1. What is Xpath
   Ans- X path is xml path, it Is one of the locator which help us to locate the webelement.Address of Element. Using x path we can locate and perform action.
   Analogy- your friend ask you to pick
2. Difference between Relative(//) and absolute(/) Xpath?
   Ans- if you are identifying x path from start node(<html>) of xml is called absolute X path. Relative x path start in between of xml.
   Analogy- You are going for shopping 1) you can start from entry of mall and search formal shirt.
   2)you direct go to men section and find out shirt is relative.
   Advantage of relative x path
   - Avoid long x path
   - If starting node is changed then no need to modify script
3. How do you uniquely identify the element which has same X path?
   Ans- I have seen in my application where 2 or more than 2 elements having same x path in that case I used index to uniquely identify that. [1] [2] etc. Radio button I got.
4. Difference between findElement & findElements?

findElement	findElements
Return single element	Return multiple elements in form of List
Always return 1st element	Return all element
Throw Exception if element not found(NoSuchElement)	Empty list
FindElement()	FindElements()

5. Difference between implicit and Explicit Wait.
   Implicit- it will wait for defined time mentioned in the script whether page loaded earlier or not it will wait. It will wait only for element identification. This will wait for all element identification
   driver.manage().timeouts().implicitlywait(20,TimeUnit.SECEND);

   Explicit- Smarter than Implicit, it will move to next step if condition is met early
   WebDriverWait wait=new WebDriverWait(driver,20);
   Wait.until(Expectedconditons.visibilityOfElementLocated(By.id("ffu")));

   Most of the I have used explicit wait.
6. Total number of Links in your webpage?
   List<WebElement> links=driver.findElements(By.xpath("//a"); or by tagname("a")
   Int count=links.size();

   For(int i=0;i<count;i++)
   {
   String s=linkes.get(i).getText();
   Sopln(s);
   }

7. Difference b/w close() & quit()
   Close will close current browser which you are working on. Quit will close all browser which is opened in test session. All browser instances.
8. How do you switch between Frames

9. Assert & verify.
   Assert – It will stop execution of script after this.Next line will not be executed
   Verify- It will continue execution of next line

```
@Test
Public void f1()
{
Assert.assertEquals(a,e) //failed
Hg; //skipped
Kj; //skipped
Hu; //skipped
}

@Test
Public void f1()
{
Verify //failed
Hg; //Executed
Kj; // Executed
Hu; // Executed
}
```

10. What is Automation Testing?
    Automation testing is process of testing of software application using automation tool to find **defects**. Most popular tool selenium,APM UFT,RFT etc.
    Automation test is required when we have huge amout of regression test.
11. Benefit of automation Test?
    - Save time and Money
    - Easy reporting(it generate correctly) in manual it can be human error
    - Compatablity testing on different OS
    - Automation testing is more realable
    - Minimum manual intervation
    - Maxium coverage
    - More powerfull and versetile due to itegration
    - Mostly used for regression testing
    - No manual intervention is required, we can scheduled through Jenkins in night
12. What type of test have you automated?
    Regression testing in terms of functional testing
    Sanity test and smoke testing
13. How many test caes automated per day.
    Depends on complexity of application and scenarios
    5-6: for complex application and lengthy scenario
    20: easy application and small scenarios
    Can be 2-3 for very lengthy
14. What is Framework?
    A framework define a set of rule or best practice which we can follow in systematic way to achive desired result is known as Framework.
    Below type of Framework
    - Data driven
    - Keyword driven
    - Hybrid driven(combined 2 or more together)
15. Have you created any framework?
    Yes I have created or I have involved in framework creation.
    Manipulate your answer acoordingly as per company requirement.
16. Can you explain the framework which you have used in your project?
    As per copy
17. What do you prefer using selenium?

- Free and open source
- Great support through community/blocks/blog
- Cross browser testing support(FireFox,IE,Chrome,safari,Opera ETC)
- Support multiple platform(Window,Apple macbook,linux etc)
- Multiple language support by selenium WebDriver(java,perl,python C Sharp,Ruy,PHP,perl)
- Provide easy integration (TestNG,AutoIT,Seculi,Maven, Junit,Jenkins..)

QTP

1. Cost
2. Heavy in size
3. Support only IE
4. Only vb screepting not object oriented pgm

Adv

Own data driven approach

Own kEyword driven approach

Own repository

Own

18. What is WebDriver
    Selenium webdriver (selenium 2 or selenium 3) is used for automation.
    And explain y selenium
    Webdriver doesnot required any server. This is completely API
    Selenium Webdriver architechture Explain….

19. What type of WebDriver available in Selenium?
    - FireFox Driver
    - Gecko Driver
    - InternetExplorer Driver
    - Chrome Driver
    - Html/Unit Driver
    - Opera Driver
    - Safari Driver
    - Android Driver
    - iPhone Driver
    - EvenFiring Webdriver
20. Which WebDriver implementation claims to be fastest?
    Html/Unit Driver- it doesnot launch any webpage(selenium3.0)
    Go through headless browser testing

21. What kind of OpenSource Framework support by Selenium
    - Junit - TDD
    - TestNG –TDD (Test driven dev)
    - Cucumber –BDD FM(behavior driven development)
    - Jbehave –BDD FM
22. Explain locators
    In selenium WebDriver we have 8 types of locator in By class
    - Tag name
    - Id
    - Name
    - Class
    - Link text
    - Partial link text
    - Css selector
    - X path
23. Soft assert and Hard assert

Soft- soft assert dnt throw any error immidiatly. It will collect all the error and throw after execution.
Hard- if assertion fail ,it will terminate the pgm

Soft A1-failed
Soft A2-failed
Soft A3-Passed

hard A1-failed
Soft A2- Not executed
Soft A3-Not executed

```
public class AssertSample {
 @Test
 public void fA()
 {
            Assert.assertEquals("Google", "Google");
 }
 @Test
 public void fB()
 {
            Assert.assertEquals("gmail", "gmail");
 }
 @Test
 public void fC()
 {
            Assert.assertEquals("gmail", "yahoo");
 }
 @Test(dependsOnMethods ="fC")
 public void fD()
 {
            Assert.assertEquals("gmail", "gmail");
 }
          }
```

PASSED: fA
PASSED: fB
FAILED: fC
SKIPPED:fD

24. Verification point available in selenium
    Assert class which is present in TestNG, in selenium WebDRiver we donot have verification.

25. How to launch browser
    <2.53 no geckodriver
    3.x- geckodriver

    System.setProperty("webdriver.chrome.driver", "Path of chromrdriver.exe")
    WebDriver driver=new ChromeDriver();
    WebDriver driver=new FirefoxDriver();

26. Is FirefoxDriver is class or Interface
    FireFox is class which is implementing WebDriver interface.

    WebDriver driver=new ChromeDriver();
    Driver is object reference name.
    Its Dynamic polymorphism/dynamic binding/runtime polymorphism

27. Super interface of wEbDriver
    SearchContext
28. Explain WebDriver driver=new ChromeDriver();
    Explained earlier, inhertitance concept
29. Why WebDriver driver=new ChromeDriver();? Why not FirefoxDriver driver=new ChromeDriver();?
    If we have to change browser after some day we have to change entire code. It again rework
    We can not achieve cross browser Testing using this.

```
If( browserName.equals("chrome"))
{
WebDriver driver=new ChromeDriver;
}
Elseif()
{
}
Else
{
}
```

30. What are different Execption in selenium, Please explain?
    - WebDriverException
    - TimeOutException
    - NoAlertExecption
    - NoSuchElementExecption
    - NoSuchWindowExeption
    - StaleElementReferenceExecption

31.what is constructor?use of constructor in class?

Constructor is a block of code that initializes the newly created object. A constructor resembles an instance method in java but it's not a method as it doesn't have a return type. In short constructor and method are different.

32.why we need to compile a program?

A compiler is a program that translates from one language into another — in general, it's used to mean a program which reads in a source code language and translates to an object code form, which is almost always the machine language of a microprocessor, or sometimes a "bytecode" language for an interpreter (or for another compiler).

33.upcasting in java?

Creating object of child class and store as a reference of parent type

```
B b=new A();
```

34.polymorphism

Polymorphism is a OOPs concept where one name can have many forms.For example, you have a smartphone for communication. The communication mode you choose could be anything. It can be a call, a text message, a picture message, mail, etc. So, the goal is common that is communication, but their approach is different. This is called Polymorphism**.**

35.singleTon class

Singleton class means you can create only one object for the given class. You can create a singleton class by making its constructor as private, so that you can restrict the creation of the object. Provide a static method to get instance of the object, wherein you can handle the object creation inside the class only. In this example we are creating object by using static block.

```
package com.java2novice.algos;
```

```
public class MySingleton {

   private static MySingleton myObj;

   static
            {
      myObj = new MySingleton();
   }

   private MySingleton(){

   }

   public static MySingleton getInstance(){
      return myObj;
   }

   public void testMe(){
      System.out.println("Hey.... it is working!!!");
   }

   public static void main(String a[]){
      MySingleton ms = getInstance();
      ms.testMe();
   }
}
```

36.overloading method

Method Overloading is a feature that allows a class to have more than one method having the same name, if their argument lists are different. It is similar to constructor overloading in Java, that allows a class to have more than one constructor having different argument lists.

let's get back to the point, when I say argument list it means the parameters that a method has: For example the argument list of a method add(int a, int b) having two parameters is different from the argument list of the method add(int a, int b, int c) having three parameters.

Important Points

- Two or more methods can have same name inside the same class if they accept different arguments. This feature is known as method overloading.
- Method overloading is achieved by either:
    - changing the number of arguments.
    - or changing the datatype of arguments.
- Method overloading is not possible by changing the return type of methods.

37.what is testng?use of testng in framework?

TestNG is a testing framework inspired from JUnit and NUnit with more functionality added to make execution more efficient and powerful. It is an open source automated testing framework. It is similar to JUnit but it is more powerful than JUnit. 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 & parameterising.

Benefits of TestNG

Major advantages of testing are:

• Logs can be generated
• Annotations make code efficient and easy to manage
• Ability to produce HTML Reports of execution
• Test cases can be Grouped & Prioritised
• Parallel testing is possible
• Data Parameterization is possible

Annotations in TestNG

@BeforeSuite: Method with this annotation will run before all tests in the test suite
@AfterSuite: Method with this annotation will run after all tests in the test suite

@BeforeTest: Method with this annotation will run before each and every test method with tag in xml file
@AfterTest: Method with this annotation will run after each and every test method with tag in xml file
@BeforeClass: Method with this annotation will run before first test method in current class
@AfterClass: Method with this annotation will run after last test method in current class
@BeforeMethod: Method with this annotation will run before each test method
@AfterMethod: Method with this annotation will run after each test method
@Test: The annotated method is a part of a test case

38. how to handle alert,file upload popups
Generally JavaScript popups are generated by web application and hence they can be easily controlled by the browser.
Webdriver offers the ability to cope with javascript alerts using Alerts APIClick here to view Alert API Details
// Get a handle to the open alert, prompt or confirmation
Alert alert = driver.switchTo().alert();
Alert is an interface. There below are the methods that are used
//Will Click on OK button.
alert.accept();
// Will click on Cancel button.
alert.dismiss()
//will get the text which is present on th Alert.
alert.getText();
//Will pass the text to the prompt popup
alert.sendkeys();
//Is used to Authenticate by passing the credentials
alert.authenticateUsing(Credentials credentials)

39.Access modifiers public,private,protected use in selenium
40.What is POM.XML

POM stands for Project Object Model. It is fundamental unit of work in Maven. It is an XML file that resides in the base directory of the project as pom.xml.

The POM contains information about the project and various configuration detail used by Maven to build the project(s).

POM also contains the goals and plugins. While executing a task or goal, Maven looks for the POM in the current directory. It reads the POM, gets the needed configuration information, and then executes the goal. Some of the configuration that can be specified in the POM are following −

- project dependencies
- plugins
- goals
- build profiles
- project version
- developers
- mailing list

Before creating a POM, we should first decide the project group (groupId), its name (artifactId) and its version as these attributes help in uniquely identifying the project in repository.

```
<project xmlns = "http://maven.apache.org/POM/4.0.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.companyname.project-group</groupId>
  <artifactId>project</artifactId>
  <version>1.0</version>
</project>
```

It should be noted that there should be a single POM file for each project.

- All POM files require the **project** element and three mandatory fields: **groupId, artifactId, version**.
- Projects notation in repository is **groupId:artifactId:version**.
- Minimal requirements for a POM −

41.difference between .xml & .html
Similarities between HTML and XML?
1. Both are languages of web.
2. Both are markup languages.
3. Both are originated from SGML. [Standardized General Markup Language]
4. Tags are basic building blocks of both HTML and XML documents.

Differences between HTML and XML
1. HTML tags are predefined tags where as XML tags are user defined tags.
2. HTML tags are limited number of tags where as XML tags are extensible.
3. HTML tags are case insensitive where as XML tags are sensitive.
4. HTML tags are meant for displaying the data but not for describing the data where as XML tags are meant for describing the data.
5. HTML focuses on how data looks where as XML focuses on what data is.
7.handling frames in selenium

42. what is polymorphism?types of polymorphism?

43. Access Modiefiers(Private,public,protected)

44. Object life cycle architecture?

As you work with objects in Java, understanding how objects are born, live their lives, and die is important. This topic is called the life cycle of an object, and it goes something like this:

45. Before an object can be created from a class, the class must be loaded. To do that, the Java runtime locates the class on disk (in a .class file) and reads it into memory. Then Java looks for any static initializers that initialize static fields — fields that don't belong to any particular instance of the class, but rather belong to the class itself and are shared by all objects created from the class.

A class is loaded the first time you create an object from the class or the first time you access a static field or method of the class. For example, when you run the main method of a class, the class is initialized because the main method is static.

2. An object is created from a class when you use the new keyword. To initialize the class, Java allocates memory for the object and sets up a reference to the object so the Java runtime can keep track of it. Then, Java calls the class constructor, which is like

a method but is called only once, when the object is created. The constructor is responsible for doing any processing required to initialize the object, such as initializing variables, opening files or databases, and so on.

3. The object lives its life, providing access to its public methods and fields to whoever wants and needs them.

4. When it's time for the object to die, the object is removed from memory and Java drops its internal reference to it. You don't have to destroy objects yourself. A special part of the Java runtime called the garbage collector takes care of destroying all objects when they are no longer in use

5.Innerclass &Anonymous class?

6.how to print without main method "hello"?

by using static block

```
static
{
System.out.println("Hello world");
}
System.exit(0);
```

The java.lang.**System**.**exit**() method **exits** current program by terminating running Java virtual machine. This method takes a status code. A non-**zero** value of status code is generally used to indicate abnormal termination. ... **exit**(1) or **exit**(-1) or any other non-**zero** value – Generally indicates unsuccessful termination.

7.Method overloading & overriding & Inheritance?

1.what is selenium webdriver?advantage of against RC,IDE?

WebDriver is a web automation framework that allows you to **execute your tests against different browsers**, not just Firefox (unlike Selenium IDE).

WebDriver also enables you to **use a programming language** in creating your test scripts (not possible in Selenium IDE).

- You can now use **conditional operations** like if-then-else or switch-case
- You can also perform **looping** like do-while.

Following programming languages are supported by WebDriver

- Java
- .Net
- PHP
- Python
- Perl
- Ruby

You do not have to know all of them. You just need to be knowledgeable in one. However, in this tutorial, we will be using Java with Eclipse as our IDE.

Difference between Selenium RC and Webdriver

Before the advent of WebDriver in 2006, there was another, **automation tool called Selenium Remote Control.** Both WebDriver and Selenium RC have following features:

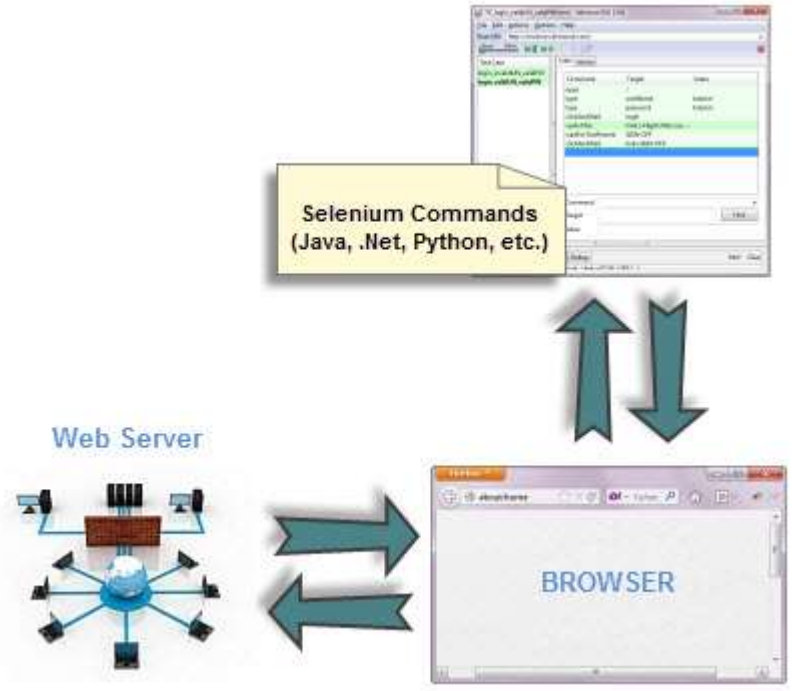
- They both allow you to **use a programming language** in designing your test scripts.
- They both allow you to **run your tests against different browsers.**

So how do they differ? Let us discuss the answers.

