on a web page. In case, these locators do not work you can use JavaScriptExecutor. You can use JavaScriptExecutor to perform an desired operation on a web element. Selenium supports javaScriptExecutor. There is no need for an extra plugin or add-on. You just need to import (**org.openga.selenium.JavascriptExecutor**) in the script as to use JavaScriptExecutor.

<u>Components of JavascriptExecutor in Selenium:</u>

JavascriptExecutor consists of two methods that handle all essential interactions using JavaScript in Selenium.

1. executeScript method - This method executes the test script in the context of the currently selected window or frame. The script in the method runs as an anonymous function. If the script has a return statement, the following values are returned:

- For an HTML element, the method returns a **WebElement**.
- For a decimal, the method returns **Long**.
- For a non-decimal number, the method returns **Long**.
- For a Boolean, the method returns **Boolean**.
- For other cases, the method returns a **String**.

2. executeAsyncScript method - This method executes the asynchronous piece of JavaScript on the current window or frame. An asynchronous script will be executed while the rest of the page continues parsing, which enhances responsiveness and application performance.

<u>Scenarios To Use JavaScriptExecutor in Selenium:</u>Let's examine some scenarios we could handle using JavaScriptExecutor Interface for Selenium test automation.

//Creating reference variable of JavascriptExecutor by casting from WebDriver interface

```
JavascriptExecutor js=(JavascriptExecutor)driver;
```

1. To Click on a Button

```
js.executeScript("document.getElementById('enter element id').click();");
```

//or

```
js.executeScript("arguments[0].click();", webelement);
```

2. To Type Text in a Text Box without using sendKeys() method

```
js.executeScript("document.getElementById(id').value='someValue";");

js.executeScript("document.getElementById('Email').value='Selenium
Testing.com';");
```

3. To Handle Checkbox by passing the value as true or false

```
js.executeScript("document.getElementById('enter element id').checked =
false;");
```

4**. To generate Alert Pop window in Selenium Webdriver**

```
js executeScript("alert("Welcome To Selenium Testing");");
```

5. To refresh browser window using Javascript

```
js.executeScript("history.go(0)");
```

//or

```
js.executeScript("location.reload()");
```

6. To get the innertext of the entire webpage in Selenium

```
String innerText = js.executeScript(" return document.documentElement.
innerText;").toString();
System.out.println(innerText);
```

7. To get the Title of our webpage

```
String titleText = js.executeScript("return document.title;").toString();
System.out.println(titleText);
```

8. To get the domain name

```
String domainName= js.executeScript("return document.domain;").toString();
System.out.println(domainName);
```

9. To get the URL of a webpage

```
String url= js.executeScript("return document.URL;").toString();
System.out.println(url);
```

10. To get the Height and Width of a web page

```
js.executeScript("return window.innerHeight;").toString();
js.executeScript("return window.innerWidth;").toString();
```

11. To find a hidden element in selenium using JavaScriptExecutor

```
js.executeScript("arguments[0].click();", element);
```

12. To navigate to a different page using Javascript

```
js.executeScript("window.location = 'https://www.lambdatest.com");
```

13. To perform Scroll on an application using Selenium:

a) To scroll the page vertically for 500px

```
js.executeScript("window.scrollBy(0,500)");
```

b) To scroll the page vertically till the end

```
js.executeScript("window.scrollBy(0,document.body.scrollHeight)");
```

14. Adding an element in DOM

We can also add an element in DOM if required. However, as are only concerned with mimicking user interactions in the browser, this option may not be used.

```
js.executeScript("var btn=document.createElement('newButton');"
+"document.body.appendChild(btn);");
```

Selenium 3rd Party Support tools for Window feature: While working on real time application you might get some windows based feature like:

- Authentication popup
- File download popup
- File upload popup
- Security popups

These windows popups can't be handled by selenium directly, to support selenium allows you to use 3rd party tools like:

- **AutoIT**
- **Sikuli** (It's a image based tool, which creates problem when you run you script in different system which are having different screen resolution)
- **Robot** Class (Its creates a problem while running script on server machine or if you are running script on local machine then you can't perform any other operation)

→ Write autoIT script and save the file extension with ".au3" extension. After compilation of ".au3", you will get ".exe" file. Now you can run .exe file to handle any windows based popup using Selenium
Runtime.getRuntime().exec(".exe file path");
Runtime.getRuntime().exec(".exe file path "+" "+"path of the file");

BDD with Selenium

"Cucumber is a testing tool that supports **Behavior Driven Development (BDD)**. It offers a way to write tests that anybody can understand, regardless of their technical knowledge. In BDD, users (business analysts, product owners) first write scenarios or acceptance tests that describe the behavior of the system from the customer's perspective, for review and sign-off by the product owners before developers write their codes. It lets us define application behavior in plain meaningful English text using a simple grammar defined by a language called **Gherkin**. Cucumber itself is written in **Ruby**, but it can be used to **"test"** code written in Ruby or other languages including but not limited to Java ,C# and Python.

Advantages of Cucumber:

- It is helpful to involve business stakeholders who can't easily read code Cucumber Testing tool focuses on end-user experience
- Style of writing tests allow for easier reuse of code in the tests
- Quick and easy set up and execution
- Cucumber test tool is an efficient tool for testing

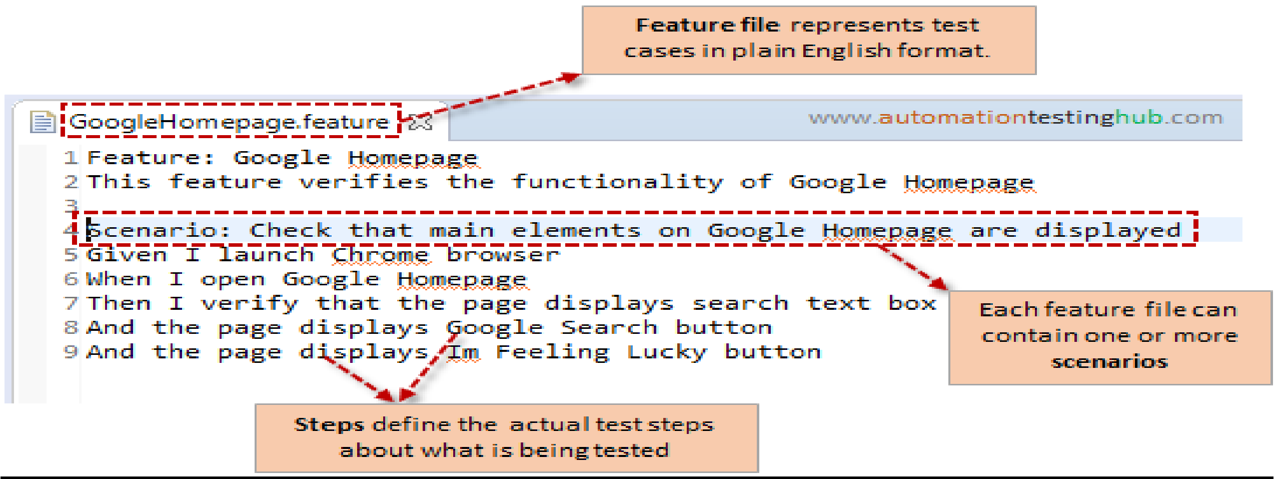
For every cucumber project there is a single directory at the root of the project named **"features"**. This is where all of your cucumber features will reside. In this directory you will find additional directories, which is **step_definition**. Cucumber framework mainly consists of three major parts- **Feature File**, **Step Definitions**, and the **Test Runner File**.

Feature File:

A standalone unit or a single functionality (such as a login) for a project can be called a Feature. Each of these features will have scenarios that must be tested using Selenium integrated with Cucumber. A file that stores data about features, their descriptions, and the scenarios to be tested, is called a **Feature**

File. Cucumber tests are written in these Feature Files that are stored with the extension- "**.feature**". A Feature File can be given a description to make the documentation more legible.

Example: The Login function on a website
Feature File Name: *googleSearch.feature*
Description: The user shall be able to login upon entering the correct username and password in the correctfields. The user should be directed to the homepage if the username and password entered are correct. Keywords such as GIVEN, WHEN, and THEN used to write the test in Cucumber are called **Annotations**.