1. Architecture

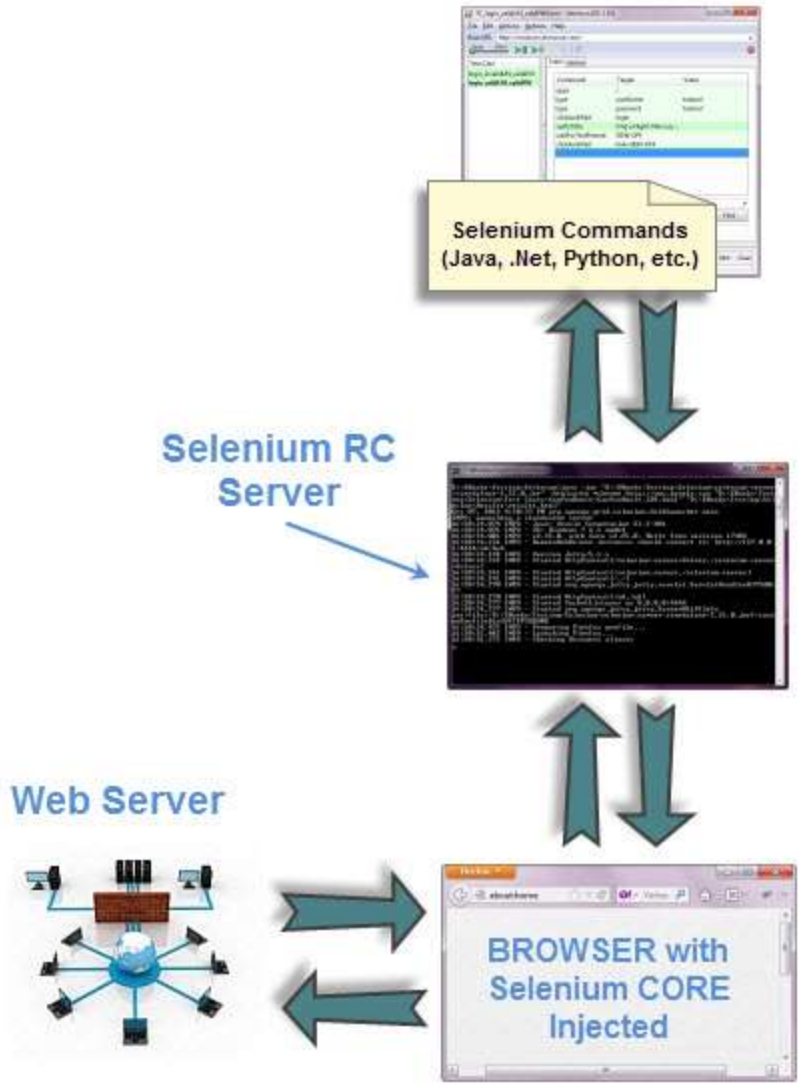
WebDriver's architecture is simpler than Selenium RC's.

- It controls the browser from the OS level
- All you need are your programming language's IDE (which contains your Selenium commands) and a browser.



Selenium RC's architecture is way more complicated.

- You first need to launch **a separate application called Selenium Remote Control (RC) Server** before you can start testing
- The Selenium RC Server **acts as a "middleman" between your Selenium commands and your browser**
- When you begin testing, Selenium RC Server "injects" a **Javascript program called Selenium Core** into the browser.
- Once injected, Selenium Core will start receiving instructions relayed by the RC Server from your test program.
- When the instructions are received, **Selenium Core will execute them as Javascript commands.**
- The browser will obey the instructions of Selenium Core and will relay its response to the RC Server.
- The RC Server will receive the response of the browser and then display the results to you.
- RC Server will fetch the next instruction from your test script to repeat the whole cycle.



2. Speed

WebDriver is faster than Selenium RC since it speaks directly to the browser uses the browser's own engine to control it.

Selenium RC is slower since it uses a Javascript program called Selenium Core. This Selenium Core is the one that directly controls the browser, not you.

3. Real-life Interaction

WebDriver interacts with page elements in a more realistic way. For example, if you have a disabled text box on a page you were testing, WebDriver really cannot enter any value in it just as how a real person cannot.

Selenium Core, just like other JavaScript codes, can access disabled elements. In the past, Selenium testers complain that Selenium Core was able to enter values to a disabled text box in their tests. Differences in API

4. API

Selenium RC's API is more matured but contains redundancies and often confusing commands. For example, most of the time, testers are confused whether to use type or typeKeys; or whether to use click, mouseDown, or mouseDownAt. Worse, **different browsers interpret each of these commands in different ways too!**
WebDriver's API is simpler than Selenium RC's. It does not contain redundant and confusing commands.

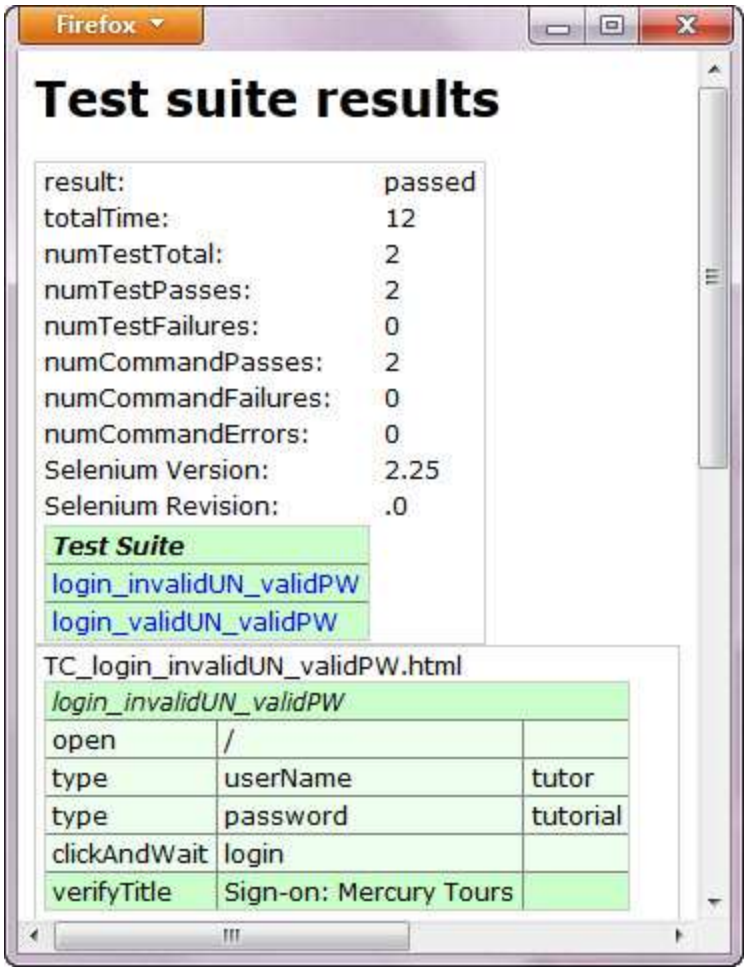
5. Browser Support

WebDriver can support the headless HtmlUnit browser
HtmlUnit is termed as "headless" because it is an invisible browser - it is GUI-less.
It is a very fast browser because no time is spent in waiting for page elements to load. This accelerates your test execution cycles.
Since it is invisible to the user, it can only be controlled through automated means.
Selenium RC cannot support the headless HtmlUnit browser. It needs a real, visible browser to operate on.

Limitations of WebDriver

WebDriver Cannot Readily Support New Browsers
Remember that WebDriver operates on the OS level. Also, remember that different browsers communicate with the OS in different ways. If a new browser comes out, it may have a different process of communicating with the OS as compared to other browsers. So, **you have to give the WebDriver team quite some time to figure that new process out** before they can implement it on the next WebDriver release.
However, it is up to the WebDriver's team of developers to decide if they should support the new browser or not.

Selenium RC Has Built-In Test Result Generator
Selenium RC automatically generates an HTML file of test results. The format of the report was pre-set by RC itself. Take a look at an example of this report below.



WebDriver has no built-in command that automatically generates a Test Results File. You would have to rely on your IDE's output window, or design the report yourself using the capabilities of your programming language and store it as text, HTML, etc.

Summary

- WebDriver is a tool for testing web applications **across different browsers** using different programming languages.
- You are now able to make powerful tests because WebDriver **allows you to use a programming language** of your choice in designing your tests.
- WebDriver is **faster than Selenium RC** because of its simpler architecture.
- WebDriver **directly talks to the browser** while Selenium RC needs the help of the RC Server in order to do so.
- WebDriver's API is more **concise** than Selenium RC's.
- WebDriver **can support HtmlUnit** while Selenium RC cannot.
- The only drawbacks of WebDriver are:
    - It **cannot readily support new browsers**, but Selenium RC can.
    - It **does not have a built-in command** for automatic generation of test results.

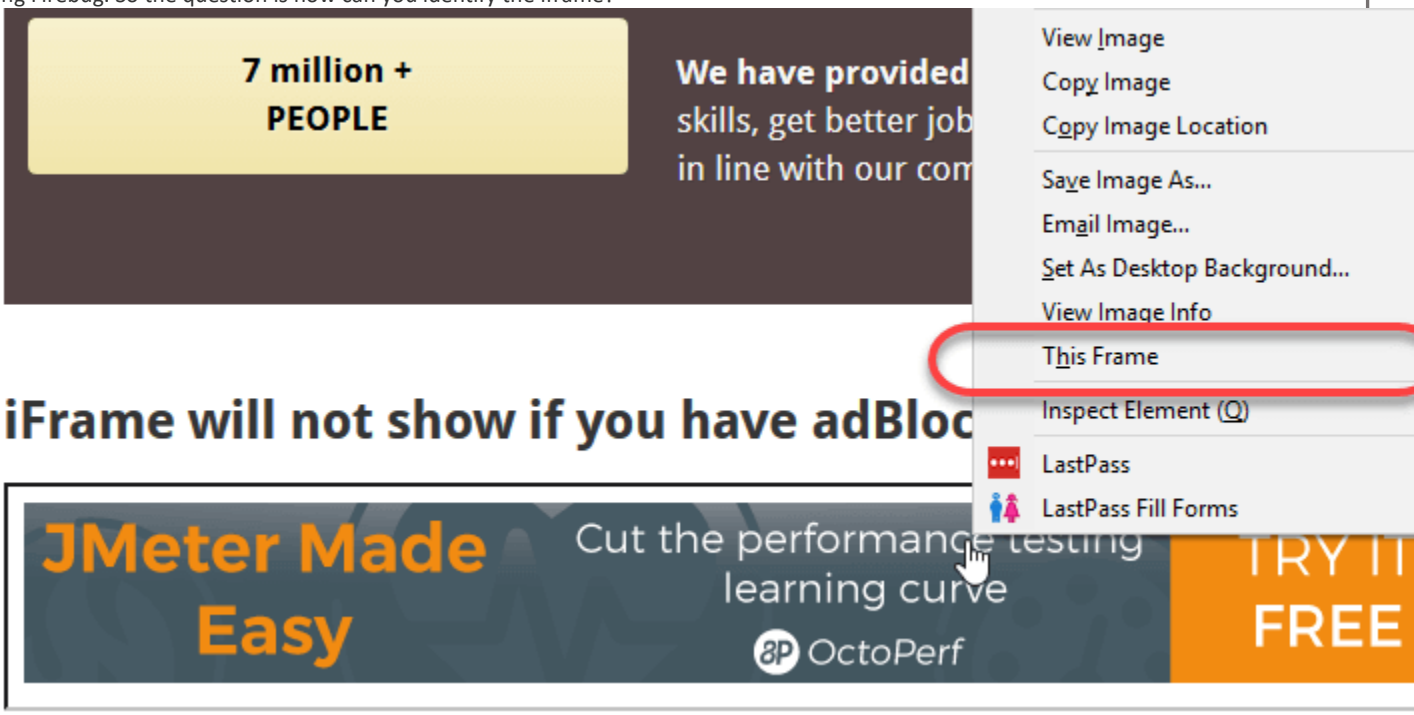
2.How to handle javascript popup?

Alert class see above

3.How to handle iframes?
What is Iframe?

IFrame is a web page which is embedded in another web page or an HTML document embedded inside another HTML document.
The IFrame is often used to insert content from another source, such as an advertisement, into a Web page. The <**iframe**> tag specifies an inline frame.
In this tutorial, you will learn -
1. How to identify the iframe:
2. How do switch over the elements in iframes using Web Driver commands:
3. Concept of Nested Frames (Frames inside Frames):

How to identify the iframe:
We cannot detect the frames by just seeing the page or by inspecting Firebug.
Observe the below image, Advertisement being displayed is an Iframe, we cannot locate or recognize that by just inspecting using Firebug. So the question is how can you identify the iframe?



We can identify the iframes using methods given below:

- Right click on the element, If you find the option like 'This Frame' then it is an iframe.(Please refer the above diagram)
- Right click on the page and click 'View Page Source' and Search with the 'iframe', if you can find any tag name with the 'iframe' then it is meaning to say the page consisting an iframe.

In above diagram, you can see that '**This Frame**' option is available upon right clicking, so we are now sure that it is an iframe.
We can even identify total number of iframes by using below snippet.

```
Int size = driver.findElements(By.tagName("iframe")).size();
```

How to switch over the elements in iframes using Web Driver commands:
Basically, we can switch over the elements in frames using 3 ways.

- **By Index**
- **By Name or Id**
- **By Web Element**

Switch to the frame by index:
Index is one of the attributes for the Iframe through which we can switch to it.
Index of the iframe starts with '0'.
Suppose if there are 100 frames in page, we can switch to the iframe by using index.

- driver.switchTo().frame(0);
- driver.switchTo().frame(1);

Switch to the frame by Name or ID:

Name and ID are attributes of iframe through which we can switch to the it.

- driver.switchTo().frame("iframe1");
- driver.switchTo().frame("id of the element");

Example of Switching to iframe through ID:

Let's take an example of iframe displaying in the below image. Our requirement is to click the iframe.
We can access this iframe through this below URL:http://demo.guru99.com/test/guru99home/

It is impossible to click iframe directly through XPath since it is an iframe. First we have to switch to the frame and then we can click using xpath.

Step 1)

```
WebDriver driver = new FirefoxDriver();


driver.get("http://demo.guru99.com/test/guru99home/");


driver.manage().window().maximize();
```

- We initialise the Firefox driver.
- Navigate to the "guru99" site which consist the iframe.
- Maximized the window.

Step 2)

```
driver.switchTo().frame("a077aa5e");
```

- In this step we need to find out the id of the iframe by inspecting through Firebug.
- Then switch to the iframe through ID.

Step 3)

```
driver.findElement(By.xpath("html/body/a/img")).click();
```

- Here we need to find out the xpath of the element to be clicked.
- Click the element using web driver command shown above.

Here is the complete code:

```


public class SwitchToFrame_ID {
public static void main(String[] args) {


                        WebDriver driver = new FirefoxDriver(); //navigates to the Browser
                driver.get("http://demo.guru99.com/test/guru99home/");
                  // navigates to the page consisting an iframe


                  driver.manage().window().maximize();
                  driver.switchTo().frame("a077aa5e"); //switching the frame by ID
```

```
                                        System.out.println("********We are switch to the iframe*******");
                                driver.findElement(By.xpath("html/body/a/img")).click();
                                  //Clicks the iframe


                                        System.out.println("*********We are done***************");
    }
}
```

Output:

Browser navigates to the page consisting the above iframe and clicks on the iframe.

Switch to the frame by Web Element:

We can even switch to the iframe using web element .

- driver.switchTo().frame(WebElement);

How to switch back to the Main Frame

We have to come out of the iframe.

To move back to the parent frame, you can either use switchTo().parentFrame() or if you want to get back to the main (or most parent) frame, you can use switchTo().defaultContent();

```
driver.switchTo().parentFrame();
driver.switchTo().defaultContent();
```

How to switch over the frame, if we CANNOT switch using ID or Web Element:

Suppose if there are 100 frames in the page, and there is no ID available, in this case, we just don't know from which iframe required element is being loaded (It is the case when we do not know the index of the frame also).

The solution for the above concern is, we must find the index of the iframe through which the element is being loaded and then we need to switch to the iframe through the index.