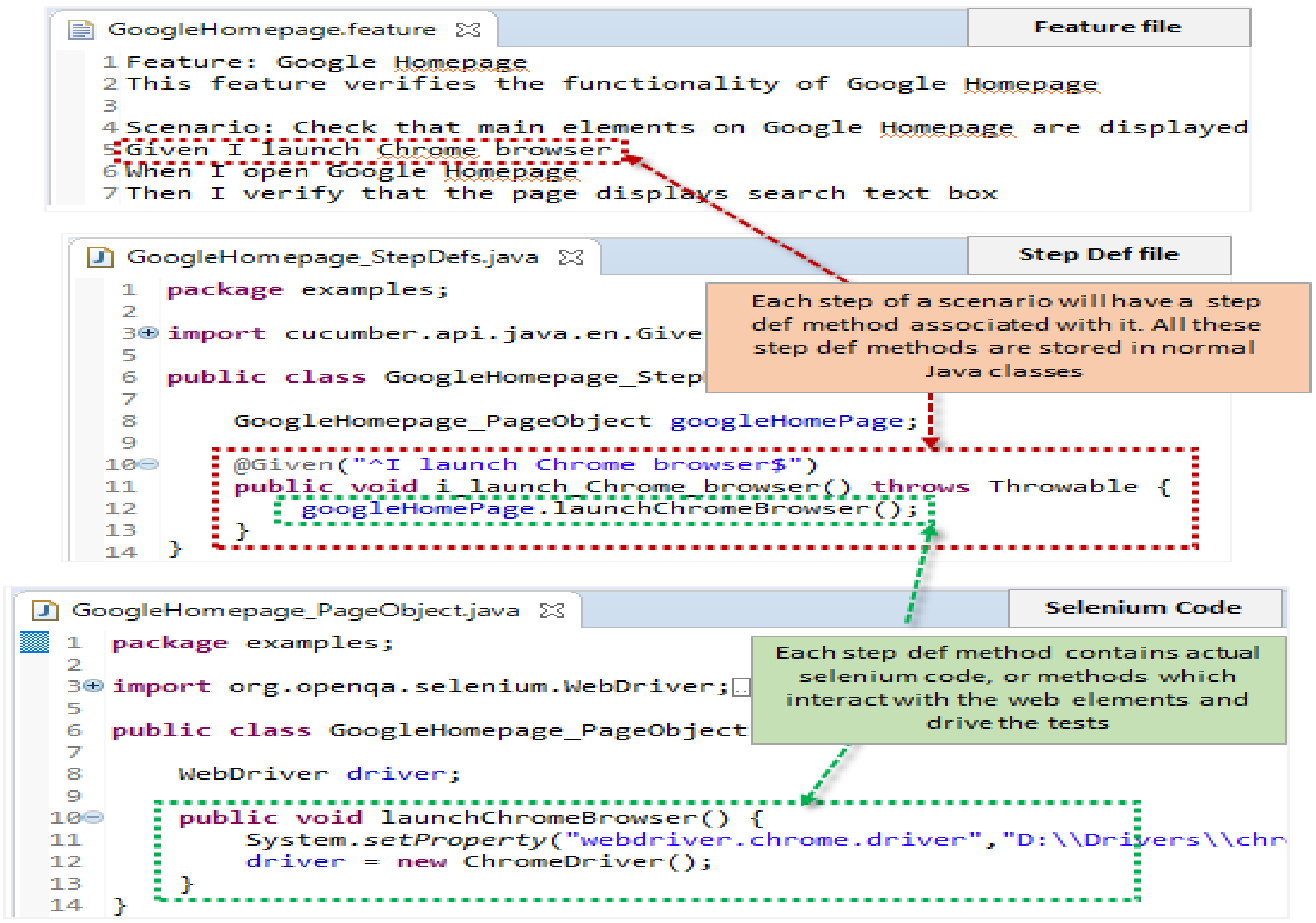
Feature file represents test cases in plain English format.
GoogleHomepage.feature
www.automationtestinghub.com
Each feature file can contain one or more scenarios
Steps define the actual test steps about what is being tested

```
Feature: Google Homepage
This feature verifies the functionality of Google Homepage

Scenario: Check that main elements on Google Homepage are displayed
Given I launch Chrome browser
When I open Google Homepage
Then I verify that the page displays search text box
And the page displays Google Search button
And the page displays Im Feeling Lucky button
```

Step Definitions: Now that the features are written in the feature files, the code for the related scenario has to be run. To know which batch of code needs to be run for a given scenario, Steps Definitions come into the picture. A Steps Definitions file stores the mapping data between each step of a scenario defined in the feature file and the code to be executed. Step Definitions can use both Java and Selenium commands for the Java functions written to map a feature file to the code.

Test Runner File:

To run the test, one needs a **Test Runner File**, which is a JUnit Test Runner Class containing the Step Definition location and the other prim ary metadata required to run the test. The Test Runner File uses the **@RunWith()** Annotation from JUnit for executing tests. It also uses the **@CucumberOptions** Annotation to define the location of feature files, step definitions, reporting integrations, etc.



```
Feature: Google Homepage
This feature verifies the functionality of Google Homepage

Scenario: Check that main elements on Google Homepage are displayed
Given I launch Chrome browser
When I open Google Homepage
Then I verify that the page displays search text box
```

Each step of a scenario will have a step def method associated with it. All these step def methods are stored in normal Java classes

```
package examples;

import cucumber.api.java.en.Give

public class GoogleHomepage_Step

    GoogleHomepage_PageObject googleHomePage;

    @Given("^I launch Chrome browser$")
    public void i_launch_Chrome_browser() throws Throwable {
        googleHomePage.launchChromeBrowser();
    }
}
```

Each step def method contains actual selenium code, or methods which interact with the web elements and drive the tests

```
package examples;

import org.openqa.selenium.WebDriver;

public class GoogleHomepage_PageObject

    WebDriver driver;

    public void launchChromeBrowser() {
        System.setProperty("webdriver.chrome.driver","D:\\Drivers\\chr
        driver = new ChromeDriver();
    }
}
```

<u>Important Terms used in Gherkin:</u>

- ➢ Feature
- ➢ Background
- ➢ Scenario
- ➢ Given
- ➢ When
- ➢ Then
- ➢ And
- ➢ But
- ➢ Scenario Outline Examples

<u>Feature</u>: The file should have extension .feature and each feature file should have only one feature. The feature keyword being with the Feature: and after that add, a space and name of the feature will be written.

Scenario: Each feature file may have multiple scenarios, and each scenario starts with Scenario: followed by scenario name.

Background: Background keyword helps you to add some context to the scenario. It can contain some steps of the scenario, but the only difference is that it should be run before each scenario.

Given: The use of Given keyword is to put the system in a familiar state before the user starts interacting with the system. However, you can omit writing user interactions in Given steps if Given in the "Precondition" step.

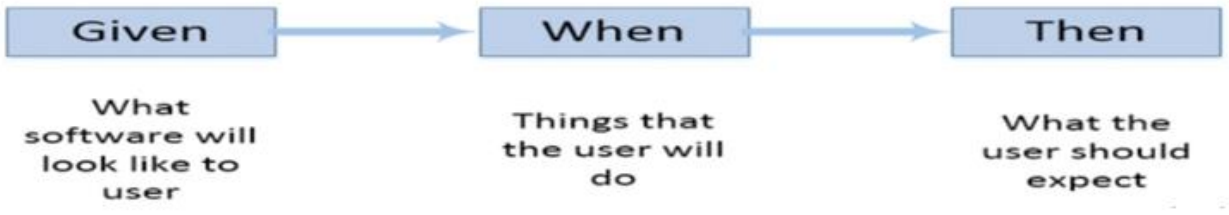
Syntax: Given - a test step that defines the 'context'
Given I am on "/."

When: When the step is to define action performed by the user.
Syntax: When - a test step that defines the 'action' performed
When I perform "Sign In."

Then: The use of 'then' keyword is to see the **outcome** after the action in when step. However, you can only verify noticeable changes.
Syntax: Then - test step that defines the 'outcome."
Then I should see "Welcome Tom."

And & But: You may have multiple given when or Then.
Syntax: But - additional test step which defines the 'action' 'outcome."
But I should see "Welcome Tom.
And - additional test step that defines the 'action' performed
And I write "EmailAddress" with Tomjohn@gmail.com.

Given, When, Then, and, but are test steps. You can use them interchangeably. The interpreter doesn't display any error. However, they will surely not make any 'sense' when read.


Given
When
Then
What software will look like to user
Things that the user will do
What the user should expect

<u>Gherkin Example 1:</u>

Feature: Login functionality of social networking site Facebook.

Given I am a facebook user.
When I enter username as username.
And I enter the password as the password
Then I should be redirected to the home page of facebook

<u>Gherkin Example 2:</u>

Feature: User Authentication

Background:
Given the user is already registered to the website
Scenario:
Given the user is on the login page
When the user inputs the correct email address
And the user inputs the correct password
And the user clicks the Login button
Then the user should be authenticated
And the user should be redirected to their dashboard
And the user should be presented with a success message

BDD (Behavioral Driven Development) is a **software development** approach that was developedfrom **Test Driven Development (TDD)**.

*BDD includes test case development on the basis of the behavior of software functionalities. All test cases are written in the form of simple English statements inside a <u>feature file</u>, which is human- generated. Acceptance test case statements are entirely focused on user actions. BDD is written in simple English language statements, not in a typical programming language. BDD improves communication between technical and non-technical teams and stakeholders. In the following example, we are going to take the login function of a web application**.** In order to ensure the working of Login Functionality, we are developing acceptance test cases on thebasis of BDD.*
Feature: Login FunctionTo enter in the System User must be able to Access software when login is successful.