Below are the steps for finding the index of the Frame by which the element is being loaded by using below snippet

Step 1)

```
WebDriver driver = new FirefoxDriver();
driver.get("http://demo.guru99.com/test/guru99home/");
driver.manage().window().maximize();
```

- Initialise the Firefox driver.
- Navigate to the "guru99" site which consisting the iframe.
- Maximized the window.

Step 2)

```
int size = driver.findElements(By.tagName("iframe")).size();
```

- The above code finds the total number of iframes present inside the page using the tagname 'iframe'.

Step 3)

Objective for this step would be finding out the index of iframe.

```
for(int i=0; i<=size; i++){
        driver.switchTo().frame(i);
        int total=driver.findElements(By.xpath("html/body/a/img")).size();
        System.out.println(total);
          driver.switchTo().defaultContent();}
```

Above "forloop" iterates all the iframes in the page and it prints '1' if our required iframe was found else returns '0'.

Here is the complete code till step 3:

```
public class IndexOfIframe {
public static void main(String[] args) {
          WebDriver driver = new FirefoxDriver();
          driver.get("http://demo.guru99.com/test/guru99home/");
          driver.manage().window().maximize();
```

```
//driver.manage().timeouts().implicitlyWait(100, TimeUnit.SECONDS);
int size = driver.findElements(By.tagName("iframe")).size();

for(int i=0; i<=size; i++){
        driver.switchTo().frame(i);
        int total=driver.findElements(By.xpath("html/body/a/img")).size();
        System.out.println(total);
driver.switchTo().defaultContent();}}}
```

Execute this program and output would be like below:

Output:

1
0
0
0
0
0

Verify the output, you can find the series of 0's and 1's.

- Wherever you find the '1' in output that is the index of Frame by which the element is being loaded.
- Since the index of the iframe starts with '0' if you find the 1 in the 1st place, then the index is 0.
- If you find 1 in 3rd place, the index is 2.

We can comment out the for loop, once we found the index.

Step 4)

```
driver.switchTo().frame(0);
```

- Once you find the index of the element, you can switch over the frame using above command.
- driver.switchTo().frame(index found from the Step 3);

Step5)

```
driver.findElement(By.xpath("html/body/a/img")).click();
```

- The above code will clicks the iframe or element in the iframe.

So the complete code would be like below:

```
public class SwitchToframe   {
public static void main(String[] args) throws NoSuchElementException{
        WebDriver driver = new FirefoxDriver();
        driver.get("http://demo.guru99.com/test/guru99home/");
        driver.manage().window().maximize();
        //int size = driver.findElements(By.tagName("iframe")).size();

      /*for(int i=0; i<=size; i++){
        driver.switchTo().frame(i);
        int total=driver.findElements(By.xpath("html/body/a/img")).size();
                System.out.println(total);
        driver.switchTo().defaultContent(); //switching back from the iframe
       }*/

                //Commented the code for finding the index of the element
        driver.switchTo().frame(0); //Switching to the frame
                System.out.println("********We are switched to the iframe*******");
                driver.findElement(By.xpath("html/body/a/img")).click();
```

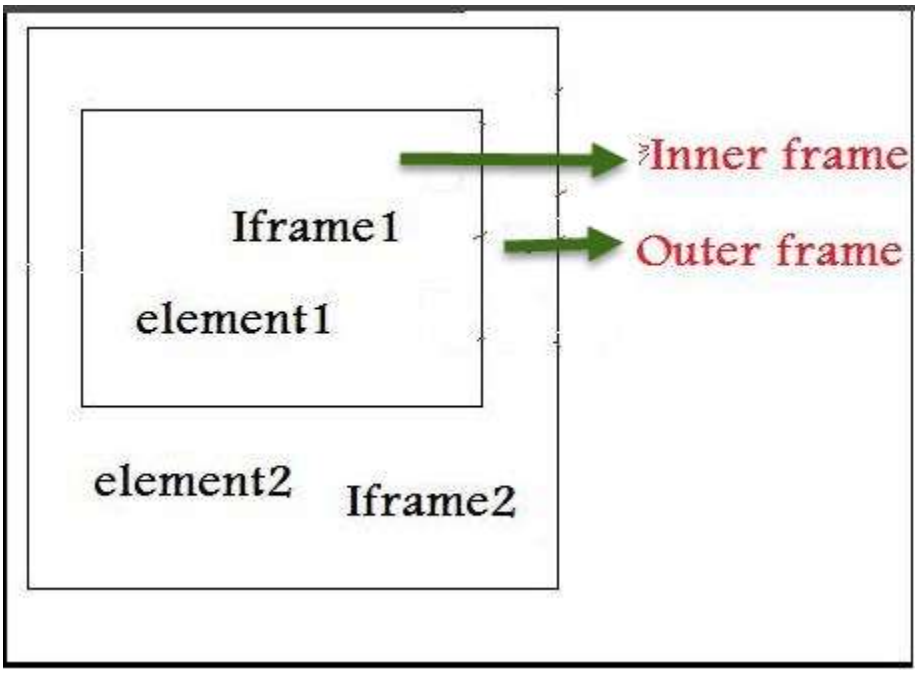
```
                    //Clicking the element in line with Advertisement
            System.out.println("**********We are done***************");
              }
            }
```

Output:
Browser navigates to the page consisting the above iframe and clicks on the iframe.

Concept of Nested Frames(Frames inside Frames):
Let's assume that there are two frames one inside other like shown in below image and our requirement is printing the text in the outer frame and inner frame.
In the case of nested frames,

- At first we must switch to the outer frame by either Index or ID of the iframe
- Once we switch to the outer frame we can find the total number of iframes inside the outer frame, and
- We can switch to the inner frame by any of the known methods.

While exiting out of the frame, we must exit out in the same order as we entered into it from the inner frame first and then outer frame.



The Html code for the above nested frame is as shown below.

The above HTML code clearly explains the iframe tag (highlighted in green) within another iframe tag, indicating presence of nested iframes.
Below are the steps for switching to outer frame and printing the text on outer frames:
Step 1)

```
WebDriver driver=new FirefoxDriver();
    driver.get("Url");
    driver.manage().window().maximize();
    driver.manage().timeouts().implicitlyWait(2, TimeUnit.SECONDS);
    int size = driver.findElements(By.tagName("iframe")).size();
    System.out.println("Total Frames --" + size);

            // prints the total number of frames
            driver.switchTo().frame(0); // Switching the Outer Frame
    System.out.println (driver.findElement(By.xpath("xpath of the outer element ")).getText());
```

- Switch to the outer Frame.
- Prints the text on outer frame.

Once we switch to the outer frame, we should know whether any inner frame present inside the outer frame

Step 2)

```
size = driver.findElements(By.tagName("iframe")).size();
  // prints the total number of frames inside outer frame
  System.out.println("Total Frames --" + size);
```

- Finds the total number of iframes inside outer frame.
- If size was found '0' then there is no inner frame inside the frame.

Step 3)

```
driver.switchTo().frame(0); // Switching to innerframe
System.out.println(driver.findElement(By.xpath("xpath of the inner element ")).getText());
```

- Switch to the inner frame
- Prints the text on the inner frame.

Here is the complete code:

```
public class FramesInsideFrames {
public static void main(String[] args) {
WebDriver driver=new FirefoxDriver();
            driver.get("Url");
            driver.manage().window().maximize();
            driver.manage().timeouts().implicitlyWait(2, TimeUnit.SECONDS);

            int size = driver.findElements(By.tagName("iframe")).size();
            System.out.println("Total Frames --" + size);

                    // prints the total number of frames
                    driver.switchTo().frame(0); // Switching the Outer Frame
            System.out.println (driver.findElement(By.xpath("xpath of the outer element ")).getText());

                    //Printing the text in outer frame
                    size = driver.findElements(By.tagName("iframe")).size();
            // prints the total number of frames inside outer frame

            System.out.println("Total Frames --" + size);
            driver.switchTo().frame(0); // Switching to innerframe
```

```
                System.out.println(driver.findElement(By.xpath("xpath of the inner element ")).getText());

                        //Printing the text in inner frame
                        driver.switchTo().defaultContent();
            }
}
```

Output:
The output of the above code would print the text in the Inner frame and Outer frame.
This article on "Handling frames through Selenium" is contributed by Raghavi Avadhuri.

4.how to hadle filedownload popup?

5.what is the use of TestngListeners?

7.what is ant & use in framework?

1. What is Automation Testing?
Automation testing is the process of testing the software using an automation tool to find the defects. In this process, executing the test scripts and generating the results are performed automatically by automation tools. Some most popular tools to do automation testing are HP QTP/UFT, Selenium WebDriver, etc.,
2. What are the benefits of Automation Testing?
This is one of the common interview questions in any Automation testing job.

1. Saves time and money. Automation testing is faster in execution.
2. Reusability of code. Create one time and execute multiple times with less or no maintenance.
3. Easy reporting. It generates automatic reports after test execution.
4. Easy for compatibility testing. It enables parallel execution in the combination of different OS and browser environments.
5. Low-cost maintenance. It is cheaper compared to manual testing in a long run.
6. Automated testing is more reliable.
7. Automated testing is more powerful and versatile.
8. It is mostly used for regression testing. Supports execution of repeated test cases.
9. Minimal manual intervention. Test scripts can be run unattended.
10. Maximum coverage. It helps to increase the test coverage.

3. What are the challenges and limitations of Selenium WebDriver?
As we all know Selenium WebDriver is a tool which automates the browser to mimic real user actions on the web. Selenium is a free open source testing tool. Some of the challenges with selenium webdriver are as follows

1. We cannot test windows application
2. We cannot test mobile apps
3. Limited reporting
4. Handling dynamic Elements
5. Handling page load
6. Handling pop up windows
7. Handling captcha

4. What type of tests have you automated?
Our main focus is to automate test cases to do *Regression testing, Smoke testing, and Sanity testing.* Sometimes based on the project and the test time estimation, we do focus on End to End testing.
5. How many test cases you have automated per day?
It depends on Test case scenario complexity and length. I did automate 2-5 test scenarios per day when the complexity is limited. Sometimes just 1 or fewer test scenarios in a day when the complexity is high.
6. What is a Framework?
A framework defines a set of rules or best practices which we can follow in a systematic way to achieve the desired results. There are different types of automation frameworks and the most common ones are:

- Data Driven Testing Framework
- Keyword Driven Testing Framework
- Hybrid Testing Framework

Detailed Explanation: Types of Framework

7. Have you created any Framework?

If you are a beginner: No, I didn't get a chance to create a framework. I have used the framework which is already available.
If you are an experienced tester: Yes, I have created a framework. Or I have involved in the creation of the framework.

8. Can you explain the Framework which you have used in your Selenium Project?

Here we have clearly explained each component of Framework. Check this post to learn more about explain automation framework to the interviewer.

9. Why do you prefer Selenium Automation Tool?

1. Free and open source
2. Have large user base and helping communities
3. Cross-browser compatibility
4. Platform compatibility
5. Multiple programming languages support

10. What is Selenium?

Selenium is an open source (free) automated testing suite to test web applications. It supports different platforms and browsers. It has gained a lot of popularity in terms of web-based automated testing and giving a great competition to the famous commercial tool HP QTP (Quick Test Professional) AKA HP UFT (Unified Functional Testing).
Selenium is a set of different software tools. Each tool has a different approach in supporting web based automation testing. It has four components namely,

1. Selenium IDE (Selenium Integrated Development Environment)
2. Selenium RC (Selenium Remote Control)
3. Selenium WebDriver
4. Selenium Grid

11. What is Selenium IDE?

Selenium IDE (Integrated Development Environment) is a Firefox plugin. It is the simplest framework in the Selenium Suite. It allows us to record and playback the scripts. Even though we can create scripts using Selenium IDE, we need to use Selenium RC or Selenium WebDriver to write more advanced and robust test cases.

12. What is Selenese?

Selenese is the language which is used to write test scripts in Selenium IDE.

13. Which is the only browser that supports Selenium IDE to be used?

Firefox

14. What is Selenium RC?

Selenium RC AKA Selenium Remote control / Selenium 1. Selenium Remote Control was the main Selenium project for a long time before the WebDriver merge brought up Selenium 2. Selenium 1 is still actively supported (in maintenance mode). It relies on JavaScript for automation. It supports Java, Javascript, Ruby, PHP, Python, Perl and C#. It supports almost every browser out there.

15. What is Selenium WebDriver?

Selenium WebDriver AKA Selenium 2 is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it. Selenium WebDriver supports Java, C#, PHP, Python, Perl, Ruby.

16. What is Selenium Grid?

Selenium Grid is a tool used together with Selenium RC to run tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and operating systems. In simple words, it is used to distribute your test execution on multiple platforms and environments concurrently.

17. When do you use Selenium Grid?

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution

18. What are the advantages of Selenium Grid?

It allows running test cases in parallel thereby saving test execution time.
It allows multi-browser testing
It allows us to execute test cases on multi-platform

19. What is a hub in Selenium Grid?

A hub is a server or a central point that controls the test executions on different machines.

20. What is a node in Selenium Grid?

Node is the machine which is attached to the hub. There can be multiple nodes in Selenium Grid.

21. What are the types of WebDriver APIs available in Selenium?

- Firefox Driver
- Gecko Driver

- InternetExplorer Driver
- Chrome Driver
- HTML Driver
- Opera Driver
- Safari Driver
- Android Driver
- iPhone Driver
- EventFiringWebDriver

22. Which WebDriver implementation claims to be the fastest?

The fastest implementation of WebDriver is the HTMLUnitDriver. It is because the HTMLUnitDriver does not execute tests in the browser.

23. What are the Programming Languages supported by Selenium WebDiver?

- Java
- C#
- Python
- Ruby
- Perl
- PHP

24. What are the Operating Systems supported by Selenium WebDriver?

- Windows
- Linux
- Apple

25. What are the Open-source Frameworks supported by Selenium WebDriver?

- JUnit
- TestNG

Read: TestNG Complete Tutorial

26. What are the Locators available in Selenium?

Different types of locators are:

1. ID – Practical example
2. ClassName – Practical example
3. Name – Practical example
4. TagName – Practical example
5. LinkText – Practical example
6. PartialLinkText – Practical example
7. XPath – Practical example
8. CSS Selector – Practical example

Click here to see the detailed post on Locators.

27. What is an XPath?

XPath is used to locate the elements. Using XPath, we could navigate through elements and attributes in an XML document to locate web elements such as textbox, button, checkbox, Image etc., in a web page.

Learn How To Write Dynamic XPath

28. What is the difference between "/" and "//"

Single Slash "/" – Single slash is used to create XPath with absolute path i.e. the XPath would be created to start selection from the document node/start node.

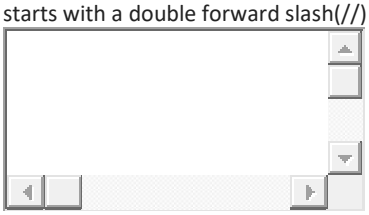
Double Slash "//" – Double slash is used to create XPath with relative path i.e. the XPath would be created to start selection from anywhere within the document.

29. What is the difference between Absolute Path and Relative Path?

Absolute XPath starts from the root node and ends with desired descendant element's node. It starts with top HTML node and ends with input node. It starts with a single forward slash(/) as shown below.

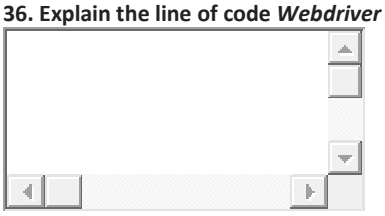
```
1 /html/body/div[3]/div[1]/form/table/tbody/tr[1]/td/input
```

Relative XPath starts from any node in between the HTML page to the current element's node(last node of the element). It starts with a double forward slash(//) as shown below.



```
1 //input[@id='email']
```

30. What is the difference between Assert and Verify in Selenium?
Assert: In simple words, if the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.
Verify: In simple words, there won't be any halt in the test execution even though the verify condition is true or false.
For detailed post check the below link.
Assert Vs. Verify
31. What are Soft Assert and Hard Assert in Selenium?
Soft Assert: Soft Assert collects errors during *@Test* Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.
Hard Assert: Hard Assert throws an *AssertException*immediately when an assert statement fails and test suite continues with next *@Test*
Detailed Post: Soft Assert
32. What are the verification points available in Selenium?
In Selenium IDE, we use Selenese Verify and Assert Commands as Verification points
In Selenium WebDriver, there is no built-in features for verification points. It totally depends on our coding style. some of the Verification points are
To check for page title
To check for certain text
To check for certain element (text box, button, drop down, etc.)
33. How to launch a browser using Selenium WebDriver?
WebDriver is an Interface. We create an Object of a required driver class such as FirefoxDriver, ChromeDriver, InternetExplorerDriver etc.,
To launch Firefox Driver:
WebDriver driver = new FirefoxDriver();
To launch Chrome Driver:
WebDriver driver = new ChromeDriver();
To launch Internet Explorer Driver:
WebDriver driver = new InternetExplorerDriver();
34. Is the FirefoxDriver a Class or an Interface?
FirefoxDriver is a Java class, and it implements the *WebDriver* interface.
35. What is the super interface of WebDriver?
SearchContext.
36. Explain the line of code *Webdriver driver = new FirefoxDriver();* ?



```
1 Webdriver driver = new FirefoxDriver();
```

'*WebDriver*' is an interface and we are creating an object of type WebDriver instantiating an object of FirefoxDriver class.
37. We do create a reference variable 'driver' of type WebDriver

```
1 WebDriver driver = new FirefoxDriver();

2

3 instead of creating

4

5 FirefoxDriver driver = new FirefoxDriver();
```

What is the purpose of doing this way?
If we create a reference variable driver of type WebDriver then we could use the same driver variable to work with any browser of our choice such as IEDriver, SafariDriver etc.,

38. What are the different exceptions you have faced in Selenium WebDriver?
Some of the exceptions I have faced in my current project are

1. ElementNotVisibleException
2. StaleElementReferenceException

Element Not visible Exception:
This exception will be thrown when you are trying to locate a particular element on webpage that is not currently visible eventhough it is present in the DOM. Also sometimes, if you are trying to locate an element with the xpath which associates with two or more element.

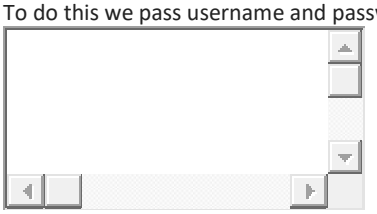
Stale Element Reference Exception:
A stale element reference exception is thrown in one of two cases, the first being more common than the second.
The two reasons for Stale element reference are

1. The element has been deleted entirely.
2. The element is no longer attached to the DOM.

We face this stale element reference exception when the element we are interacting is destroyed and then recreated again. When this happens the reference of the element in the DOM becomes stale. Hence we are not able to get the reference to the element.
Some other exceptions we usually face are as follows:

- WebDriverException
- TimeoutException
- NoAlertPresentException
- NoSuchWindowException
- NoSuchElementException
- TimeoutException

39. How To Login Into Any Site If It Is Showing Any Authentication Pop-Up For Username And Password?
To do this we pass username and password with the URL



```
1 http://username:password@url

2 e.g. http://myUserName:myPassword@softwaretestingmaterial.com
```

40. What are the types of waits available in Selenium WebDriver?
In Selenium we could see three types of waits such as Implicit Waits, Explicit Waits and Fluent Waits.

- Implicit Waits – Click to view detailed post
- Explicit Waits – Click to view detailed post
- Fluent Waits – Click to view detailed post

41. What is Implicit Wait In Selenium WebDriver?

Implicit waits tell to the WebDriver to wait for a certain amount of time before it throws an exception. Once we set the time, WebDriver will wait for the element based on the time we set before it throws an exception. The default setting is 0 (zero). We need to set some wait time to make WebDriver to wait for the required time.

Practical example

42. What is WebDriver Wait In Selenium WebDriver?

WebDriverWait is applied on a certain element with defined *expected condition* and *time*. This wait is only applied to the specified element. This wait can also throw an exception when an element is not found.

Practical example

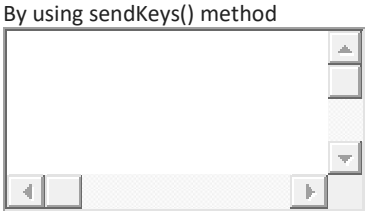
43. What is Fluent Wait In Selenium WebDriver?

FluentWait can define the maximum amount of time to wait for a specific condition and frequency with which to check the condition before throwing an "*ElementNotVisibleException*" exception.

Practical example

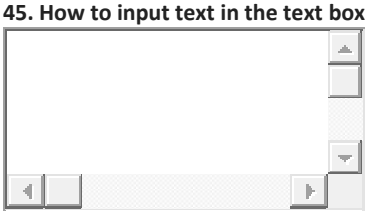
44. How to input text in the text box using Selenium WebDriver?

By using sendKeys() method



```
1 WebDriver driver = new FirefoxDriver();

2 driver.get("https://www.gmail.com");

3 driver.findElement(By.xpath("xpath")).sendKeys("Software Testing Material Website");
```

45. How to input text in the text box without calling the sendKeys()?



```
1 // To initialize js object

2 JavascriptExecutor JS = (JavascriptExecutor)webdriver;

3 // To enter username

4 JS.executeScript("document.getElementById(User').value='SoftwareTestingMaterial.com'");

5 // To enter password

6 JS.executeScript("document.getElementById('Pass').value='tester'");
```

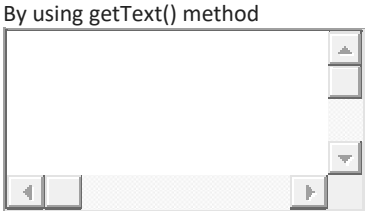
46. How to clear the text in the text box using Selenium WebDriver?

By using clear() method

```
1 WebDriver driver = new FirefoxDriver();

2 driver.get("https://www.gmail.com");

3 driver.findElement(By.xpath("xpath_of_element1")).sendKeys("Software Testing Material Website");

4 driver.findElement(By.xpath("xpath_of_element1")).clear();
```

47. How to get a text of a web element?

By using getText() method



```
    package softwareTestingMaterial;

    import org.openqa.selenium.By;
4
    import org.openqa.selenium.WebDriver;
5
    import org.openqa.selenium.chrome.ChromeDriver;
6
    import org.testng.annotations.Test;
7
    public class TestTestTest {
8


9
     @Test
10
     public void testmethod(){
11
    System.setProperty("webdriver.chrome.driver", "D:\\Selenium Environment\\Drivers\\chromedriver.exe");
12
                WebDriver driver = new ChromeDriver();
13
                driver.get("https://www.google.com");
14
                String availableText = driver.findElement(By.xpath("//*[@id='gbw']/div/div/div[1]/div[1]/a")).getText();
15
                System.out.println("Text Available is :"+availableText);
16
     }

    }
```

48. How to get an attribute value using Selenium WebDriver?

By using getAttribute(value);

It returns the value of the attribute passed as a parameter.

HTML:

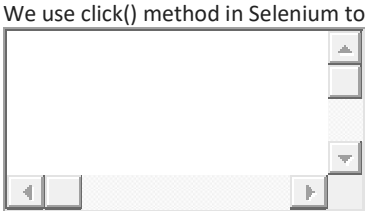
```
1 <input name="nameSelenium" value="valueSelenium">SoftwareTestingMaterial</input>
```

Selenium Code:

```
1 String attributeValue = driver.findElement(By.name("nameSelenium")).getAttribute("value");

2 System.out.println("Available attribute value is :"+attributeValue);

3 Output: valueSelenium
```

49. How to click on a hyperlink using Selenium WebDriver?

We use click() method in Selenium to click on the hyperlink



```
1 driver.findElement(By.linkText("Software Testing Material Website")).click();
```

50. How to submit a form using Selenium WebDriver?

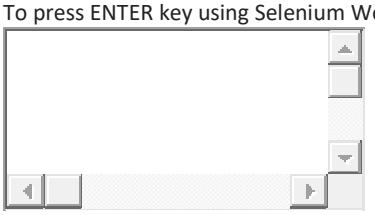
We use "submit" method on element to submit a form

```
1 driver.findElement(By.id("form_1")).submit();
```

Alternatively, you can use click method on the element which does form submission

51. How to press ENTER key on text box In Selenium WebDriver?

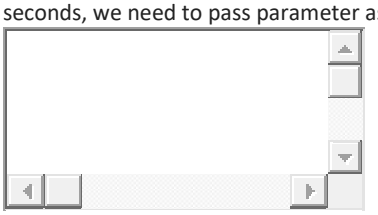
To press ENTER key using Selenium WebDriver, We need to use Selenium Enum Keys with its constant ENTER.



```
1 driver.findElement(By.xpath("xpath")).sendKeys(Keys.ENTER);
```

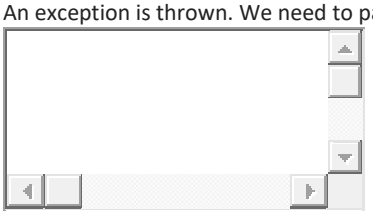
52. How to pause a test execution for 5 seconds at a specific point?

By using **java.lang.Thread.sleep***(long milliseconds)* method we could pause the execution for a specific time. To pause 5 seconds, we need to pass parameter as 5000 (5 seconds)



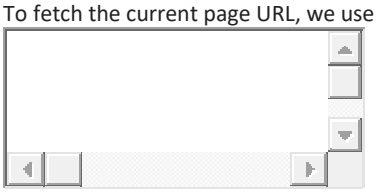
```
1 Thread.sleep(5000)
```

53. Is Selenium Server needed to run Selenium WebDriver Scripts?
When we are distributing our Selenium WebDriver scripts to execute using Selenium Grid, we need to use Selenium Server.
54. What happens if I run this command. *driver.get("www.softwaretestingmaterial.com") ;*
An exception is thrown. We need to pass HTTP protocol within driver.get() method.



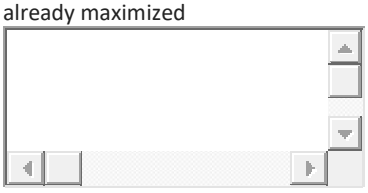
```
1 driver.get("https://www.softwaretestingmaterial.com");
```

55. What is the alternative to *driver.get()* method to open an URL using Selenium WebDriver?
Alternative method to *driver.get("url")* method is *driver.navigate.to("url")*
56. What is the difference between *driver.get()* and*driver.navigate.to("url")*?
driver.get(): To open an URL and it will wait till the whole page gets loaded
driver.navigate.get(): To navigate to an URL and It will not wait till the whole page gets loaded
57. Can I navigate back and forth in a browser in Selenium WebDriver?
We use Navigate interface to do navigate back and forth in a browser. It has methods to move back, forward as well as to refresh a page.
driver.navigate().forward(); – to navigate to the next web page with reference to the browser's history
driver.navigate().back(); – takes back to the previous webpage with reference to the browser's history
driver.navigate().refresh(); – to refresh the current web page thereby reloading all the web elements
driver.navigate().to("url"); – to launch a new web browser window and navigate to the specified URL
58. What are the different types of navigation commands?
Refer above question (Can I navigate back and forth in a browser)
59. How to fetch the current page URL in Selenium?
To fetch the current page URL, we use *getCurrentURL()*



```
1 driver.getCurrentUrl();
```

60. How can we maximize browser window in Selenium?
To maximize browser window in selenium we use *maximize()* method. This method maximizes the current window if it is not already maximized



```
1 driver.manage().window().maximize();
```

61. How to delete cookies in Selenium?

To delete cookies we use *deleteAllCookies()* method

```
1 driver.manage().deleteAllCookies();
```

62. What are the ways to refresh a browser using Selenium WebDriver?

There are multiple ways to refresh a page in selenium

- Using *driver.navigate().refresh()* command as mentioned in the question 45
- Using driver.get("URL") on the current URL or using *driver.getCurrentUrl()*
- Using driver.navigate().to("URL") on the current URL or *driver.navigate().to(driver.getCurrentUrl());*
- Using *sendKeys(Keys.F5)* on any textbox on the webpage

63. What is the difference between driver.getWindowHandle() and driver.getWindowHandles() in Selenium WebDriver?

driver.getWindowHandle() – It returns a handle of the current page (a unique identifier)

driver.getWindowHandles() – It returns a set of handles of the all the pages available.

64. What is the difference between driver.close() and driver.quit() methods?

Purpose of these two methods (driver.close and driver.quit) is almost same. Both allow us to close a browser but still, there is a difference.

driver.close(): To close current WebDriver instance

driver.quit(): To close all the opened WebDriver instances

65. What is the difference between driver.findElement() and driver.findElements() commands?

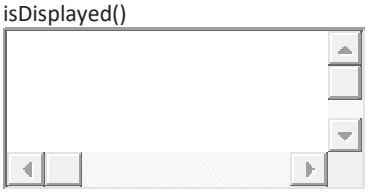
The difference between driver.findElement() and driver.findElements() commands is-

- findElement() returns a single WebElement (found first) based on the locator passed as parameter. Whereas findElements() returns a list of WebElements, all satisfying the locator value passed.
- Syntax of findElement()-
  WebElement textbox = driver.findElement(By.id("textBoxLocator"));
  Syntax of findElements()-
  List <WebElement> elements = element.findElements(By.id("value"));
- Another difference between the two is- if no element is found then findElement() throws NoSuchElementException whereas findElements() returns a list of 0 elements.

66. How to find whether an element is displayed on the web page?

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. isDisplayed()



```
1 boolean elePresent = driver.findElement(By.xpath("xpath")).isDisplayed();
```

2. isSelected()

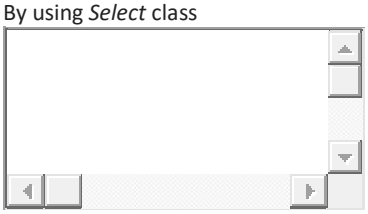
```
1 boolean eleSelected= driver.findElement(By.xpath("xpath")).isSelected();
```

3. isEnabled()

```
1 boolean eleEnabled= driver.findElement(By.xpath("xpath")).isEnabled();
```

67. How to select a value in a dropdown?

By using *Select* class



```
1 WebElement mySelectElement = driver.findElement(By.name("dropdown"));

2 Select dropdown = new Select(mySelectElement);

3 dropdown.selectByVisibleText(Text);

4 dropdown.selectByIndex(Index);

5 dropdown.selectByValue(Value);
```

Practical Example:

68. How to capture Screenshot in Selenium WebDriver?

Test cases may fail while executing the test scripts. While we are executing the test cases manually we just take a screenshot and place in a result repository. The same can be done by using Selenium WebDriver.

Some of the scenarios we may need to capture a screenshot using Selenium WebDriver are

i. Application issues

ii. Assertion Failure

iii. Difficulty to find Webelements on the web page

iv. Timeout to find Webelements on the web page

Selenium provides an interface called *TakesScreenshot*which has a method *getScreenShotAs* which can be used to take a screenshot of the application under test.

In Selenium 3, we may face few issues while capturing Screenshots. To overcome we use aShot utility. Click on below links to see posts related to the normal way of capturing a screenshot and capturing a screenshot using aShot utility.

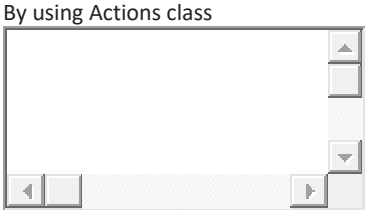
Capture screenshot using Selenium WebDriver

Full Page Screenshot using aShot utility

Failed Test Cases Screenshot

69. How to mouse hover on a web element using WebDriver?

By using Actions class



```
1 WebElement ele = driver.findElement(By.xpath("xpath"));

2 //Create object 'action' of an Actions class

3 Actions action = new Actions(driver);
```

```
4 //Mouseover on an element

5 action.moveToElement(ele).perform();
```

Practical Example.

70. How can we handle web based pop-up?

To handle alerts popups we need to do switch to the alert window and call Selenium WebDriver Alert API methods.

Practical Example.

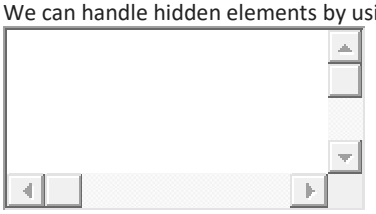
71. How can we handle windows based pop up?

Selenium doesn't support windows based applications. It is an automation testing tool which supports only web application testing. We could handle windows based popups in Selenium using some third party tools such as AutoIT, Robot class etc.

72. How to handle hidden elements in Selenium WebDriver?

It is one of the most important selenium interview questions.

We can handle hidden elements by using javaScript executor



```
1 (JavascriptExecutor(driver)).executeScript("document.getElementsByClassName(ElementLocator).click();");
```

73. How can you find Broken Links in a page using Selenium WebDriver?

Practical Example.

74. How to find more than one web element in the list?

```
1  // To store the list

2  List <WebElement> eleList = driver.findElements(By.xpath("xpath"));

3  // To fetch the size of the list

4  int listSize = eleList.size();

5  //for loop

6  for (int i=0; i<listSize; i++)

7  {

8   // Clicking on each link

9   links.get(i).click();

10  // Navigating back to the previous page that stores the links

11  driver.navigate().back();

12 }
```

75. How to read a JavaScript variable in Selenium WebDriver?

By using JavascriptExecutor

```
1 // To initialize the JS object.

2 JavascriptExecutor JS = (JavascriptExecutor) webdriver;

3 // To get the site title.

4 String title = (String)JS.executeScript("return document.title");

5 System.out.println("Title of the webpage : " + title);
```

76. What is JavaScriptExecutor and in which cases JavaScriptExecutor will help in Selenium automation?

In general, we click on an element using click() method in Selenium.