Scenario: Login
Given User has its Email
And Password
When User enters the correct Email and Password
Then It should be logged in
Scenario: Unsuccessful Login
When User enters either wrong Email or Password
Then It should be reverse back on the login page with an error message

Consider a scenario where you want to test Google Homepage. One of the test scenarios will be to verify that the page displays all the main elements. As part of a testcase, let us say that you want to check that the homepage displays the search text box,“Google Search” button and “I’m Feeling Lucky” button. With BDD, you can write this test scenario in the below format:

- **Given** I launch Chrome browser
- **When** I open Google Homepage
- **Then** I verify that the page displays search text box
- **And** the page displays Google Search button
- **And** the page displays I’m Feeling Lucky button

Why Cucumber?

There are multiple behaviour-driven development tools such as Cucumber, Concordion, SpecFlow, JDave etc, that let you write your test cases in the format given in the above. Cucumber is one of the most popular tools because of the reasons given below:

- Cucumber BDD is open source and hence, its free to use
- With Cucumber, you can write your test scripts in multiple languages such as Java,Ruby, .NET, Python etc
- Cucumber easily integrates with Selenium, Ruby on Rails, Watir and other web basedtesting tools
- Cucumber is one of the most widely used BDD tools. Due to this, you will find lots ofonline tutorials and forms to help you with your doubts and queries.

Basic Terms of Cucumber:

- Feature File
- Features
- Tags
- Scenario
- Gherkin Language
- Step Definition

Here are some of the best practices in Cucumber Testing:

- The versions of **Cucumber-java, Cucumber-junit,** and **Cucumber-core** jars should be the same for seamless connectivity.
- Adding an after hook to the code for capturing screenshots when the test fails can help diagnose the issue and debug it.
- Use Tags for organizing tests based on tag definition. This helps in cases where all tests don't have to be run every time. Such tests can be marked using tags and run only when required. This saves time and processing capacity of the system and resources.
- As always, it is important to run the Cucumber Selenium tests on real browsers and devices.

Scenario Outline: Scenario outlines are used when the same test has to be performed with different data set. We have to test login functionality with multiple different setsof username and password.

Feature: Login Functionality Feature
In order to ensure Login Functionality works,
I want to run the cucumber test to verify it is working

Scenario Outline: Login Functionality
Given user navigates to SOFTWARETESTINGHELP.COM
When user logs in using Username as <**username**> and Password <**password**>
Then login should be successful

```
Examples:
|username      |password     |
|Tom           |password1    |
|Harry          |password2    |
|Jerry          |password3    |
```

- As shown in above example column names are passed as a parameterto **When** statement.
- In place of Scenario, you have to use Scenario Outline.
- Examples are used to pass different arguments in the tabular format. Vertical pipes are used to separate two different columns. An example can contain manydifferent columns.

***Tags**:* Cucumber by default runs all scenarios in all the feature files. In real time projects, therecould be hundreds of feature file which are not required to run at all times.

- **For instance**: Feature files related to smoke test need not run all the time. So if you mentiona tag as smokeless in each feature file which is related to smoke test and runs cucumber testwith @SmokeTest tag. Cucumber will run only those feature files specific to given tags. Pleasefollow the below example. You can specify multiple tags in one feature file.
- Example of use of single tags:
  @SmokeTest
  Feature: Login Functionality Feature
  In order to ensure Login Functionality works,
  I want to run the cucumber test to verify it is working

Jenkins Overview:

Jenkins is an open-source DevOps tool that has been popularly used for continuous integration and continuous delivery processes. It is a Java-based application and platform-independent. It is a build tool; used for running builds from the source code repository, running unit tests, and sending the buildreports to the respective member or team. This CI server runs in servlet containers such as Apache Tomcat. Jenkins facilitates continuous integration and continuous delivery in software projects by automating parts related to build, test, anddeployment. This makes it easy for developers to continuously work on the betterment of the productby integrating changes to the project.

Jenkins automates the software builds in a continuous manner and lets the developers know about theerrors at an early stage. A strong Jenkins community is one of the prime reasons for its popularity. Jenkins is not only extensible but also has a thriving plugin ecosystem.Some of the possible steps that can be performed using Jenkins are:

- Software build using build systems such as Gradle, Maven, and more.
- Automation testing using test frameworks such as Nose2, PyTest, Robot, Selenium, and more.
- Execute test scripts (using Windows terminal, Linux shell, etc.
- Achieve test results and perform post actions such as printing test reports, and more.
- Execute test scenarios against different input combinations for obtaining improved test coverage.
- Continuous Integration (CI) where the artifacts are automatically created and tested. This aids in identification of issues in the product at an early stage of development.

Advantages of using Jenkins are:

- It is a cross-platform and can be used on Windows, Linux, Mac OS, and Solarisenvironments
- It is a free and open source tool
- Widely used and well documented
- Integration with a wide variety of tool and technologies
- Change Support: Jenkins generates the list of all changes done in repositories likeSVN.
- Permanent links: Jenkins provides direct links to the latest build or failed buildthat can be used for easy communication
- Installation: Jenkins is easy to install either using direct installation file (exe) orwar file to deploy using application server.
- Email integration: Jenkins can be configured to email the content of the status ofthe build.
- Easy Configuration: To configure various tasks on Jenkins is easy.
- TestNG test: Jenkins can be configured to run the automation test buildon Testng after each build of SVN.
- Multiple VMs: Jenkins can be configured to distribute

the build on multiplemachines.

- Project build: Jenkins documents the details of jar, version of jar and mapping ofbuild and jar numbers.
- Plugins: 3rd party plugin can be configured in Jenkins to use features andadditional functionality.

Apart from Jenkins, we have many more tools in the market such as:

- Anthill
- Bamboo
- Cruise Control
- Team City and many more.

Why Jenkins and Selenium?

- Running Selenium tests in Jenkins allows you to run your tests every time your softwarechanges and deploy the software to a new environment when the tests pass.
- Jenkins can schedule your tests to run at specific time.
- You can save the execution history and Test Reports.
- Jenkins supports Maven for building and Testing a project in continuous integration.

TestNG Interview Questions

Q: What is TestNG?

A: TestNG is an open source automated testing framework; where NG of TestNG means Next Generation. TestNG is similar to JUnit (especially JUnit 4), but its not a JUnit extension. Its inspired by JUnit. It is designedto be better than JUnit, especially when testing integrated classes.

Q: What are the features of TestNG?

A: Features of TestNG are:

- Annotations.
- TestNG uses more Java and OO features.
- Supports testing integrated classes (e.g., by default, no need to create a new test class instance for everytest method).
- Separate compile-time test code from run-time configuration/data info.
- Flexible runtime configuration.
- Introduces „test groups‟. Once you have compiled your tests, you can just ask TestNG to run all the"front-end" tests, or "fast", "slow", "database", etc...

- Supports Dependent test methods, parallel testing, load testing, partial failure.
- Flexible plug-in API.
- Support for multi threaded testing.

Q: What are the advantages of TestNG over Junit?

A: Advantages of TestNG over Junit are:

- In Junit we have to declare @BeforeClass and @AfterClass which is a constraint where as in TestNGthere is no constraint like this.
- Additional Levels of setUp/tearDown level are available in TestNG like @Before/AfterSuite,@Before/AfterTest and @Before/AfterGroup
- No Need to extend any class in TestNG.
- There is no method name constraint in TestNG as in Junit. You can give any name to the test methods inTestNG.
- In TestNG we can tell the test that one method is dependent on another method where as in Junit this isnot possible. In Junit each test is independent of another test.
- Grouping of testcases is available in TestNG where as the same is not available in Junit.
- Execution can be done based on Groups. For ex. If you have defined many cases and segregated them bydefining 2 groups as Sanity and Regression. Then if you only want to execute the “Sanity” cases then just tell TestNG to execute the “Sanity” and TestNG will automatically execute the cases belonging tothe “Sanity” group

Q: What are the basic steps involved in writing TestNG tests?

A: Writing a test in TestNG basically involves following steps:

Write the business logic of your test and insert TestNG annotations in your code. Add the information about your test (e.g. the class name, the groups you wish to run, etc...) in atestng.xml file or in build.xml. Run TestNG.

Q**: Give examples of some of the annotations supported by TestNG.**

A: TestNG supports the following annotations:

@BeforeSuite, @AfterSuite, @BeforeClass, @AfterClass, @BeforeTest, @AfterTest, @BeforeGroups, @AfterGroups, @BeforeMethod, @AfterMethod, @DataProvider, @Factory, @Listeners, @Parameters,@Test.

Q: What are the benefits of using annotations?
A: Following are some of the benefits of using annotations:

- TestNG identifies the methods it is interested in by looking up annotations. Hence, method names arenot restricted to any pattern or format.
- We can pass additional parameters to annotations.
- Annotations are strongly typed, so the compiler will flag any mistakes right away.
- Test classes no longer need to extend anything (such as TestCase, for JUnit 3).

Q: What are the different ways in which TestNG can be invoked?
A: You can invoke TestNG in several different ways:

- Using Eclipse
- With ant
- From the command line
- Using IntelliJ's IDEA

Q: Give a an example to invoke TestNG from command line.