For example:

```
1 driver.findElement(By.id("Id Value")).click();
```

Sometimes web controls don't react well against selenium commands and we may face issues with the above statement (click()). To overcome such kind of situation, we use JavaScriptExecutor interface.

It provides a mechanism to execute Javascript through Selenium driver. It provides "executescript" & "executeAsyncScript" methods, to run JavaScript in the context of the currently selected frame or window.

There is no need to write a separate script to execute JavaScript within the browser using Selenium WebDriver script. Just we use predefined interface named 'Java Script Executor'. We need to import the JavascriptExecutor package in the script.

Package:

```
1 import org.openqa.selenium.JavascriptExecutor;
```

Syntax:

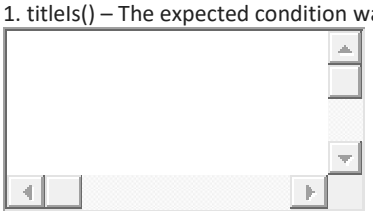
```
1 JavascriptExecutor js = (JavascriptExecutor) driver;

2 js.executeScript(Script,Arguments);
```

Script *– The JavaScript to execute*
Arguments *– The arguments to the script(Optional). May be empty.*
Returns *– One of Boolean, Long, String, List, WebElement, or null.*
Let's see some scenarios we could handle using this Interface:
1. To type Text in Selenium WebDriver without using sendKeys() method
2. To click a Button in Selenium WebDriver using JavaScript
3. To handle Checkbox
4. To generate Alert Pop window in selenium
5. To refresh browser window using Javascript
6. To get innertext of the entire webpage in Selenium
7. To get the Title of our webpage
8. To get the domain
9. To get the URL of a webpage
10. To perform Scroll on an application using Selenium
11. To click on a SubMenu which is only visible on mouse hover on Menu
12. To navigate to different page using Javascript

77. How do you read test data from excels?
Test data can efficiently be read from excel using JXL or POI API. POI API has many advantages than JXL.
Click here to see a practical example of using Apache POI.

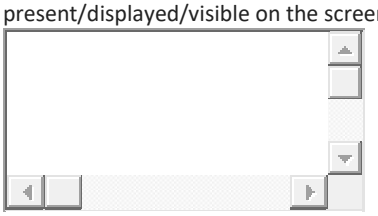
78. Is it possible to automate the captcha using Selenium?
No, It's not possible to automate captcha and bar code reader.

79. How to handle Ajax calls in Selenium WebDriver?
Handling AJAX calls is one of the common issues when using Selenium WebDriver. We wouldn't know when the AJAX call would get completed and the page has been updated. In this post, we see how to handle AJAX calls using Selenium.
AJAX stands for Asynchronous JavaScript and XML. AJAX allows the web page to retrieve small amounts of data from the server without reloading the entire page. AJAX sends HTTP requests from the client to server and then process the server's response without reloading the entire page. To handle AJAX controls, wait commands may not work. It's just because the actual page is not going to refresh.
When you click on a submit button, the required information may appear on the web page without refreshing the browser. Sometimes it may load in a second and sometimes it may take longer. We have no control over loading time. The best approach to handle this kind of situations in selenium is to use dynamic waits (i.e. WebDriverWait in combination with ExpectedCondition)
Some of the methods which are available are as follows:
1. titleIs() – The expected condition waits for a page with a specific title.



```
1 wait.until(ExpectedConditions.titleIs("Deal of the Day"));
```

2. elementToBeClickable() – The expected condition waits for an element to be clickable i.e. it should be present/displayed/visible on the screen as well as enabled.

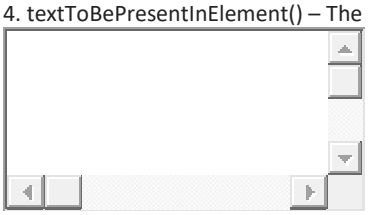


```
1 wait.until(ExpectedConditions.elementToBeClickable(By.xpath("xpath")));
```

3. alertIsPresent() – The expected condition waits for an alert box to appear.

```
1 wait.until(ExpectedConditions.alertIsPresent()) !=null);
```

4. textToBePresentInElement() – The expected condition waits for an element having a certain string pattern.



```
1 wait.until(ExpectedConditions.textToBePresentInElement(By.id("title'"), "text to be found"));
```

80. List some scenarios which we cannot automate using Selenium WebDriver?

1. Bitmap comparison Is not possible using Selenium WebDriver
2. Automating Captcha is not possible using Selenium WebDriver
3. We can not read bar code using Selenium WebDriver

81. What is Object Repository in Selenium WebDriver?

Object Repository is used to store element locator values in a centralized location instead of hard coding them within the scripts. We do create a property file (.*properties*) to store all the element locators and these property files act as an object repository in Selenium WebDriver.

82. How you build Object Repository in your project?

In QTP, there is an Object Repository concept. When a user records a test, the objects and its properties are captured by default in an Object Repository. QTP uses this Object Repository to play back the scripts. Coming to Selenium, there is no default Object Repository concept. It doesn't mean that there is no Object Repository in Selenium. Even though there is no default one still we could create our own. In Selenium, we call objects as locators (such as ID, Name, Class Name, Tag Name, Link Text, Partial Link Text, XPath, and CSS). Object repository is a collection of objects. One of the ways to create Object Repository is to place all the locators in a separate file (i.e., properties file). But the best way is to use Page Object Model. In the Page Object Model Design Pattern, each web page is represented as a class. All the objects related to a particular page of a web application are stored in a class.

83. What is page object model in Selenium?

Page Object Model is a Design Pattern which has become popular in Selenium Test Automation. It is widely used design pattern in Selenium for enhancing test maintenance and reducing code duplication. Page object model (POM) can be used in any kind of framework such as modular, data-driven, keyword driven, hybrid framework etc. A page object is an object-oriented class that serves as an interface to a page of your Application Under Test(AUT). The tests then use the methods of this page object class whenever they need to interact with the User Interface (UI) of that page. The benefit is that if the UI changes for the page, the tests themselves don't need to change, only the code within the page object needs to change. Subsequently, all changes to support that new UI is located in one place.

84. What is page factory?

We have seen that 'Page Object Model' is a way of representing an application in a test framework. For every 'page' in the application, we create a Page Object to reference the 'page' whereas a 'Page Factory' is one way of implementing the 'Page Object Model'.

85. What is the difference between Page Object Model (POM) and Page Factory?

Page Object is a class that represents a web page and hold the functionality and members.
Page Factory is a way to initialize the web elements you want to interact with within the page object when you create an instance of it.

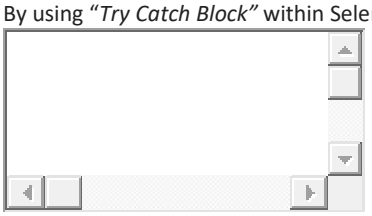
86. What are the advantages of Page Object Model Framework?

Code reusability – We could achieve code reusability by writing the code once and use it in different tests.
Code maintainability – There is a clean separation between test code and page specific code such as locators and layout which becomes very easy to maintain code. Code changes only on Page Object Classes when a UI change occurs. It enhances test maintenance and reduces code duplication.
Object Repository – Each page will be defined as a java class. All the fields in the page will be defined in an interface as members. The class will then implement the interface.

Readability – Improves readability due to clean separation between test code and page specific code

87. How can you use the Recovery Scenario in Selenium WebDriver?

By using *"Try Catch Block"* within Selenium WebDriver Java tests.



```
1 try {

2     driver.get("www.SoftwareTestingMaterial.com");

3 }catch(Exception e){

4     System.out.println(e.getMessage());

5 }
```

88. How to Upload a file in Selenium WebDriver?

There are two cases which are majorly used to upload a file in Selenium WebDriver such as using *SendKeys* Method and using *AutoIT* Script.

Practical Example.

89. How to Download a file in Selenium WebDriver?

By using *AutoIT* script, we could download a file in Selenium WebDriver.

Practical Example

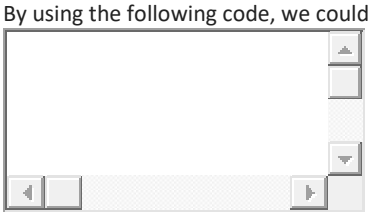
90. How to run Selenium WebDriver Test from the command line?

Run Java Program using Command Prompt

Run TestNG using Command Prompt

91. How to switch between frames in Selenium?

By using the following code, we could switch between frames.



```
1 driver.switchTo().frame();
```

92. How to connect a Database in selenium?

As we all know Selenium WebDriver is a tool to automate User Interface. We could only interact with Browser using Selenium WebDriver.

We use JDBC Driver to connect the Database in Selenium (While using Java Programming Language).

Practical Example

93. How To Resize Browser Window Using Selenium WebDriver?

To resize the browser window to particular dimensions, we use 'Dimension' class to resize the browser window.

Practical Example

94. How To Scroll Web Page Down Or UP Using Selenium WebDriver?

JavaScript **scrollBy**() method scrolls the document by the specified number of pixels.

Practical Example

95. How To Perform Right Click Action (Context Click) In Selenium WebDriver?

We use Actions class in Selenium WebDriver to do Right-Click (Context Click) action.

Practical Example

96. How To Perform Double Click Action In Selenium WebDriver?

We use Actions class to do Double click action in selenium.

Practical Example

97. How To Perform Drag And Drop Action in Selenium WebDriver?

In some applications, we may face a situation to automate drag and drop an item from one location to another location. We could not achieve these using basic elements. Selenium has provided an “Actions” class to handle this kind of scenarios. We overcome this kind of scenarios such as drag and drop using Actions Class.
To achieve this, we use Actions class in Selenium WebDriver.
Practical Example

98. How To Highlight Element Using Selenium WebDriver?

By using JavascriptExecutor interface, we could highlight the specified element
Practical Example

99. Have you used any crossbrowsertesting tool to run selenium scripts on cloud?

I have used BrowserStack to run selenium tests on multiple browsers & Multiple operating systems in parallel. Earlier we have made a video on how to use BrowserStack to run selenium scripts on the cloud. Find the link in the description below.

100. What is desired capabilities?

In Selenium we use desired capabilities to handle SSL certificates in chrome browser
We need to create an instance of DesiredCapabilities

```
1 DesiredCapabilities desiredCapability = DesiredCapabilities.chrome();
```

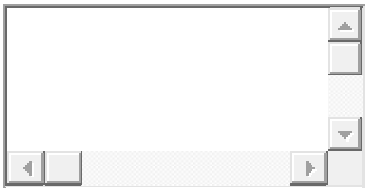
101. What is Continuous Integration?

Continuous Integration is abbreviated as CI. Continuous Integration is a development practice which aims to make sure the correctness of a software. After each commit, a suite of tests run automatically and test the software to ensure whether the software is running without any breaks. If any test fails, we will get immediate feedback say “build is broken”.
In simple words, continuous integration is a process of verifying the correctness of a software.
Some of the continuous integration tools are Jenkins, TeamCity, Bamboo, Travis, Circle Ci, Bitbucket.
We can schedule the test suite execution using these CI Tools.

102. How to achieve Database testing in Selenium?

As we all know Selenium WebDriver is a tool to automate User Interface. We could only interact with Browser using Selenium WebDriver.
Sometimes, we may face a situation to get the data from the Database or to modify (update/delete) the data from the Database. If we plan to automate anything outside the vicinity of a browser, then we need to use other tools to achieve our task. To achieve the Database connection and work on it, we need to use JDBC API Driver.
The Java Database Connectivity (JDBC) API provides universal data access from the Java programming language. Using the JDBC API, you can access virtually any data source, from relational databases to spreadsheets and flat files. It lets the user connect and interact with the Database and fetch the data based on the queries we use in the automation script. JDBC is a SQL level API that allows us to execute SQL statements. It creates a connectivity between Java Programming Language and the database.
Using JDBC Driver we could do the following
i. Establish a Database connection
ii. Send SQL Queries to the Database
iii. Process the results

103. How to delete Browser Cookies with Selenium Web Driver?



```
1 driver.Manage().Cookies.DeleteAllCookies();
```

TestNG – Interview Questions:
Here we have dealt with some important TestNG interview questions. If you want to learn more interview questions related to TestNG then here you go. We have a special post on TestNG Interview Questions. Also, you could find TestNG Complete Tutorial here

104. What is TestNG?

TestNG is a testing framework designed to simplify a broad range of testing needs, from unit testing to integration testing.

105. What are the annotations available in TestNG?

@BeforeTest
@AfterTest
@BeforeClass
@AfterClass
@BeforeMethod
@AfterMethod
@BeforeSuite
@AfterSuite
@BeforeGroups
@AfterGroups
@Test

106. What is TestNG Assert and list out some common Assertions supported by TestNG?

TestNG Asserts help us to verify the condition of the test in the middle of the test run. Based on the TestNG Assertions, we will consider a successful test only if it is completed the test run without throwing any exception.

Some of the common assertions supported by TestNG are

- assertEqual(String actual,String expected)
- assertEqual(String actual,String expected, String message)
- assertEquals(boolean actual,boolean expected)
- assertTrue(condition)
- assertTrue(condition, message)
- assertFalse(condition)
- assertFalse(condition, message)

For Complete Post

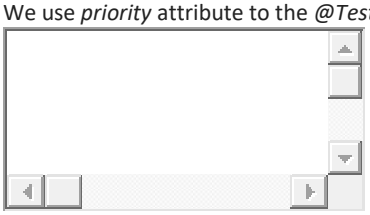
107. How to create and run TestNG.xml?

In TestNG framework, we need to create T**estNG XML** file to create and handle multiple test classes. We do configure our test run, set test dependency, include or exclude any test, method, class or package and set priority etc in the XML file.

For Complete Post

108. How to set test case priority in TestNG?

We use *priority* attribute to the *@Test* annotations. In case priority is not set then the test scripts execute in alphabetical order.



```
1  package TestNG;

2  import org.testng.annotations.*;

3  public class PriorityTestCase{

4   @Test(priority=0)

5   public void testCase1() {

6   system.out.println("Test Case 1");

7   }

8   @Test(priority=1)

9   public void testCase2() {

10  system.out.println("Test Case 2");
```

```
11  }

12 }
```

Output:

```
1 Test Case 1

2 Test Case 2
```

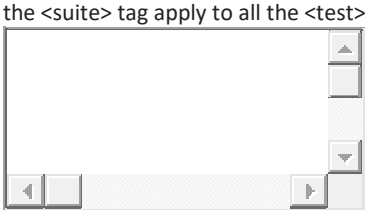
109. What is Parameterized testing in TestNG?

Parameterized tests allow developers to run the same test over and over again using different values.
There are two ways to set these parameters:

- *with testng.xml* - Practical Example
- *with Data Providers* – Practical Example

110. How to run a group of test cases using TestNG?

TestNG allows you to perform sophisticated groupings of test methods. Not only can you declare that methods 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 tests and doesn't require you to recompile anything if you want to run two different sets of tests back to back.
Groups are specified in your testng.xml file and can be found either under the <test> or <suite> tag. Groups specified in the <suite> tag apply to all the <test> tags underneath.



```
1 @Test (groups = { "smokeTest", "functionalTest" })

2 public void loginTest(){

3 System.out.println("Logged in successfully");

4 }
```

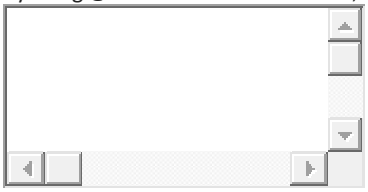
View Complete Post

111. What is the use of @Listener annotation in TestNG?

Ans. TestNG listeners are used to configure reports and logging. One of the most widely used listeners in TestNG is *ITestListener* interface. It has methods like *onTestStart, onTestSuccess*, *onTestFailure*, *onTestSkipped* etc. We should implement this interface creating a listener class of our own. Next, we should add the listeners annotation (*@Listeners*) in the Class which was created.
Practical Example

112. How can we create a data-driven framework using TestNG?

By using *@DataProvider* annotation, we can create a Data Driven Testing Framework.