A: Assuming that you have TestNG in your class path, the simplest way to invoke TestNG is as follows:

```
java org.testng.TestNG testng1.xml [testng2.xml testng3.xml ...]
```

Q: What is testng.xml file used for?
A: File *testng.xml* captures your entire testing in XML. This file makes it easy to describe all your test suites and their parameters in one file, which you can check in your code repository or e-mail to coworkers. It also makes it easy to extract subsets of your tests or split several runtime configurations (e.g., testng-database.xmlwould run only tests that exercise your database).

Q: What is test suite?
A: A Test suite is a collection of test cases that are intended to test a behavior or set of behaviors of softwareprogram. In TestNG, we cannot define a suite in testing source code, but it is represented by one XML file as suite is the feature of execution. This also allows flexible configuration of the *tests* to be run. A suite cancontain one or more tests and is defined by the <suite> tag. <suite> is

a root tag of your testng.xml. It describes a test suite, which in turn is made of several <test> sections.

Q: How can you disable a test in TestNG?
A: Annotation *@Test(enabled = false)* helps to disable the test case which you want to ignore.
Q: What is group test?
A: Group tests permits you dispatch methods into proper portions and preform sophisticated groupings of testmethods. Not only can you declare those methods that belong to groups, but you can also specify groups that contain other groups. Then, TestNG can be invoked and asked to include a certain set of groups (or regular expressions) while excluding another set. This gives you maximum flexibility in how you partition your testsand doesn't require you to recompile anything if you want to run two different sets of tests back to back.

Q: How to you specify a group in testng.xml?
A: 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.

Q: What is exception test?
A: TestNG provides a option of tracing the Exception handling of code. You can test whether a code throws desired exception or not. The **expectedExceptions** parameter is used along with @Test annotation. Now, let'ssee *@Test(expectedExceptions)* in action.

Q: What is dependency test?
A: Sometimes, you may need to invoke methods in a Test case in a particular order or you want to share some data and state between methods. This kind of dependency is supported by TestNG as it supports the declarationof explicit dependencies between test methods. TestNG allows you to specify dependencies either with:

- Using attributes *dependsOnMethods* in @Test annotations **OR**
- Using attributes *dependsOnGroups* in @Test annotations.

Q: What is difference between dependsOn Groups and dependsOnMethods?

➢ On using groups, we are no longer exposed to refactoring problems. As long as we don‟t modify the dependsOnGroups or groups attributes, our tests will keep running with the proper dependencies set up.

- Whenever a new method needs to be added in the dependency graph, all we need to do is put it in the right group and make sure it depends on the correct group. We don't need to modify any other method.

Q: What is parametric testing?

A: In most cases, you'll come across a scenario where the business logic requires a hugely varying number oftests. *Parameterized tests* allow developers to run the same test over and over again using different values. TestNG lets you pass parameters directly to your test methods in two different ways: With testng.xml With Data Providers

Q: How do you pass parameters with testng.xml?

A: We define the simple parameters in the testng.xml file and then reference those parameters in source files.

Q: What does it mean to pass parameters using dataproviders?

A: When you need to pass complex parameters or parameters that need to be created from Java (complex objects, objects read from a property file or a database, etc...), in such cases parameters can be passed usingDataproviders. A Data Provider is a method annotated with *@DataProvider*. This annotation has only one string attribute: its name. If the name is not supplied, the Data Provider‟s name automatically defaults to themethod‟s name. A Data Provider returns an array of objects.

Q: How can you run the JUnit tests using TestNG?

A: Put JUnit library on the TestNG classpath, so it can find and use JUnit classes, change your test runner fromJUnit to TestNG in Ant and then run TestNG in "mixed" mode. This way you can have all your tests in the same project, even in the same package, and start using TestNG. This approach also allows you to convert yourexisting JUnit tests to TestNG incrementally. Also define property *junit="true"* in the <test> tag of the testng.xml.

Q: What are different ways in which you can generate the reports of TestNg results?

A: There are two main ways to generate a report with TestNG:

Listeners : For implementing a listener class, the class has to implement the *org.testng.ITestListener* interface. These classes are notified at runtime by TestNG when the test starts, finishes, fails, skips, orpasses.

Reporters : For implementing a reporting class, the class has to implement an *org.testng.IReporter* interface. These classes are called when the whole suite run ends. The object containing the informationof the whole test run is passed to this class when called.

Java Interview Questions

Q: What is the most important feature of Java?
A: Java is a platform independent language.

Q: What do you mean by platform independence?
A: Platform independence means that we can write and compile the java code in one platform (eg Windows)and can execute the class in any other supported platform eg (Linux,Solaris,etc).
Q: Are JVM's platform independent?
A: JVM's are not platform independent. JVM's are platform specific run time implementation provided by thevendor.

Q:What is a JVM?
A: JVM is Java Virtual Machine which is a run time environment for the compiled java class files.

Q: What is the difference between a JDK and a JVM?
A: JDK is Java Development Kit which is for development purpose and it includes execution environmentalso. But JVM is purely a run time environment and hence you will not be able to compile your source files using a JVM.

Q: What is the base class of all classes?
A: java.lang.Object

Q: Does Java support multiple inheritance?
A: Java doesn't support multiple inheritance.

Q: Is Java a pure object oriented language?
A: Java uses primitive data types and hence is not a pure object oriented language.

Q: Are arrays primitive data types?
A: In Java, Arrays are objects.

Q: What is difference between Path and Classpath?
A: Path and Classpath are operating system level environment variales. Path is used define where the systemcan find the executables(.exe) files and classpath is used to specify the location .class files.

Q: What are local variables? A: Local varaiables are those which are declared within a block of code like methods. Local variables shouldbe initialised before accessing them.

Q: What are instance variables?
A: Instance variables are those which are defined at the class level. Instance variables need not be initializedbefore using them as they are automatically initialized to their default values.

Q: How to define a constant variable in Java?
A: The variable should be declared as static and final. So only one copy of the variable exists for all instancesof the class and the value can't be changed also. static final int PI = 2.14; is an example for constant.

Q: Should a main method be compulsorily declared in all java classes?
A: No not required. main method should be defined only if the source class is a java application.

Q: What is the return type of the main method?
A: Main method doesn't return anything hence declared void.

Q: Why is the main method declared static?
A: main method is called by the JVM even before the instantiation of the class hence it is declared as static.

Q: What is the argument of main method?
A: main method accepts an array of String object as arguement.

Q: Can a main method be overloaded?
A: Yes. You can have any number of main methods with different method signature and implementation inthe class.

Q: Can a main method be declared final?
A: Yes. Any inheriting class will not be able to have it's own default main method.

Q: Does the order of public and static declaration matter in main method?
A: No it doesn't matter but void should always come before main().

Q: Can a source file contain more than one Class declaration?
A: Yes a single source file can contain any number of Class declarations but only one of the class can bedeclared as public.

Q: What is a package?
A: Package is a collection of related classes and interfaces. package declaration should be first statement in ajava class.

Q: Which package is imported by default?
A: java.lang package is imported by default even without a package declaration.

Q: Can a class declared as private be accessed outside it's package?
A: Not possible.

Q: Can a class be declared as protected?
A: A class can't be declared as protected. only methods can be declared as protected.

Q: What is the access scope of a protected method?
A: A protected method can be accessed by the classes within the same package or by the subclasses of theclass in any package.

Q: What is the purpose of declaring a variable as final?
A: A final variable's value can't be changed. final variables should be initialized before using them.

Q: What is the impact of declaring a method as final?
A: A method declared as final can't be overridden. A sub-class can't have the same method signature with adifferent implementation.

Q: I don't want my class to be inherited by any other class. What should I do?
A: You should declared your class as final. But you can't define your class as

final, if it is an abstract class. Aclass declared as final can't be extended by any other class.

Q: Can you give few examples of final classes defined in Java API?
A: java.lang.String,java.lang.Math are final classes.

Q: How is final different from finally and finalize?
A: final is a modifier which can be applied to a class or a method or a variable. final class can't be inherited,final method can't be overridden and final variable can't be changed.

finally is an exception handling code section which gets executed whether an exception is raised or not bythe try block code segment.

finalize() is a method of Object class which will be executed by the JVM just before garbage collectingobject to give a final chance for resource releasing activity.

Q: Can a class be declared as static?
A: No a class cannot be defined as static. Only a method, a variable or a block of code can be declared asstatic.

Q: When will you define a method as static?
A: When a method needs to be accessed even before the creation of the object of the class then we shoulddeclare the method as static.

Q: What are the restriction imposed on a static method or a static block of code?
A: A static method should not refer to instance variables without creating an instance and cannot use "this"operator to refer the instance.

Q: I want to print "Hello" even before main is executed. How will you acheive that?
A: Print the statement inside a static block of code. Static blocks get executed when the class gets loaded intothe memory and even before the creation of an object. Hence it will be executed before the main method. And it will be executed only once.

Q: What is the importance of static variable?
A: static variables are class level variables where all objects of the class refer to

the same variable. If oneobject changes the value then the change gets reflected in all the objects.

Q: Can we declare a static variable inside a method?
A: Static varaibles are class level variables and they can't be declared inside a method. If declared, the classwill not compile.

Q: What is an Abstract Class and what is it's purpose?
A: A Class which doesn't provide complete implementation is defined as an abstract class. Abstract classesenforce abstraction.

Q: Can a abstract class be declared final?
A: Not possible. An abstract class without being inherited is of no use and hence will result in compile timeerror.

Q: What is use of a abstract variable?
A: Variables can't be declared as abstract. only classes and methods can be declared as abstract.

Q: Can you create an object of an abstract class?
A: Not possible. Abstract classes can't be instantiated.

Q: Can a abstract class be defined without any abstract methods?
A: Yes it's possible. This is basically to avoid instance creation of the class.

Q: Class C implements Interface I containing method m1 and m2 declarations. Class C has providedimplementation for method m2. Can i create an object of Class C?
A: No not possible. Class C should provide implementation for all the methods in the Interface I. Since ClassC didn't provide implementation for m1 method, it has to be declared as abstract. Abstract classes can't beinstantiated.

Q: Can a method inside a Interface be declared as final?
A: No not possible. Doing so will result in compilation error. public and abstract are the only applicablemodifiers for method declaration in an interface.

Q: Can an Interface implement another Interface?
A: Intefaces doesn't provide implementation hence a interface cannot implement another interface.

Q: Can an Interface extend another Interface?
A: Yes an Interface can inherit another Interface, for that matter an Interface can extend more than oneInterface.

Q: Can a Class extend more than one Class?
A: Not possible. A Class can extend only one class but can implement any number of Interfaces.

Q: Why is an Interface be able to extend more than one Interface but a Class can't extend more thanone Class?
A: Basically Java doesn't allow multiple inheritance, so a Class is restricted to extend only one Class. But anInterface is a pure abstraction model and doesn't have inheritance hierarchy like classes(do remember thatthe base class of all classes is Object). So an Interface is allowed to extend more than one Interface.

Q: Can an Interface be final?
A: Not possible. Doing so will result in compilation error.

Q: Can a class be defined inside an Interface?
A: Yes it's possible.

Q: Can an Interface be defined inside a class?
A: Yes it's possible.

Q: What is a Marker Interface?
A: An Interface which doesn't have any declaration inside but still enforces a mechanism.

Q: Which OO Concept is achieved by using overloading and overriding?
A: Polymorphism.

Q: If I only change the return type, does the method become overloaded?
A: No it doesn't. There should be a change in method arguements for a method to be overloaded.

Q: Why does Java not support operator overloading?
A: Operator overloading makes the code very difficult to read and maintain. To maintain code simplicity,Java doesn't support operator overloading.

Q: Can we define private and protected modifiers for variables in interfaces?
A: No

Q: What is Externalizable?
A: Externalizable is an Interface that extends Serializable Interface. And sends data into Streams in Compressed Format. It has two methods, writeExternal(ObjectOuput out) and readExternal(ObjectInputin)

Q: What modifiers are allowed for methods in an Interface?
A: Only public and abstract modifiers are allowed for methods in interfaces.

Q: What is a local, member and a class variable?
A: Variables declared within a method are "local" variables. Variables declared within the class i.e not withinany methods are "member" variables (global variables). Variables declared within the class i.e not within any methods and are defined as "static" are class variables

Q: What is an abstract method?
A: An abstract method is a method whose implementation is deferred to a subclass.

Q: What value does read() return when it has reached the end of a file?
A: The read() method returns -1 when it has reached the end of a file.

Q: Can a Byte object be cast to a double value?
A: No, an object cannot be cast to a primitive value.

Q: What is the difference between a static and a non-static inner class?
A: A non-static inner class may have object instances that are associated with instances of the class's outerclass. A static inner class does not have any object instances.

Q: What is an object's lock and which object's have locks?
A: An object's lock is a mechanism that is used by multiple threads to obtain synchronized access to the object. A thread may execute a synchronized method of an object only after it has acquired the object'slock. All objects and classes have locks. A class's lock is acquired on the class's Class object.

Q: What is the % operator?
A: It is referred to as the modulo or remainder operator. It returns the remainder of dividing the first operandby the second operand.

Q: When can an object reference be cast to an interface reference?
A: An object reference be cast to an interface reference when the object implements the referenced interface.

Q: Which class is extended by all other classes?
A: The Object class is extended by all other classes.

Q: Which non-Unicode letter characters may be used as the first character of an identifier?
A: The non-Unicode letter characters $ and _ may appear as the first character of an identifier

Q: What restrictions are placed on method overloading?
A: Two methods may not have the same name and argument list but different return types.

Q: What is casting?
A: There are two types of casting, casting between primitive numeric types and casting between object references. Casting between numeric types is used to convert larger values, such as double values, to smaller values, such as byte values. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.

Q: What is the return type of a program's main() method?
A: void.

Q: If a variable is declared as private, where may the variable be accessed?
A: A private variable may only be accessed within the class in which it is declared.

Q: What do you understand by private, protected and public?
A: These are accessibility modifiers. Private is the most restrictive, while public is the least restrictive. Thereis no real difference between protected and the default type (also known as package protected) within the context of the same

package, however the protected keyword allows visibility to a derived class in a different package.

Q: What is Downcasting ?
A: Downcasting is the casting from a general to a more specific type, i.e. casting down the hierarchy

Q: What modifiers may be used with an inner class that is a member of an outer class?
A: A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

Q: How many bits are used to represent Unicode, ASCII, UTF-16, and UTF-8 characters?
A: Unicode requires 16 bits and ASCII require 7 bits. Although the ASCII character set uses only 7 bits, it isusually represented as 8 bits. UTF-8 represents characters using 8, 16, and 18 bit patterns. UTF-16 uses 16-bit and larger bit patterns.

Q: What restrictions are placed on the location of a package statement within a source code file?
A:A package statement must appear as the first line in a source code file (excluding blank lines and comments).

Q: What is a native method?
A: A native method is a method that is implemented in a language other than Java.

Q: What are order of precedence and associativity, and how are they used?
A: Order of precedence determines the order in which operators are evaluated in expressions. Associatitydetermines whether an expression is evaluated left-to-right or right-to-left

Q: Can an anonymous class be declared as implementing an interface and extending a class?
A: An anonymous class may implement an interface or extend a superclass, but may not be declared to doboth.

Q: What is the range of the char type?
A: The range of the char type is 0 to 2^16 - 1.

Q: What is the range of the short type?
A: The range of the short type is -(2^15) to 2^15 - 1.

Q: Why isn't there operator overloading?
A: Because C++ has proven by example that operator overloading makes code almost impossible tomaintain.

Q: What does it mean that a method or field is "static"?
A: Static variables and methods are instantiated only once per class. In other words they are class variables, not instance variables. If you change the value of a static variable in a particular object, the value of that variable changes for all instances of that class. Static methods can be referenced with the name of the classrather than the name of a particular object of the class (though that works too). That's how library methodslike System.out.println() work. out is a static field in the java.lang.System class.

Q: Is null a keyword?
A: The null value is not a keyword.

Q: Which characters may be used as the second character of an identifier, but not as the first characterof an identifier?
A: The digits 0 through 9 may not be used as the first character of an identifier but they may be used after thefirst character of an identifier.

Q: If a class is declared without any access modifiers, where may the class be accessed?
A: A class that is declared without any access modifiers is said to have package access. This means that theclass can only be accessed by other classes and interfaces that are defined within the same package.

Q: Does a class inherit the constructors of its superclass?
A: A class does not inherit constructors from any of its superclasses.

Q: Name the eight primitive Java types.
A: The eight primitive types are byte, char, short, int, long, float, double, and boolean.

Q: What restrictions are placed on the values of each case of a switch statement?
A: During compilation, the values of each case of a switch statement must evaluate to a value that can bepromoted to an int value.

Q: What is the difference between a while statement and a do statement?
A: A while statement checks at the beginning of a loop to see whether the next loop iteration should occur. Ado statement checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the body of a loop at least once.

Q: What modifiers can be used with a local inner class?
A: A local inner class may be final or abstract.

Q: When does the compiler supply a default constructor for a class?
A: The compiler supplies a default constructor for a class if no other constructors are provided.

Q: If a method is declared as protected, where may the method be accessed?
A: A protected method may only be accessed by classes or interfaces of the same package or by subclasses ofthe class in which it is declared.

Q: What are the legal operands of the instance of operator?
A: The left operand is an object reference or null value and the right operand is a class, interface, or arraytype.

Q: Are true and false keywords?
A: The values true and false are not keywords.

Q: What happens when you add a double value to a String?
A: The result is a String object.

Q: What is the diffrence between inner class and nested class?
A: When a class is defined within a scope od another class, then it becomes inner class. If the access modifierof the inner class is static, then it becomes nested class.