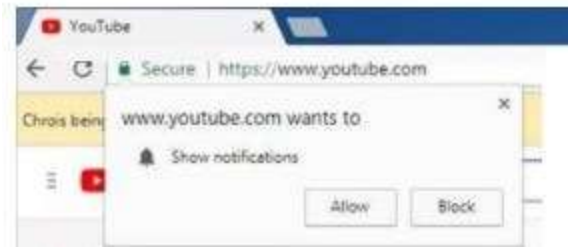
```
1  @DataProvider(name="getData")
2   public Object[][] getData(){
3   //Object [][] data = new Object [rowCount][colCount];
4   Object [][] data = new Object [2][2];
5
6   data [0][0] = "FirstUid";
7   data [0][1] = "FirstPWD";
8
9   data[1][0] = "SecondUid";
10  data[1][1] = "SecondPWD";
11
12  return data;
13
14  }
```

Practical Example

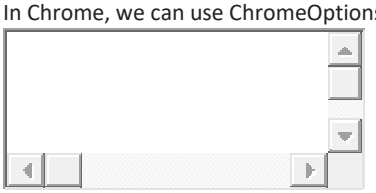
113. Where you have applied OOPS in Automation Framework?

Check this link

114. How to handle browser (chrome) notifications in Selenium?



In Chrome, we can use ChromeOptions as shown below.



```
1 ChromeOptions options = new ChromeOptions();
2 options.addArguments("disable-infobars");
3 WebDriver player = new ChromeDriver(options);
```

1. **What is Stale Element Exception? How to handle it?**

· A WebElement is a reference to an element in the DOM.

- A StaleElementException is thrown when the element you were interacting is destroyed and then recreated. Most complex web pages these days will move things about on the fly as the user interacts with it and this requires elements in the DOM to be destroyed and recreated.
- When this happens the reference to the element in the DOM that you previously had becomes stale and you are no longer able to use this reference to interact with the element in the DOM. When this happens you will need to refresh your reference, or in real world terms find the element again
- We can use Actions class to click on specific element where this excpetion is coming up

2. What are different Locators that are used by WebDriver?

- ID
- Name
- Xpath
- Tag Name
- CSS Selector
- Partial Link Text
- Class Name

3. What are different Xpath functions that you have used in your Project?

- Relative Path
- Contains
- By Attribute name and value
- Parent to Child and Vice Versa relationship
- Following-sibling
- Preceeding-sibling
- And

4. What will happen in background when execute new FirefoxDriver() ?

- Firefox binary will be triggered and FF browser will open with default options.
- FirefoxDriver object is created

5. What is the below statement means and Why?

--- WebDriver driver = new FirefoxDriver();

- WebDriver is an interface which contain several abstract methods such as get(...), findElamentBy(...) etc.
- We simply create reference of web Driver and we can assign objects (Firefox driver, CromeDriver, IEDriver, Andriod driver etc) to it.

6. How do you handle inner Frames and Adjacent Frames?

- SwitchTo frame1, SwitchTo frame2 (inner frame) work on the element and switchto default content
- Use SwitchTo frame to move the control inside frame.

7. How to click on an element which is not visible using selenium WebDriver?

- We can use JavascriptExecutor to click

8. Difference between verify and assert?

- If you want your test execution not to abort when a validation fails then we use Verify otherwise we use Assert.

9. What is the use of @FindByannotation?

- @FindBy is used to identify element in the Page Factory approach

10. What are the common exceptions you see in Selenium? How are those handled?

- **NoSuchElementException** :FindBy method can't find the element.

- **StaleElementReferenceException** : This tells that element is no longer appearing on the DOM page.
- **TimeoutException**: This tells that the execution is failed because the command did not complete in enough time.
- **ElementNotVisibleException**: Thrown to indicate that although an element is present on the DOM, it is not visible, and so is not able to be interacted with
- **ElementNotSelectableException**: Thrown to indicate that may be the element is disabled, and so is not able to select.

11. Do you use Thread.sleep?

- Rarely

12. What are different wait mechanisms that can be used in scripting?

- driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
- WebDriverWait
- Thread.sleep(8000);
- FluentWait

13. What are different pop-ups that you have handle in your projects?

- JavaScript Pop Ups --- Alert alert = driver.switchTo().alert();
- Browser Pop Ups --- Browser Profiles, Robot tool Kit, AutoIT
- Native OS Pop Ups--- Browser Profiles, Robot tool Kit, AutoIT

14. What are multiple ways of refreshing the current browser?

- Driver.navigate().refresh();
- SendKeys – Keys.F5
- Driver.get(driver.getCurrentURL)
- Driver.navigate().to(driver.getCurrentURL)

15. How do you handle HTTP Proxy Authentication pop ups in browser?

- Form authentications URL - http://UserName:Password@Example.com

16. How do you handle Ajax dropdowns?

- With help of Selenium Sync commands like ImplicitWait, WebDriverWait or FluentWait

17. What is the difference between findelement() and findelements()?

findelement()

- findELement will find the first matching element.
- findELement will throw NoSuchElementException if no matching element found
- public WebElementfindElement(By by)

findelements()

- findELements will all the matching elements.
- findELements – will return an empty list if no matching elements found and no exception will be thrown
- public java.util.ListfindElements(By by)

18. What are the limitations of web driver?

- It does not support and non-web-based applications; it only supports web based applications.
- It cannot readily support new browsers or new firefox versions
- It is open source tool so in case of any technical issues you need to rely on the selenium community forums to get your issue resolved.
- No inbuilt reporting capability so you need plugins like JUnit and TestNG for test reports.

19. What is the default port for Selenium Grid?

- 4444

20. What is Exit/Done Criteria?

- Test can be executed multiple times and does not fail (unless defect has been found). In a nutshell, the automated test case is not fragile.
- Test is verifying/validating expected results
- Code reviewed and all comments are addressed
- Updates test case execution status in QC/Rally successfully
- The script is not dependent on login credentials/etc as it should be smart enough to randomly select users from the DB. The point here is to try and emulate real-life scenarios.
- NO STATIC INPUT. We want to make each test unique each time it executes as this emulates a real life scenario (e.g. different users interacting with the system). There are some instances where static data/input is required/needed. Use good judment in this area. Normally static input should be used for setup operations (e.g. Test/Dev – environment specifics in your testing.xml file)
- Code committed in SVN

21. Difference between Absolute path & Relative path?

- Absolute path will start with root path (/) and Relative path will from current path (//)
- Absolute
  xPath: /html/body/div[3]/div[2]/div[2]/div[2]/div[2]/div[2]/div[2]/div/div[4]/div[1]/div/div[@id='main']/div[@id='Blog1']/div[1]/div[1]/div/div[1]/div/h3/a
- Relative xPath : //h3/a[text()='Working on New Window']

22. What are desiredcapabilities?

- Desired Capabilities help to set properties for the Web Driver. A typical use case would be to set the path for the Firefox Driver if your local installation doesn't correspond to the default settings.
- https://code.google.com/p/selenium/wiki/DesiredCapabilities

23. Difference between Web driver listener and TestNG Listener?

- TestNG and Web driver Listener have different interfaces to implement and call them. They bothmodify respective behavior. You can use Listeners in Annotation. Below 2 URL gives the detailed list of listener and their interfaces.
- http://testng.org/doc/documentation-main.html#testng-listeners
- http://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/support/events/AbstractWebDriverEventListener.html

24. Describe your framework.

- Explain the framework

24. Which is the best way to locate an element?

- Finding elements by ID is usually going to be the fastest option, because at its root, it eventually calls down to document.getElementById(), which is optimized by many browsers.
- Finding elements by XPath is useful for finding elements using very complex selectors, and is the most flexible selection strategy, but it has the potential to be very slow, particularly in IE. In IE 6, 7, or 8, finding by XPath can be an order of magnitude slower than doing the same in Firefox. IE provides no native XPath-over-HTML solution, so the project must use a JavaScript XPath implementation, and the JavaScript engine in legacy versions of IE really is that much slower.

- If you have a need to find an element using a complex selector, I usually recommend using CSS Selectors, if possible. It's not quite as flexible as XPath, but will cover many of the same cases, without exhibiting the extreme performance penalty on IE that XPath can.

25. What is the difference between "GET" and "NAVIGATE" to open a web page in selenium web driver?

- navigate().to() and get() do exactly the same thing. One's just a lot easier to type than the other!
- The navigate interface also has the ability to move backwards and forwards in your browser's history:

26. Please tell me the difference b/w implicitly Wait and Explicit wait.

- **Implicit Wait** sets internally a timeout that will be used for all consecutive Web Element searches. It will try lookup the element again and again for the specified amount of time before throwing a NoSuchElementException if the element could not have been found. It does only this and can't be forced into anything else - it waits for elements to show up.

- **Explicit Wait** or just Wait is a one-timer used by you for a particular search. It is more extendible in the means that you can set it up to wait for any condition you might like. Usually, you can use some of the prebuilt Expected Conditions to wait for elements to become clickable, visible, invisible, etc., or just write your own condition that suits your needs.

27. How we can retrieve the dynamically changing Ids?

- When we login Facebook the login label's id changes dynamically thus resulting in failure.
- We have a hierarchy of locators and Facebook Is dynamic in nature, so we are not able to use "id" for identification for that we have remaining 7 locator's for that : 2. xpath ().. 3. name..4. css.. 5. link text.. 6. partiallinktext...7.tag name. So u can use any one for identifying it. Most probably u can use "xpath" or "css-locator" and if there r tag then link text or partial-link text. it depend on u . But we never use id's in Ajax application because it's not possible.

28. What is the difference between driver.Close() and driver.Quit () method?

- driver.close() - It is used to close the browser or page currently which is having the focus.
- driver.quit() - It is used to shut down the web driver instance or destroy the web driver instance (Close all the windows)

29. How to scroll web element?(not browser)

```
FirefoxProfile profile=new FirefoxProfile();
profile.setEnableNativeEvents(true);
WebDriver driver=new FirefoxDriver(profile);
driver.navigate("http://jqueryui.com/draggable/");
Thread.sleep(6000L);
WebElement element=driver.findElement(By.xpath("//div[@id='draggable']"));
Actions actn=new Actions(driver);
actn.dragAndDropBy(element, 50, 50).build().perform();
}
```

30. What is the basic use of Firefox profiles and how can we use them using selenium?

- A profile in Firefox is a collection of bookmarks, browser settings, extensions, passwords, and history; in short, all of your personal settings.
- We use them to change user agent, changing default download directory, changing versions etc.
- http://code.google.com/p/selenium/wiki/FirefoxDriver

31. Customize the name of file going to be downloaded?

- You have to download AUTO IT.exe file and has to be install and later you have create .au3 file (in this file you have to specify the commands in VB script like your file name, where have to save, it will be easy may be 3 or 4 steps) using AUTOIT...then right click the .au3 file you have to compile ....after that you will get the .exe file with the name of .au3 file ..In eclipse you will give the code like this
- <----processbuildder .exe="" .start="" au3="" file="" of="" path="" processbuilder="" ps="new" the="">

32. Difference between flex and flash application.

- In flash there is no code just based on creativity(design) we will complete the work(time consuming process) whereas flex contain some small functions which is integrated with mxml, PHP..(no tool is there to develop flex we want to use the properties of css and style sheet)

33. What is Error Collector in TestNG? What is its use?

- This class allows the collection of errors during the process of retrieving thetest data for the test method parameters
- http://testngdatabind.sourceforge.net/apidocs/net/sf/testng/databinding/core/error/ErrorCollector.html

34. How to run tests in multiple browser parallel?

- selenium grid

35. How to prepare Customized html Report using TestNG in hybrid framework.

- Junit: with the help of ANT.
- TestNG: using inbuilt default.html to get the HTML report. Also XST reports from ANT,Selenium, TestNG combination.
- Using our own customized reports using XSL jar for converting XML content to HTML.

36. How to refresh a page without using context click?

- Using sendKeys.Keys method
- Using navigate.refresh() method
- Using navigate.refresh() method
- Using get() method
- Using sendKeys() method

1.UsingsendKeys.Keys method

```
driver.get("https://accounts.google.com/SignUp");
driver.findElement(By.id("firstname-placeholder")).sendKeys(Keys.F5);
```

2.Usingnavigate.refresh() method

```
driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/100-selenium-interview-questions.html");
driver.navigate().refresh();
```

3.Using navigate.to() method

```
driver.get("http://ruchi-myseleniumblog.blogspot.in/2014/01/selenium-hybrid-framework-using.html");
driver.navigate().to(driver.getCurrentUrl());
```

4.Using get() method

```
driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/basic-core-java-interview-questions.html");
driver.get(driver.getCurrentUrl());
```

5.UsingsendKeys() method

```
driver.get("https://accounts.google.com/SignUp");
driver.findElement(By.id("firstname-placeholder")).sendKeys("\uE035");
```

37. How to find broken images in a page using Selenium Web driver.

- Get xpath and then using tag name; get all the links in the page
- Use HttpURLConnector class and sent method GET
- Get the response code for each link and verify if it is 404/500

38. How to disable cookies in browser.

- Using deleteAllVisibleCookies() in selenium

39. How to change user agent in Firefox by selenium web driver.

FirefoxProfile profile = new FirefoxProfile();
profile.setPreference("general.useragent.override", "some UA string");
Web Driver driver = new FirefoxDriver(profile);

40. What is the MOST challenging test problem in my career in Automation?

In my career
Changing XPATHS' between testing server and production server-by keeping generic xpath
Keep separate property files for production and UAT
Automating flash apps

41. How to handle network latency using selenium?

- Using driver.manage.pageLoadingtime for network latency

42. How does u handle dynamic elements without using xpath (with example?)

- By using classname or css.

43. How to work with dynamic web table?

- You can get the total number of tags within a tag by giving the xpath of the element by using this function -List ele = driver.findElements(By.xpath("Xpath of the table"));
- Now you can use a for each loop to loop through each of the tags in the above list and then read each value by using getText() method.

44. What is object repository?

It is collection of object names their properties, attributes and their values. It may be excel, XML, property file or text file

45. How to handle alerts and confirmation boxes. Confirmation boxes and Alerts are handled in same way in selenium.

var alert = driver.switchTo().alert();
alert.dismiss(); //Click Cancel or Close window operation
alert.accept(); //Click OK
Handle Confirmation boxes via JavaScript,
driver.executeScript("window.confirm = function(message){return true;};");

46. How to mouse hover on an element?

Actions action = new Actions(webdriver);
WebElement we = webdriver.findElement(By.xpath("html/body/div[13]/ul/li[4]/a"));
action.moveToElement(we).moveToElement(webdriver.findElement(By.xpath("/expression-here"))).click().build().perform();

47. How to switch between the windows?

```
private void handlingMultipleWindows(String windowTitle) {
        Set windows = driver.getWindowHandles();
for (String window : windows) {
driver.switchTo().window(window);
          if (driver.getTitle().contains(windowTitle)) {  return;  }  }  }
```

48. How to switch between frames?

WebDriver's driver.switchTo().frame() method takes one of the three possible arguments:

A number.

Select a frame by its (zero-based) index. That is, if a page has three frames, the first frame would be at index "0", the second at index "1" and the third at index "2". Once the frame has been selected, all subsequent calls on the WebDriver interface are made to that frame.

A name or ID.

Select a frame by its name or ID. Frames located by matching name attributes are always given precedence over those matched by ID.

A previously found WebElement.

Select a frame using its previously located WebElement.

Get the frame by it's id/name or locate it by driver.findElement() and you'll be good.

49. How to take the screen shots in seelnium2.0?

```
//store screenshots
public static void captureScreenShot(String filePath) {
  File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
try {
FileUtils.copyFile(scrFile, new File(filePath));
  } catch (IOException e) {
   // TODO Auto-generated catch block
e.printStackTrace();
}
}
```

50. What is the default time for selenium Ide and webdriver?

Default timeout in selenium ide is 30 seconds.

For web driver go to below URL:

http://assertselenium.com/2013/01/29/webdriver-wait-commands/

51. Write down scenarios which we can't automate?

Barcode Reader, Captcha etc.

52. Differences between jxl and ApachePOI.

- jxl does not support XLSX files
- jxl exerts less load on memory as compared to ApachePOI
- jxl doesn't support rich text formatting while ApachePOI does.
- jxl has not been maintained properly while ApachePOI is more up to date.
- Sample code on Apache POI is easily available as compare to jxl.

53. Does Selenium support https protocols?

- Yes

54. "I want to find the location of ""b"" in the below code, how can I find out without using xpath, name, id, csslocator, index.

a

b

c

- driver.findElement(By.xpath("//*[contains(text(),'b')]")).click(); or
- //div/button[contains(text(),'b']

55. Name 5 different exceptions you had in selenium web driver and mention what instance you got it and how do you resolve it?

- WebDriverException
- NoAlertPresentException
- NoSuchWindowException
- NoSuchElementException
- TimeoutException

1. WebDriverException

WebDriver Exception comes when we try to perform any action on the non-existing driver.

```
WebDriver driver = new InternetExplorerDriver();
driver.get("http://google.com");
driver.close(); driver.quit();
```

2. NoAlertPresentException

When we try to perform an action i.e., either accept() or dismiss() which is not required at a required place; gives us this exception.

```
try{
driver.switchTo().alert().accept();
}
catch (NoAlertPresentException E){
E.printStackTrace();
}
```

3. NoSuchWindowException

When we try to switch to an window which is not present gives us this exception:

```
WebDriver driver = new InternetExplorerDriver();
driver.get("http://google.com");
driver.switchTo().window("Yup_Fail");
driver.close();
```

In the above snippet, line 3 throws us an exception, as we are trying to switch to an window that is not present.