Q: Can an abstract class be final?
A: An abstract class may not be declared as final

Q: What is numeric promotion?
A: Numeric promotion is the conversion of a smaller numeric type to a larger numeric type, so that integer and floating-point operations may take place. In numerical promotion, byte, char, and short values are converted to int values. The int values are also converted to long values, if necessary. The long and floatvalues are converted to double values, as required

Q: What is the difference between a public and a non-public class?
A: A public class may be accessed outside of its package. A non-public class may not be accessed outside ofits package.

Q: To what value is a variable of the boolean type automatically initialized?
A: The default value of the boolean type is false

Q: What is the difference between the prefix and postfix forms of the ++ operator?
A: The prefix form performs the increment operation and returns the value of the increment operation. The postfix form returns the current value all of the expression and then performs the increment operation onthat value.

Q: What restrictions are placed on method overriding?
A: Overridden methods must have the same name, argument list, and return type. The overriding method maynot limit the access of the method it overrides. The overriding method may not throw any exceptions that may not be thrown by the overridden method.

Q: What is a Java package and how is it used?
A: A Java package is a naming context for classes and interfaces. A package is used to create a separate namespace for groups of classes and interfaces. Packages are also used to organize related classes and interfacesinto a single API unit and to control accessibility to these classes and interfaces.

Q: What modifiers may be used with a top-level class?
A: A top-level class may be public, abstract, or final.

Q: What is the difference between an if statement and a switch statement?
A: The if statement is used to select among two alternatives. It uses a boolean expression to decide which alternative should be executed. The switch

statement is used to select among multiple alternatives. It usesan int expression to determine which alternative should be executed.

Q: What are the practical benefits, if any, of importing a specific class rather than an entire package(e.g. import java.net.* versus import java.net.Socket)?
A: It makes no difference in the generated class files since only the classes that are actually used are referenced by the generated class file. There is another practical benefit to importing single classes, and this arises when two (or more) packages have classes with the same name. Take java.util.Timer and javax.swing.Timer, for example. If I import java.util.* and javax.swing.* and then try to use "Timer", I getan error while compiling (the class name is ambiguous between both packages). Let's say what you really wanted was the javax.swing.Timer class, and the only classes you plan on using in java.util are Collection and HashMap. In this case, some people will prefer to import java.util.Collection and import java.util.HashMap instead of importing java.util.*. This will now allow them to use Timer, Collection, HashMap, and other javax.swing classes without using fully qualified class names in.

Q: Can a method be overloaded based on different return type but same argument type ?
A: No, because the methods can be called without using their return type in which case there is ambiquity forthe compiler.

Q: What happens to a static var that is defined within a method of a class ?
A: Can't do it. You'll get a compilation error

Q: How many static init can you have ?
A: As many as you want, but the static initializers and class variable initializers are executed in textual order and may not refer to class variables declared in the class whose declarations appear textually after the use,even though these class variables are in scope.

Q: What is the difference between method overriding and overloading?
A: Overriding is a method with the same name and arguments as in a parent, whereas overloading is the samemethod name but different arguments

Q: What is constructor chaining and how is it achieved in Java ?
A: A child object constructor always first needs to construct its parent (which in

turn calls its parent constructor.). In Java it is done via an implicit call to the no-args constructor as the first statement.

Q: What is the difference between the Boolean & operator and the && operator?
A: If an expression involving the Boolean & operator is evaluated, both operands are evaluated. Then the & operator is applied to the operand. When an expression involving the && operator is evaluated, the first operand is evaluated. If the first operand returns a value of true then the second operand is evaluated. The&& operator is then applied to the first and second operands. If the first operand evaluates to false, the evaluation of the second operand is skipped.

Q: Which Java operator is right associative?
A: The = operator is right associative.

Q: Can a double value be cast to a byte?
A: Yes, a double value can be cast to a byte.

Q: What is the difference between a break statement and a continue statement?
A: A break statement results in the termination of the statement to which it applies (switch, for, do, or while).A continue statement is used to end the current loop iteration and return control to the loop statement.

Q: Can a for statement loop indefinitely?
A: Yes, a for statement can loop indefinitely. For example, consider the following: for(;;) ;

Q: To what value is a variable of the String type automatically initialized?
A: The default value of an String type is null.

Q: What is the difference between a field variable and a local variable?
A: A field variable is a variable that is declared as a member of a class. A local variable is a variable that isdeclared local to a method.

Q: How are this() and super() used with constructors?
A: this() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor.

Q: What does it mean that a class or member is final?
A: A final class cannot be inherited. A final method cannot be overridden in a subclass. A final field cannotbe changed after it's initialized, and it must include an initializer statement where it's declared.

Q: What does it mean that a method or class is abstract?
A: An abstract class cannot be instantiated. Abstract methods may only be included in abstract classes. However, an abstract class is not required to have any abstract methods, though most of them do. Eachsubclass of an abstract class must override the abstract methods of its superclasses or it also should be declared abstract.

Q: What is a transient variable?
A: transient variable is a variable that may not be serialized.

Q: How does Java handle integer overflows and underflows?
A: It uses those low order bytes of the result that can fit into the size of the type allowed by the operation.

Q: What is the difference between the >> and >>> operators?
A: The >> operator carries the sign bit when shifting right. The >>> zero-fills bits that have been shifted out.

Q: Is sizeof a keyword?
A: The sizeof operator is not a keyword.

Some Additional Question for Java

What is OOPs?
Ans: Object oriented programming organizes a program around its data,i.e.,objects and aset of well defined interfaces to that data.An object-oriented program can be characterized as data controlling access to code.

What is the difference between Procedural and OOPs?
Ans: a) In procedural program, programming logic follows certain procedures and the instructions are executed one after another. In OOPs program, unit of program is object,which is nothing but combination of data and code.
b) In procedural program,data is exposed to the whole program whereas in

OOPs program,it is accessible with in the object and which in turn assures the security of thecode.

What are Encapsulation, Inheritance and Polymorphism?
Ans: Encapsulation is the mechanism that binds together code and data it manipulatesand keeps both safe from outside interference and misuse.

Inheritance is the process by which one object acquires the properties of another object.

Polymorphism is the feature that allows one interface to be used for general classactions.

What is the difference between Assignment and Initialization?
Ans: Assignment can be done as many times as desired whereas initialization can be doneonly once.

What are Class, Constructor and Primitive data types?
Ans: Class is a template for multiple objects with similar features and it is a blue print forobjects. It defines a type of object according to the data the object can hold and the operations the object can perform.

Constructor is a special kind of method that determines how an object is initializedwhen created.

Primitive data types are 8 types and they are: byte, short, int, longfloat, double Boolean, char

What is the difference between constructor and method?
Ans: Constructor will be automatically invoked when an object is created whereasmethod has to be called explicitly.

What are methods and how are they defined?
Ans: Methods are functions that operate on instances of classes in which they are defined.Objects can communicate with each other using methods and can call methods in other classes. Method definition has four parts. They are name of the method, type of object or primitive type the method returns, a list of parameters and the body of the method. A method's signature is acombination of the first three parts mentioned above.

What is the use of bin and lib in JDK?
Ans: Bin contains all tools such as javac, appletviewer, awt tool, etc., whereas libcontains API and all packages.

What is casting?
Ans: Casting is used to convert the value of one type to another.

How many ways can an argument be passed to a subroutine and explain them?
Ans: An argument can be passed in two ways. They are passing by value and passing byreference.

Passing by value: This method copies the value of an argument into the formal parameter of the subroutine.

Passing by reference: In this method, a reference to an argument (not the value ofthe argument) is passed to the parameter.

What is the difference between an argument and a parameter?
Ans: While defining method, variables passed in the method are called parameters. Whileusing those methods, values passed to those variables are called arguments.

What are different types of access modifiers?
Ans: public: Any thing declared as public can be accessed from anywhere.
private: Any thing declared as private can't be seen outside of its class.
protected: Any thing declared as protected can be accessed by classes in the samepackage and subclasses in the other packages.
default modifier : Can be accessed only to classes in the same package.

What is final, finalize() and finally?
Ans: final : final keyword can be used for class, method and variables.
A final class cannot be subclassed and it prevents other programmers from subclassing a secure class to invoke insecure methods.
A final method can' t be overridden
A final variable can't change from its initialized value.
finalize( ) : finalize( ) method is used just before an object is destroyed and can becalled just prior to garbage collection.
finally : finally, a key word used in exception handling, creates a block of code

thatwill be executed after a try/catch block has completed and before the code following the try/catch block.The finally block will execute whether or not an exception is thrown.

for example, if a method opens a file upon exit, then you will not want the code that closes the file to be bypassed by the exception-handling mechanism. This finally keyword is designed to address this contingency

What is Selenium 2.0? I have heard this buzz word many times.
Ans: Selenium 2.0 is a consolidation of two web testing tools:
Selenium RC and Web Driver

Why are two tools being combined as Selenium 2.0?
Ans: Selenium 2.0 promises to give much cleaner API than Selenium RC and at the same time not beingrestricted by JavaScript Security restriction like same origin policy.

So everyone is going to use Selenium 2.0?
Ans: Well no, for example if you are using Selenium Perl client driver than there is no similar offering fromSelenium 2.0 and you would have to stick to Selenium 1.0 till there is similar library available for Selenium 2.0

So how do I specify my browser configurations with Selenium 2.0?
Ans: Selenium 2.0 offers following browser/mobile configuration: Android Driver, Chrome Driver, Firefox Driver, HtmlUnitDriver, InternetExplorerDriver, IPhoneDriver, IPhoneSimulatorDriver, RemoteWebDriver.

And all of them have been implemented from interface WebDriver. To be able to use any of these drivers, youneed to instantiate their corresponding classes.

How is Selenium 2.0 configuration different from Selenium 1.0?
Ans: In case of Selenium 1.0, you need Selenium jar file pertaining to one library; for example, in case ofJava you need Java client driver and also Selenium server jar file. With Selenium 2.0, you need language Binding (i.e. Java, C#, etc) and Selenium server jar if you are using Remote Control or Remote Web Driver.

Selenium Benefits over Web Driver:

- Selenium supports many browsers and many languages, Web Driver needs native implementations for each newlanguage/browser combo.
- Very mature and complete API
- Currently (Sept 2010) supports JavaScript alerts and confirms better
- Benefits of Web Driver Compared to Selenium
- Native automation faster and a little less prone to error and browser configuration
- Does not Requires Selenium-RC Server to be running
- Access to headlessHTMLUnit can allow really fast tests
- Great API

What is Web Driver?
Ans: Web Driver uses a different underlying framework from Selenium‟s javascript Selenium-Core. It also providesan alternative API with functionality not supported in Selenium-RC. WebDriver does not depend on a JavaScript core embedded within the browser; therefore it is able to avoid some long-running Selenium limitations.

- WebDriver‟s goal is to provide an API that establishes
- A well-designed standard programming interface for web-app testing.
- Improved consistency between browsers.
- Additional functionality addressing testing problems not well-supported in Selenium 1.0.

The Selenium developers strive to continuously improve Selenium. Integrating WebDriver is another step in that process. The developers of Selenium and of WebDriver felt they could make significant gains for the OpenSource test automation community be combining forces and merging their ideas and technologies. Integrating WebDriver into Selenium is the current result of those efforts.

When to Use WebDriver?
Ans: One should use WebDriver when requiring improved support for Multi-browser testing including improved functionality for browsers not well-supported by Selenium-1.0. Handling multiple frames, multiple browser windows, popups, and alerts. Page navigation. Drag-and-drop. AJAX-based UI elements.

Advantages of web driver compared to selenium:

- Support for iPhone and Android testing
- Implementation of listeners - a much awaited feature
- Better features for Ajax testing.
- You can easily simulate clicking on front and back button of browser.
- You can extract objects in bulk like QTP. For ex - extract all links of page. With RC this was a big hastle
- Unlike RC you don‟t have to start a server in webdriver.
- You can simulate movement of a mouse using selenium.
- Tabs and pops are more or less the same. RC can also handle and Webdriver can also handle.
- You can find coordinates of any object using Webdriver.
- You have classes in Webdriver which help you to simulate key press events of keyboard.
- Keyword driven framework is very easy to build in webdriver.

Which browsers does WebDriver support?

Ans: The existing drivers are the ChromeDriver, InternetExplorerDriver, FirefoxDriver, OperaDriver and HtmlUnitDriver. For more information about each of these, including their relative strengths and weaknesses, please follow the links to the relevant pages. There is also support for mobile testing via the AndroidDriver andIPhoneDriver.

Selenium Interview Questions

What is Selenium?

Ans. Selenium is a set of tools that supports rapid development of test automation scripts for webbased applications. Selenium testing tools provides a rich set of testing functions specifically designed to fulfil needs of testing of a web based application.

What are the main components of Selenium testing tools?

Ans. Selenium IDE, Selenium RC and Selenium Grid

What is Selenium IDE?

Ans. Selenium IDE is for building Selenium test cases. It operates as a Mozilla Firefox add on andprovides an easy to use interface for developing and running individual test cases or entire test suites. Selenium-IDE has a recording feature, which will keep account of user actions as they are performed and store them as a reusable script to play back.

What is the use of context menu in Selenium IDE?
Ans. It allows the user to pick from a list of assertions and verifications for the selected location.

Can tests recorded using Selenium IDE be run in other browsers?
Ans. Yes. Although Selenium IDE is a Firefox add on, however, tests created in it can also be run in other browsers by using Selenium RC (Selenium Remote Control) and specifying the name of the testsuite in command line.

What are the advantage and features of Selenium IDE?

- Intelligent field selection will use IDs, names, or XPath as needed
- It is a record & playback tool and the script format can be written in various languages including C#, Java, PERL, Python, PHP, HTML
- Auto complete for all common Selenium commands
- Debug and set breakpoints
- Option to automatically assert the title of every page
- Support for Selenium user-extensions.js file

What are the disadvantage of Selenium IDE tool?
Ans: 1. Selenium IDE tool can only be used in Mozilla Firefox browser.
2. It is not playing multiple windows when we record it.

What is Selenium RC (Remote Control)?
Ans. Selenium RC allows the test automation expert to use a programming language for maximum flexibility and extensibility in developing test logic. For example, if the application under test returns a result set and the automated test program needs to run tests on each element in the result set, the iteration / loop support of programming language's can be used to iterate through the result set, calling Selenium commands to run tests on each item.

Selenium RC provides an API and library for each of its supported languages. This ability to use Selenium RC with a high level programming language to develop test cases also allows the automated testing to be integrated with the project's automated build environment.

What is Selenium Grid?
Ans. Selenium Grid in the selenium testing suit allows the Selenium RC solution to scale for test suitesthat must be run in multiple environments. Selenium Grid can be used to run multiple instances of Selenium RC on various operating system and browser configurations.

How Selenium Grid works?
Ans. Selenium Grid sent the tests to the hub. Then tests are redirected to an available Selenium RC, which launch the browser and run the test. Thus, it allows for running tests in parallel with the entiretest suite.

What you say about the flexibility of Selenium test suite?
Ans. Selenium testing suite is highly flexible. There are multiple ways to add functionality to Seleniumframework to customize test automation. As compared to other test automation tools, it is Selenium's strongest characteristic. Selenium Remote Control support for multiple programming andscripting languages allows the test automation engineer to build any logic they need into their automated testing and to use a preferred programming or scripting language of one's choice.

Also, the Selenium testing suite is an open source project where code can be modified andenhancements can be submitted for contribution.

What test can Selenium do?
Ans. Selenium is basically used for the functional testing of web based applications. It can be used fortesting in the continuous integration environment. It is also useful for agile testing

What is the cost of Selenium test suite?
Ans. Selenium test suite a set of open source software tool, it is free of cost.

What browsers are supported by Selenium Remote Control?
Ans. The test automation expert can use Firefox, IE 7/8, Safari and Opera browsers to run tests inSelenium Remote Control.

What programming languages can you use in Selenium RC?
Ans. C#, Java, Perl, PHP, Python, Ruby

What are the advantages and disadvantages of using Selenium as testing tool?
Ans. Advantages: Free, Simple and powerful DOM (document object model) level testing, can be usedfor continuous integration; great fit with Agile projects.

Disadvantages: Tricky setup; dreary errors diagnosis; can not test client server applications.

What is difference between QTP and Selenium?
Ans. Only web applications can be testing using Selenium testing suite. However, QTP can be used fortesting client server applications. Selenium supports following web browsers: Internet Explorer, Firefox, Safari, Opera or Konqueror on Windows, Mac OS X and Linux. However, QTP is limited to Internet Explorer on Windows.

QTP uses scripting language implemented on top of VB Script. However, Selenium test suite has theflexibility to use many languages like Java, .Net, Perl, PHP, Python, and Ruby.

Describe technical problems that you had with Selenium tool?
Ans: As with any other type of test automation tools like SilkTest, HP QTP, Watir, Canoo Webtest, Selenium allows to record, edit, and debug tests cases. However there are several problems that seriously affect maintainability of recorded test cases, occasionally Quality Assurance Engineers complain that it takes more time to maintain automated test cases thanto perform manual testing; however this is an issue with all automated testing tools and most likely related to improper testing framework design. Another problem is complex ID for an HTML element. If IDs is auto-generated, the recorder test cases may fail during playback. The work around is to use XPath to find required HTML element. Selenium supports AJAX without problems, but QA Tester should be aware that Selenium does not know when AJAX action is completed, soClickAndWait will not work. Instead QA tester could use pause, but the snowballing effect of several 'pause' commands would really slow down total testing time of test cases. The best solution would be to use waitForElement.

What test can Selenium do?
Ans: Selenium could be used for the functional, regression, load testing of the web based applications. The automation toolcould be implemented for post release validation with continuous integration tools like Jenkins, Hudson, QuickBuild or CruiseControl.

What is the price of Selenium license per server/ Selenium license cost per client machine?
Ans: Selenium is open source software, released under the Apache 2.0 license and can be downloaded and used withoutcharge.