4. NoSuchFrameException

Similar to Window exception, Frame exception mainly comes during switching between the frames.

```
WebDriver driver = new InternetExplorerDriver();
driver.get("http://google.com");
driver.switchTo().frame("F_fail");
driver.close();
```

In the above snippet, line 3 throws us an exception, as we are trying to switch to an frame that is not present.

5. NoSuchElementException

This exception is thrown when we WebDriver doesn't find the web-element in the DOM.

```
WebDriver driver = new InternetExplorerDriver();
driver.get("http://google.com");
driver.findElement(By.name("fake")).click();
```

6. TimeoutException

Thrown when a command does not complete in enough time.

All the above exceptions were handled using try catch exceptions.

56. How do you manage the code versions in your project?

- Using SVN or other versioning tools

57. What are the test types supported by Selenium?

- Selenium supports UI and functional testing. As well it can support performance testing for reasonable load using selenium grid.

58. In what all case we have to go for "JavaScript executor".

- Consider FB main page after you login. When u scrolls down, the updates get loaded. Tohandle this activity, there is no selenium command. So you can go for javascript to setthe scroll down value like driver.executeScript("window.scrollBy(0,200)", "");

59. What is the difference between data driven and Keyword driven framework?

- Data Driven Framework: Basically this type of Framework we read data from excel/some external sheet.
- Keyword Driven Framework: In the main script keyword drives which workflow/ testcases to execute in generic term

60. What is the difference between Text area and Text field component?

- **Text Area**: in generally like comment area where you are provide like 200 characters etc.
- **Text filed**: where you type predefined text like country name etc.

61. How you will configure TestNG with selenium?

- *help-> new software -> add url*
- http://beust.com/eclipse

62. How to count total no of hyperlinks in a page and click on the 5 th link if exist else display an error message 5th link is not exist.

- Listalllinks=driver,finElementsbytagname("a");
  syso(alllinks.size());

63. There is an airline website from where you can choose source to destination place (e.g Bangalore to Delhi) which gives you 10 different flight details like

A001N
A002N
A003N
A004N
A00234M
A00123M
…
From the above list include the flights which are ends with "M "character and exclude the flights which are ends with "N" character.

- Use xpath functions :- ***endWith()*** to find element ending with N

64. What is the difference between Web driver wait and fluent wait when we need to choose web driver wait and fluent wait?

- Fluent wait / driver wait belongs to explicit wait where it used to wait for Specific element
- Fluent wait: suppress some expectation when element not found, Polling of web element we can control which web driver wait doesn't provide

68. Types of locator in selenium and which locator you used frequently?

- ID, Name, Xpath and CSS Selector

69. How to click on an element which is not visible using selenium WebDriver?

- We can use JavascriptExecutor to click

70. What is the use of @FindBy annotation?

- @FindBy is used to identify element in the Page Factory approach

Selenium Interview Questions
What is Automation Testing?
What are the different types of locators in Selenium?
What is the difference between assert and verify commands?
What is an XPath?
What is the difference between "/" and "//" in Xpath?
What is Same origin policy and how it can be handled?
When should I use Selenium Grid?
What do we mean by Selenium 1 and Selenium 2?
Which is the latest Selenium tool?
How do I launch the browser using WebDriver?
What are the different types of Drivers available in WebDriver?
What are the benefits of Automation Testing?
What are the different types of waits available in WebDriver?
How to type in a textbox using Selenium?
How can you find if an element displayed on the screen?
How can we get a text of a web element?
How to select value in a drop down?
Most Popularly Asked Selenium Interview Questions
What are the different types of navigation commands?
How to click on a hyper link using linkText?
How to handle frame in WebDriver?
When do we use findElement(and findElements(?
How to find more than one web element in the list?
Why should Selenium be selected as a test tool?
What is the difference between driver.close(and driver.quit command?
Can Selenium handle windows based pop up?
How can we handle web based pop up?

How can we handle windows based pop up?
How to assert title of the web page?
How to mouse hover on a web element using WebDriver?
How to retrieve css properties of an element?
How to capture screenshot in WebDriver?
What is Junit?

Top Selenium Interview Questions

What are Junit annotations?
What is Selenium? What are the different Selenium components?
What is TestNG and how is it better than Junit?
How to set test case priority in TestNG?
What is a framework?
What are the advantages of Automation framework?
What are the different types of frameworks?
How can I read test data from excels?
What is the difference between POI and jxl jar?
What is the difference between Selenium and QTP?
Can WebDriver test Mobile applications?
Can captcha be automated?
What are the testing types that can be supported by Selenium?
What is Object Repository? How can we create Object Repository in Selenium?
What are the limitations of Selenium?
What is the difference between Selenium IDE, Selenium RC and WebDriver?
When should I use Selenium IDE?
What is Selenese?
Source: softwaretestinghelp.com
Selenium Webdriver Interview Questions
Explain the difference between single and double slash in X-path? Single slash '/ '
Explain what is assertion in Selenium and what are the types of assertion?
How will you find an element using Selenium?
List out the technical challenges with Selenium? Technical challenges with Selenium are
List out the test types that are supported by Selenium?
Mention what is Selenium 3.0?
Mention what is the use of X-path?
What is Selenium 2.0?
What is Selenium and what is composed of?
Can we use Selenium grid for performance testing?
Can we use Selenium RC to drive tests on two different browsers on one operating system without Selenium Grid?
Explain how Selenium Grid works?
Explain how to assert text of webpage using selenium 2.0?
Explain how to iterate through options in test script?
Explain how you can capture server side log Selenium Server?
Explain how you can debug the tests in Selenium IDE ?
Explain how you can find broken images in a page using Selenium Web driver ?

Essential Selenium Interview Questions

Explain how you can handle colors in web driver?
Explain how you can handle frames using Selenium 2.0?
Explain how you can insert a break point in Selenium IDE ?
Explain how you can insert a start point in Selenium IDE?
Explain how you can login into any site if it's showing any authentication popup for password and username?
Explain how you can switch back from a frame?
Explain how you can switch between frames?
Explain how you can use recovery scenario with Selenium?
Explain in Selenium IDE how can you debug the tests?
Explain using Webdriver how you can perform double click ?
Explain what are the JUnits annotation linked with Selenium?
Explain what are the limitations of Selenium IDE?
Explain what can cause a Selenium IDE test to fail?

Explain what is Datadriven framework and Keyword driven?
Explain what is framework and what are the frameworks available in RC?
Explain what is the difference between Borland Silk and Selenium?

Frequently Asked Selenium Interview Questions

Explain what is the difference between find elements () and find element () ?
Explain what is the main difference between web-driver and RC ?
For Database Testing in Selenium Webdriver what API is required?
From Selenium IDE how you can execute a single line?
From your test script how you can create html test report?
How can we handle pop-ups in RC ?
How can you prepare customized html report using TestNG in hybrid framework ?
How can you retrive the message in an alert box ?
How do you identify an object using selenium?
How Selenium grid hub keeps in touch with RC slave machine?
How will you handle working with multiple windows in Selenium ?
How will you use Selenium to upload a file ?
How will you verify the specific position of an web element
How you can convert any Selenium IDE tests from Selenese to another language?
How you can use “submit” a form using Selenium ?
In Selenium IDE how you can generate random numbers and dates for test data ?
In selenium IDE what are the element locators that can be used to locate elements on web page?
In Selenium what are Breakpoints and Startpoints?
In which format does source view shows your script in Selenium IDE ?
List out different types of locators?
Selenium Real Time Interview Questions
List the advantages of Webdriver over Selenium Server?
Mention different exceptions you had in Selenium web driver?
Mention in what ways you can customize TestNG report?
Mention what are the advantages of Using Git Hub For Selenium?
Mention what are the capabilities of Selenium WebDriver or Selenium 2.0?
Mention what are the challenges in Handling Ajax Call in Selenium Webdriver?
Mention what are the types of Listeners in TestNG?
Mention what is desired capability? How is it useful in terms of Selenium?
Mention what is IntelliJ?
Mention what is Listeners in Selenium WebDriver?
Mention what is the difference between Implicit wait and Explicit wait?
Mention when to use AutoIT?
Mention why do you need Session Handling while working with Selenium?
Mention why to choose Python over Java in Selenium?
Other than the default port 444 how you can run Selenium Server?
To enter values onto text boxes what is the command that can be used?
To generate pdf reports mention what Java API is required?
Using Selenium how can you handle network latency ?

Selenium Interview Questions for Experienced

Using Selenium IDE is it possible to get data from a particular html table cell ?
Using web driver how you can store a value which is text box?
What are core extension ?
What are the advantages of RC?
What are the advantages of Selenium?
What are the features of TestNG and list some of the functionality in TestNG which makes it more effective?
What are the four parameter you have to pass in Selenium? Four parameters that you have to pass in Selenium are
What are the technical limitations while using Selenium RC?
What are the two modes of views in Selenium IDE ?
What if you have written your own element locator and how would you test it?
What is heightened privileges browsers?
What is JUnit Annotations and what are different types of annotations which are useful ?
What is Object Repository ?
What is regular expressions? How you can use regular expressions in Selenium ?

What is same origin policy? How you can avoid same origin policy?
What is Selenese and what are the types of Selenese ?
What is selenium RC (Remote Control)?

Selenium Tricky Interview Questions

What is the command that is used in order to display the values of a variable into the output console or log?
What is the difference between getWindowhandles() and getwindowhandle() ?
What is the difference between setSpeed() and sleep() methods?
What is the difference between type keys and type commands ?
What is the difference between verify and assert commands?
Which attribute you should consider throughout the script in frame for "if no frame Id as well as no frame name"?
Which web driver implementation is fastest?
While injecting capabilities in webdriver to perform tests on a browser which is not supported by a webdriver what is the limitation that one can come across?
While using click command can you use screen coordinate?
Why Selenium RC is used?
Why testers should opt for Selenium and not QTP? Selenium is more popular than QTP as
Why to use TestNG with Selenium RC ?
Source: Guru99.com

Selenium Webdriver Interview Questions for 2 Years Experience

Question 1) What are the annotations used in TestNG ?
Question 2) How do you read data from excel ?
Question 3) What is the use of xpath ?
Question 4) What are different types of locators ?
Question 5) What is the difference between Assert and Verify?
Question 6) What is the alternate way to click on login button?
Question 7) How do you verify if the checkbox/radio is checked or not ?
Question 8) How do you handle alert pop-up ?
Question 9) How do you launch IE/chrome browser?
Question 10) How to perform right click using WebDriver?
Question 11) How do perform drag and drop using WebDriver?
Question 12) Give the example for method overload in WebDriver.
Question 13) How do you upload a file?
Question 14) How do you click on a menu item in a drop down menu?
Question 15) How do you simulate browser back and forward ?
Question 16) How do you get the current page URL ?
Question 17) What is the difference between '/' and '//' ?
Question 18) What is the difference between findElement and findElements?
Question 19) How do you achieve synchronization in WebDriver ?
Question 20) Write the code for Reading and Writing to Excel through Selenium ?

Selenium Interview Questions for 3 Years Experience

Question 21) How to get typed text from a textbox ?
Question 22) What are the different exceptions you got when working with WebDriver ?
Question 23) What are the languages supported by WebDriver ?
Question 24) How do you clear the contents of a textbox in selenium ?
Question 25) What is a Framework ?
Question 26) What are the prerequisites to run selenium webdriver?
Question 27) What are the advantages of selenium webdriver?
Question 28) What is WebDriverBackedSelenium ?
Question 29) How to invoke an application in webdriver ?
Question 30) What is Selenium Grid ?
Question 31) How to get the number of frames on a page ?
Question 32) How do you simulate scroll down action ?
Question 33) What is the command line we have to write inside a .bat file to execute a selenium project when we are using testng ?
Question 34) Which is the package which is to be imported while working with WebDriver ?
Question 35) How to check if an element is visible on the web page ?
Question 36) How to check if a button is enabled on the page ?
Question 37) How to check if a text is highlighted on the page ?

Question 38) How to check the checkbox or radio button is selected ?
Question 39) How to get the title of the page ?
Question 40) How do u get the width of the textbox ?
Selenium Interview Questions for 4 Years Experience
Question 41) How do u get the attribute of the web element ?
Question 42) How to check whether a text is underlined or not ?
Question 43) How to change the URL on a webpage using selenium web driver ?
Question 44) How to hover the mouse on an element ?
Question 45) What is the use of getOptions() method ?
Question 46) What is the use of deSelectAll() method ?
Question 47) Is WebElement an interface or a class ?
Question 48) FirefoxDriver is class or an interface and from where is it inherited ?
Question 49) Which is the super interface of webdriver ?
Question 50) What is the difference b/w close() and quit()?

Question: 1. What do you mean by automation testing or test automation?

Answer: Automation testing automates themanual testing process which facilitates thecreation of multiple testing scripts that can be executed repeatedly without any manual intervention.

Question: 2. Are there any benefits of automation testing over manual testing?

Answer: The advantages of automation testing are just the endless. Here we have discussed on some of the major benefits that are usually experienced by testing professionals during the entire process –

- Automation testing is more accurate and less prone to human errors.
- It is faster and budget solution tool
- It helps in testing large modules in thelimited time span.
- It allows facility of parallel execution whenever needed.

Question: 3. Selenium is a single testing tool, then why it is taken as a suite by professionals?

Answer: The tools available to work with selenium are specially designed to cater different testing requirements so it is taken as atesting package not only a single tool.

Read: Agile Scrum Master Interview Questions And Answers

Question: 4. What are selenium supporting testing types?

Answer: Selenium tool supports two types of testing – Functional testing and Regression testing.

Question: 5. Are there any drawbacks of selenium testing tool?

Answer: The functionality of selenium testing tool is limited to web applications only. It cannot be used mobile apps or API testing. Further, selenium tool does not provide a facility to test Captcha and barcodes. You must have knowledge of some programming language when working on selenium. For report generation, you have to use third-party tools.

Question: 6. What do you understand by the term "Selenese"?

Answer: Selenese is a popular programming language that is used to write test scripts in Selenium IDE.

Question: 7. How to identify web elements in Selenium?

Answer: To identify web elements in Selenium, you need to work with locators. They are the address identifiers to identify web elements accurately and more precisely. The most popular locators in Selenium include ID, TagName, CSS Selector, DOM, ClassName, LinkTest, Name, XPath and many others.

Question: 8. Explain the meaning of XPath in Selenium?

Answer: Based on XML path of web elements, XPath is used to locate elements on web page. In other words, it is popular to locate HTML elements on a webpage.

Question: 9. How to count a number of elements on a page?

Answer: To count a number of elements on a page, you first need to locate them and count the final size for the page.

Question: 10. Can you name the popular selenium tool used by large industries worldwide?

Answer: Selenium Web Driver is apopular testing tool used by big Companies today.

Selenium WebDriver Interview Questions and Answers

Question: 11. Explain about the waiting methods in selenium Web Driver?

Answer: The two popular waiting methods in selenium Web Driver are – implicit wait an explicit wait. Implicit waiting method is used when Web Driver is not able to find elements in the document, then it waits for thecertain time period for theelement to appear in the document.

At the same time, the explicit waiting method is an advanced technique that allows developers to write custom codes whenever required.

Question: 12. Where implicit waiting method and explicit it waiting method can be used in your script?

Answer: Implicit waiting method is time-consuming, but suitable for all types of elements, but theexplicit wait time is used for selected elements only.

Read: Top Business Analyst Interview Questions and Answers

Question: 13. Is there any technique to check either button is enabled on thepage or not?

Answer: For this purpose, you need to use theEnabled() method in your script. The return value of the method would be a Boolean. If thereturn value is true, then thebutton will be enabled otherwise it will not.

Question: 14. Is there any technique to check either particular element is visible on thepage or not?

Answer: For this purpose, you need to use the is Displayed() method in your script. The return value from the method would be Boolean. If thereturn value is true then theelement will be visualized otherwise it will not.

Question: 15. Name the different type of drivers available in WebDriver?

Answer: These are Chrome, Firefox, Safari, Internet Explorer, iPhone, Android, HTML Unit, and Opera Drivers etc.

Question: 16. Name the different type of mobile testing drivers supported in WebDriver?

Answer: These are iPhone, Android, and Opera Mobile Drivers etc.

Question: 17. Name the programming languages that are used by Web Driver to write the test cases?

Answer: These are PHP, Python, Ruby, C#, PERL, Java etc.

Question: 18. Do you know the difference between assert commands and verifying commands in Selenium WebDriver?

Answer: Assert command and verify commands both will check the conditions for Booleans values. In case thecondition is false then assert command will halt the program but verify command does not halt the program but it takes it to the next phase of execution.