What is the latest version of Selenium components?
Ans: The latest versions are Selenium IDE 1.3.0, Selenium Server (formerly the Selenium RC Server) 2.9.0, Selenium Client Drivers Java 2.9.0, Selenium Client Drivers C# 2.9.0, Selenium Client Drivers Ruby 2.8.0, Selenium Client Drivers Python2.9, Selenium Grid 1.0.8.

What does SIDE stand for?
Ans: Selenium IDE. It was a very tricky interview question.

What is Selenium Remote Control (RC) tool?
Ans: Selenium Remote Control (RC) is the powerful solution for test cases that need more than simple browser actions and linear execution. Selenium-RC allows the developing of complex test scenarios like reading and writing files, querying a database, and emailing test reports. These tasks can be achieved by tweaking test cases in your preferred programming language.

Can Selenium test an application on iPhone's Mobile Safari browser?
Selenium should be able to handle Mobile Safari browser. There is experimental Selenium IPhone Driver for running tests on Mobile Safari on the iPhone, iPad and iPod Touch.

Can Selenium test an application on Android browser?
Selenium should be able to handle Android browser. There is experimental Selenium Android Driver for running tests inAndroid browser.

What are the disadvantages of using Selenium as testing tool?
Selenium weak points are tricky setup; dreary errors diagnosis; tests only web applications

What programming language is best for writing Selenium tests?
The web applications may be written in Java, Ruby, PHP, Python or any other web framework. There are certain advantages for using the same language for writing test cases as application under test. For example, if the team already have the experience with Java, QA Tester could always get the piece of advice while mastering Selenium test cases in Java. Sometimes it is better to choose simpler programming language that will ultimately deliver better success. In this case QA testers can adopt easier programming languages, for example Ruby, much faster comparing with Java, and can become become experts as soon as possible.

Do you know any alternative test automation tools for Selenium?
Selenium appears to be the mainstream open source tool for browser side testing, but there are many alternatives. Canoo Webtest is a great Selenium alternative and it is probably the fastest automation tool. Another Selenium alternative is Watir, but in order to use Watir QA Tester has to learn Ruby. One more alternative to Selenium is Sahi, butis has confusing interface and small developers community.

What are the technical challenges with selenium?
As you know Selenium is a free ware open source testing tool. There are many challenges with Selenium.

- Selenium Supports only web based applications
- It doesn't support any non web based (Like Win 32, Java Applet, Java Swing, .Net Client Server etc) applications
- When you compare selenium with QTP, Silk Test, Test Partner and RFT, there are many challenges in terms ofmaintainability of the test cases
- Since Selenium is a freeware tool, there is no direct support if one is in trouble with the support of applications
- There is no object repository concept in Selenium, so maintainability of the objects is very high
- There are many challenges if one have to interact with Win 32 windows even when you are working with Web basedapplications
- Bitmap comparison is not supported by Selenium
- Any reporting related capabilities, you need to depend on third party tools
- You need to learn any one of the native language like (.Net, Java, Perl, Python, PHP, Ruby) to work efficiently with thescripting side of selenium

What are the test types supported by Selenium?
Selenium could be used for testing the web based applications. The test types can be supported are: functional, regression, load testing

The automation tool could be implemented for post release validation with continuous integration tools like: Jenkins, Hudson, QuickBuild CruiseCont

What are the capabilities of Selenium IDE?
Selenium IDE (Integrated Development Environment) works similar to commercial tools like QTP, Silk Test and TestPartner etc. The below mentioned points describes well about Selenium IDE.

- Selenium IDE is a Firefox add-on.
- Selenium IDE can support recording the clicks, typing, and other actions to make a test cases.
- Using Selenium IDE A Tester can play back the test cases in the Firefox browser
- Selenium IDE supports exporting the test cases and suites to Selenium RC.
- Debugging of the test cases with step-by-step can be done breakpoint insertion is possible
- Page abstraction functionality is supported by Selenium IDE
- Selenium IDE can supports an extensibility capability allowing the use of add-ons or user extensions that expand the functionality of Selenium IDE

How to execute a single line command from Selenium IDE?
Single line command from Selenium IDE can be executed in two ways:
Right click on the command in Selenium IDE and select "Execute This Command"

Select the command in Selenium IDE and press "X" key on the keyboard

How to export Selenium IDE Test Suite to Selenium RC Suites?

From selenium IDE the test suites can be exported into the languages as .Net, Java, Perl, Python, PHP, Ruby. The below mentioned steps can explain how to export the test suites: Open the test case from Selenium IDE then Select File -> Export Test Suite As

Why Selenium RC is used?
Selenium-IDE does not directly support: condition statements, iteration, logging and reporting of test results, error handling, particularly unexpected errors, database testing, test case grouping, re-execution of failed tests, test case dependency, capture screenshots on test failures.

The reason behind why Selenium-IDE does not support the above mentioned requirements is IDE supports only HTMLlanguage. Using HTML language we cannot achieve the above mentioned requirements. Because HTML does not support conditional, looping and external source connectives. To overcome the above mentioned problems Selenium RC is used. Since Selenium RC supports the languages .Net, Java, Perl, Python, PHP, and Ruby. In these languages we can write the programme to achieve the IDE issues

What is Selenium Grid?
Selenium Grid is part of Selenium suite of projects. Selenium Grid transparently distribute your tests on multiple machines so that you can run your tests in parallel, cutting down the time required for running in-browser test suites.This will dramatically speeds up in-browser web testing, giving you quick and accurate feedback you can rely on to improve your web application.

What is Selenium WebDriver or Google WebDriver or Selenium 2.0?
WebDriver uses a different underlying framework from Selenium's javascript Selenium-Core. It also provides an alternative API with functionality not supported in Selenium-RC. WebDriver does not depend on a javascript core embedded within the browser, therefore it is able to avoid some long-running Selenium limitations. WebDriver's goal is to provide an API that establishes A well-designed standard programming interface for web-app testing. Improved consistency between browsers. Additional functionality addressing testing problems not well-supported in Selenium 1.0.

The Selenium developers strive to continuously improve Selenium. Integrating WebDriver is another step in that process. The developers of Selenium and of WebDriver felt they could make significant gains for the Open Source testautomation community be combining forces and merging their ideas and technologies. Integrating WebDriverinto Selenium is the current result of those efforts.

Does Selenium support mobile internet testing?
Selenium supports Opera. And opera is used in most of the Smart phones. So whichever Smart phone supports opera, selenium can be used to test. So, one can use Selenium RC to run the tests on mobiles.

Does Selenium support Google Android Operating System?
Yes, Selenium Web Driver or Google Web Driver or Selenium 2.0 supports Android Operating System. There are several libraries written to support Android Operating System.

What are the types of text patterns available in Selenium?
There are three types of patterns available in Selenium globbing, regular expressions, exact.

What is Selenese?
Selenium set of commands which are used for running the test are called as Selenese.There are three types of Selenese, those are:
Actions - used for performing the operations and interactions with the target elements
Assertions - used as check points
Accessors - used for storing the values in a variable

How do you add check points or verification points in Selenium?
check points or verification points are known as Assertions in Selenium. The keywords with below mentioned prefix willbe used for adding check points or verification points. Verify**,** assert**,** waitFor

What are the types of Assertions there in Selenium?
Selenium Assertions can be used in 3 modes:

assert - When an "assert" fails, the test will be aborted. If you are executing test suite, the next state case will start

verify - When a "verify" fails, the test will continue execution, logging the failure.

waitFor - "waitFor" commands wait for some condition to become true (which can be useful for testing Ajax applications). They will succeed immediately if the condition is already true. However, they will fail and halt the test if the condition does not become true within the current timeout setting

When to use Assert, Verify and WaitFor in Selenium?
assert - If the expected value is mandatory to continue with the next set of steps we will use Assert. As Assert abortsthe test, if the expected value doesn't match. It is good to use for any mandatory checks.

verify - If the expected value is optional to continue with the next set of steps we will use Verify. As Verify continuesexecuting with the next set of steps, if the expected value doesn't match. It is good to use for any optional checks.

waitFor - If your test needs to wait, if the expected value is not matching we use waitFor. We normally use waitFor forAJAX kind of controls loading within a page

What is the difference between assert and verify? What are types of assert?
In the case of the “Assert” command, as soon as the validation fails the execution of that particular test method is stopped. Following that the test method is marked as failed. Whereas, in the case of “Verify”, the test method continues execution even after the failure of an assertion statement. Although the test method will still be marked as failed but the remaining statements of the test method will be executed normally.

We have two types of assertions; **hard assertion and soft assertion**.

Hard Assertions: A hard assertion does not continue with execution until the assertion condition is True. Hard assertions usually throw an Assertion Error whenever an assertion condition fails. The test case will be immediately marked as Failed when a hard assertion condition fails.
Soft Assertions: A soft assertion continues with test execution even if the assertion condition fails. Soft Assertion does not throw any error when the assertion condition fails but continues with the next step of the test case.