Question: 19. Explain the difference between"/" and "//"in XPathSelenium Web Driver?

Answer: As we have discussed already based on XML path of web elements, XPath is used to locate elements on web page. Like UNIX, Linux, or any other programming language, single slash ("/") represents absolute path while double slash ("//") in XPath represents arelative path.

Question: 20. Name the different type of annotations used in Selenium WebDriver?

Answer: The popular annotations used in Selenium WebDriver include After, Before, Test, Ignore, AfterClass, BeforeClass, and RunWith etc.

Selenium Tricky Interview Questions and Answers

Question: 21. Which selenium technology is useful for distributed data processing?

Answer: Selenium Grid is a popular technology used for distributed data process that distributes tests on multiple machines in parallel. In other, tests can be executed in parallel on different operating systems, different web browsers at the same time by using single script only. Distributed data processing in Selenium not only reduces overall execution time and feedback is also quick.

Read: Automation Testing Interview Questions & Answers

Question: 22. How will you check result of text execution in Selenium IDE?

Answer: The result for text execution in selenium IDE is displayed in Log Window.

Question: 23. How can you modify test cases in Selenium IDE?

Answer: Obviously, test cases can be modified in Selenium IDE by table views or by looking up the source code.

Question: 24. Explain the significance of JUnit in Selenium testing tool?

Answer: JUnit is an open source framework introduced by Apache to test Java application.

Question: 25. How to add meta-data in JUnit selenium testing tool?

Answer: JUnit Annotations is a special process to add syntactic meta-data to Java code. Some of the popular examples of JUnit annotations include variables, parameters, classes, or methods etc.

Question: 26. How will you explain the difference between quit () and close () methods in Selenium?

Answer: Quit() method ends all the browsers running in Web Driver while close() method ends the current browser only.

Question: 27. Why was the concept of waiting method introduced in Selenium? Name the type of waits in Selenium?

Answer: The waiting methods werespecially introduced for AJAX based apps in Selenium WebDriver. The two popular types of waits in Selenium are names as Implicit waits and Explicit waits.

Question: 28. Is there any drawback of implementing implicit waits in Selenium?

Answer: The major drawback of implementing implicit waits in Selenium is exceeded time limits. It slows down the overall performance of test cases.

Question: 29. Why Selenium tools failsometimes?

Answer: Selenium testing is slow as compared to other automated testing tools. Another major problem complex ID generation where auto-generated ID can result into test case failure during playback.

Question: 30. Name the four parameters that have to be passed in Selenium?

Answer: These parameters are aPort number, Host, URL, and Browser etc.

1. What is Selenium? What are the different Selenium components?

According to the website SoftwareTestingHelp.com, the answer is that Selenium is one of the most popular automated testing suites. The website shows how Selenium is designed to support and encourage automation testing of functional aspects of web-based applications, as well as a wide range of browsers and platforms. Due to its open source nature, Selenium has become one of the most accepted tools amongst the testing professionals.

Your answer should also point out that rather than being a single entity, Selenium is more of an overall package of several testing tools, and so is referred to as a suite. SoftwareTestingHelp.com identifies the following sets of tools:

Selenium Integrated Development Environment (IDE): a record and playback tool, distributed as a Firefox Plugin.

Selenium Remote Control (RC): a server that allows creation of test scripts in a programming language, and execution of test scripts across a broad spectrum of browsers.

Selenium WebDriver: a tool with a number of advantages over Selenium RC, including direct communication with the web browser, and the ability to use native compatibility to automate.

Selenium Grid: used to distribute your test execution on multiple platforms and environments concurrently.

2. What functions/methods can you use to wait for a page to load, or some element on the page to show up?
Toptal.com advises that explicit wait can be used to stall, until a specific condition is met. *Thread.Sleep()* is one form of explicit wait that stalls the thread for a certain duration of time, but may not be the most reliable way to wait for a page to load. If you wish to have the thread 'sleep' for 30 seconds (30,000 milliseconds), Toptal advises *not* using something like:

```
Thread.Sleep(30000);
```

but instead, using the following … where the thread sleeps for 30 seconds (30,000 milliseconds), the following can be used:

```
WebDriverWait wait= new WebDriverWait(webDriver,30);
wait.until(ExpectedConditions.visibilityOf(h1));
```

… where Selenium waits until the given element is visible on the page, or throws TimeoutException after waiting for 30 seconds.

3. Explain the difference between single and double slash in X-path?
According to contributors to the web community on Guru99.com, the Single slash '/ ' starts selection from the document node, and allows the creation of 'absolute' path expressions.
On the other hand, the double slash ' // ' can start a selection matching anywhere in the document, and enables the user to 'relative' path expressions

4. How do you deal with frame elements in Selenium on a page?
The Toptal website advices that you must first switch to the frame in order to manipulate either the frame or its content. It's the same principle as switching to a different page before you can interact with it (see below)

```
driver.switchTo().frame(index);
```

Here, *index* is the zero-based index of the frame. Switching the frame directs all further interactions through the driver towards the selected frame. The *frame* method also works with name, element ID and reference to already located elements.
To switch back to the default frame, the *defaultContent* method can be used:

```
driver.switchTo().defaultContent();
```

5. How would you make sure that a page is loaded using Selenium and Webdriver?
The answer provided by TechBeamers.com to this question is that the following lines of code will check for successful loading. They advise that the best approach is to select an element from the page and standby until it becomes clickable.

```
selenium.waitForPageToLoad("5000");
// Or
while (!(selenium.isElementPresent("any page element ")==true)) {
  selenium.setSpeed("5");
  Thread.sleep(5);
}
```

And here is the Webdriver specific code that will achieve the same objective.

```
WebDriverWait check = new WebDriverWait(driver, 100);
check.until(ExpectedConditions.anyElement(By.id(id)));
```

QUESTION & ANSWER SOURCE: TECHBEAMERS.COM

6. What is difference between assert and verify commands?
Here is a good, comprehensive answer from SoftwareTestingHelp.com. Proof that 'difficult' doesn't necessarily mean 'tricky'… just being sure that you know your subject inside and out.
Assert: this command checks whether the given condition is true or false. Let's say we assert whether the given element is present on the web page or not. If the condition is true then the program control will execute the next test step but if the condition is false, the execution would stop and no further test would be executed.
Verify: this command also checks whether the given condition is true or false. Irrespective of the condition being true or false, the program execution doesn't halts i.e. any failure during verification would not stop the execution and all the test steps would be executed.
QUESTION & ANSWER SOURCE: SOFTWARETESTINGHELP.COM

7. What are the different exceptions you face in Selenium Webdriver?
Here is a list of exceptions provided by the Techbeamers.com website

- WebDriverException
- NoAlertPresentException
- NoSuchWindowException
- NoSuchElementException
- TimeoutException

QUESTION & ANSWER SOURCE: TECHBEAMERS.COM

8. What are the technical challenges with Selenium?
According to contributors on Guru99.com, the technical challenges with Selenium include.

- Selenium supports only web based applications
- It does not support the Bitmap comparison
- For any reporting-related capabilities, the user must depend on third-party tools
- There is no vendor support, unlike the backup available for commercial tools such as HP UFT
- Maintainability of objects can be difficult, as Selenium has no object repository concept.

QUESTION & ANSWER SOURCE: GURU99.COM

9. What is Page Object Model and Page Factory in Selenium?
Here is a comprehensive response to this question from Toptal.com.
Page Object Model is a design pattern where web pages are represented using classes. Variables in the class can then be used to hold references to elements on the web page, and methods can be used to implement behaviours. This allows an elegant way of implementing test routines that are both readable and easier to maintain and extend in the future.

```
clickLoginButton();
setCredentials(user_name, user_password);
submitLoginForm();
```

Page Factory is used to initialize every WebElement variable with a reference to a corresponding element on the actual web page using configured “locators”. Annotations, such as @FindBy, can be used to define strategies for looking up elements, along with the necessary information for identifying them:

```
@FindBy(how=How.NAME, using="username")
private WebElement user_name;
```

QUESTION & ANSWER SOURCE: TOPTAL.COM

10. How do you read a JavaScript variable in Selenium WebDriver?
Ans. It’s easy to access any JavaScript variable from the Selenium Webdriver test scripts. Just you need to use the below Java code snippet.

```
// Initialize the JS object.
JavascriptExecutor JS = (JavascriptExecutor) webdriver;
// Get the current site title.
String sitetitle = (String)JS.executeScript("return document.title");
System.out.println("My Site Title: " + sitetitle);
```

Q: How many types of Webdriver APIs are available in Selenium?
Ans. Below is the list of driver classes that you can use for the browser automation.

- AndroidDriver,
- ChromeDriver,
- EventFiringWebDriver,
- FirefoxDriver,
- HtmlUnitDriver,
- InternetExplorerDriver,
- IPhoneDriver,

- IPhoneSimulatorDriver,
- RemoteWebDriver.

Q: How would you make sure that a page is loaded using Selenium and Webdriver?
Ans. In Selenium, you can use the below lines of code to check for the successful loading of a web page. The best approach is by selecting an element from the page & stand by till it becomes clickable.

```
selenium.waitForPageToLoad("5000");
// Or
while (!(selenium.isElementPresent("any page element ")==true)) {
 selenium.setSpeed("5");
 Thread.sleep(5);
}
```

Below is the Webdriver specific code to achieve the same objective.

```
WebDriverWait check = new WebDriverWait(driver, 100);
check.until(ExpectedConditions.anyElement(By.id(id)));
```

Q: How to launch a batch file in a Selenium Webdriver project?
Ans. It's usual in a test suite to run a batch file or an executable file for setting up the pre-requisites before starting the automation. You can use the below Java code for this purpose.

```
Process batch = Runtime.getRuntime.exec("path of the batch file");
batch.waitFor();
```

Q: How do you read a JavaScript variable in Selenium WebDriver?
Ans. It's easy to access any JavaScript variable from the Selenium Webdriver test scripts. Just you need to use the below Java code snippet.

```
// Initialize the JS object.
JavascriptExecutor JS = (JavascriptExecutor) webdriver;
// Get the current site title.
String sitetitle = (String)JS.executeScript("return document.title");
System.out.println("My Site Title: " + sitetitle);
```

Q: How to run the selenium IDE test suite from the command line?
Ans. We've given the command to run the SIDE test suite in the next line.

```
Java -jar "C:\Selenium Webdriver Questions\selenium-server-standalone-2.33.0.jar" -htmlSuite "*firefox"
"http://www.google.com" " “C:\Selenium Webdriver Questions\SeleniumSuite.HTML"
```

Q: How to run the Selenium Webdriver test from the command line?
Ans. We can run any Selenium Webdriver test written in Java using the following command.

```
java -classpath ".;selenium-server-standalone-2.33.0.jar" SampleClass
```

Q: What are the different exceptions you face in Selenium Webdriver?
Ans.

- WebDriverException,
- NoAlertPresentException,
- NoSuchWindowException,
- NoSuchElementException,
- TimeoutException.

Q: How would you automatically click a screenshot whenever any exception occurs?
Ans. For this you will have to use <*EventFiringWebDriver*> class and needs to implement the <*onException*> method of the <*WebDriverEventListener*> interface. See the code example given below.

```
WebDriver browser = new FirefoxDriver();

EventFiringWebDriver eventDriver = new EventFiringWebDriver(browser).register(new AbstractWebDriverEventListener() {

 @Override
 public void onException(Throwable throwable, WebDriver browser) {

  // Take the screenshot using the Webdriver.
  File screen = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

  // Now you can copy the screenshot somewhere on your system.
  FileUtils.copyFile(screen, new File("c:\\Selenium Testing Questions\\screen.png"));
 }
});

try {

 eventDriver.findElement(By.id("test"));
 fail("Caught the Expected exception."); // Intentionally causing the exception for demo.

} catch (NoSuchElementException e) {

 // Triggering point for the <onException> event.
}
```

Q: How would you select any particular text using the Selenium Webdriver?
Ans. It seems an easy one at first but you need to do a little more to achieve this.

```
driver.get("/");
WebElement item = driver.findElement(By.xpath("//p[contains(text(),'Selenium webdriver quesions')]"));
Actions dummy = new Actions(driver);
dummy.doubleClick(item).build().perform();
```

Q: Give an example to perform drag and drop action In Selenium WebDriver?
Ans. Yes, we can use the Advanced User Interactions API to perform drag and drop operations in a Selenium Webdriver project.
Code example:

```
Actions act = new Actions(driver);
act.dragAndDrop(source_locator, target_locator).build().perform();
//Or you can use the below code style.
(new Actions(driver)).dragAndDrop(source_locator, target_locator).perform();
```

Q: How would you fill a text field without calling the sendKeys()?
Ans. It's a bit slower than the sendKeys() method but we do have means to type in a text field. See the Java code given below.

```
JavascriptExecutor JS = (JavascriptExecutor)webdriver;
```

```
JS.executeScript("document.getElementById(User').value='admin@testmail.com'");

JS.executeScript("document.getElementById('Pass').value='######'");
```

Q: How can you check the state of a checkbox/radio button?
Ans. We can call the isSelected() method to test the status of these elements.
Example Code:

```
boolean test = driver.findElement(By.xpath("checkbox/radio button XPath")).isSelected();
```

Q: How would you handle the alert popups in Selenium Webdriver?
Ans. First, you've to switch the control to the pop up then press the ok or cancel button. After that, turn back to the source page screen.
Code Example:

```
String srcPage = driver.getWindowHandle();

Alert pop = driver.switchTo().alert(); // shift control to the alert popup.

pop.accept(); // click on ok button.

pop.dismiss(); // click on cancel button.

// Move the control back to source page.

driver.switchTo().window(srcPage);  // move back to the source page.
```

Q: What is the process to start the IE/Chrome browser?
Ans. If you want to start a browser then, just set the system properties as mentioned below.

```
// For the IE web browser.

System.setProperty("webdriver.ie.driver"," iedriver.exe file path");

WebDriver driver = new InternetExplorerDriver();

// For the Chrome web browser.

System.setProperty("webdriver.chrome.driver","chrome.exe file path");

WebDriver driver = new ChromeDriver();
```

Q: How would you simulate the right click operation in WebDriver?
Ans. You can make use of the Actions class features.

```
Actions test = new Actions(driver); // Here, driver is the object of WebDriver class.

test.moveToElement(element).perform();
```

```
test.contextClick().perform();
```

Q: How would you select a menu item from a drop down menu?
Ans. There can be following two situations.

- If the menu has is using the <***select***> tag then you can call the <***selectByValue()***> or <**selectByIndex()**> or <**selectByVisibleText()**> methods of the Select class.
- If the menu doesn't use the <***select***> tag then simply find the XPath of that element and perform the click action for its selection.

Q: What is the FirefoxDriver, class or an interface? And which interface does it implement?
Ans. FirefoxDriver is a Java class, and it implements the <***WebDriver***> interface. It contains the implementations of all the methods available in the <***WebDriver***> interface.

Q: What is the name of the super interface of the Webdriver?
Ans. SearchContext.

Q: What is the main difference between the close() and quit() methods?
Ans.
close() - it closes the currently active browser window.
quit()- it will close all of the opened browser windows and the browser itself.

Q: What is the best way to check for the highlighted text on a web page?
Ans. Use the below code to verify the highlighted text for an element on the web page.

```
String clr = driver.findElement(By.xpath("//a[text()='TechBeamers']")).getCssValue("color");


String bkclr = driver.findElement(By.xpath("//a[text()='TechBeamers']")).getCssValue("background-color");


System.out.println(clr);


System.out.println(bkclr);
```

Q: How would you use a Selenium variable say "size" from the JavaScript?
Ans. ${size}

Q: What is the Selenese command to show the value of a variable in the log file?
Ans. echo()

Q #1) What is JAVA?
Ans: Java is a high-level programming language and is platform independent.
Java is a collection of objects. It was developed by Sun Microsystems. There are a lot of applications, websites and Games that are developed using Java.

Q #2) What are the features in JAVA?
Ans: Features of Java:

- **Oops concepts**
  - Object-oriented
  - Inheritance
  - Encapsulation
  - Polymorphism
  - Abstraction
- **Platform independent:** A single program works on different platforms without any modification.
- **High Performance:** JIT (Just In Time compiler) enables high performance in Java. JIT converts the bytecode into machine language and then JVM starts the execution.
- **Multi-threaded:** A flow of execution is known as a Thread. JVM creates a thread which is called main thread. The user can create multiple threads by extending the thread class or by implementing Runnable interface.

Q #3) How does Java enable high performance?

Ans: Java uses Just In Time compiler to enable high performance. JIT is used to convert the instructions into bytecodes.

Q #4) What are the Java IDE's?

Ans: Eclipse and NetBeans are the IDE's of JAVA.

Q #5) What do you mean by Constructor?

Ans: The points given below explain what a Constructor is in detail:

- When a new object is created in a program a constructor gets invoked corresponding to the class.
- The constructor is a method which has the same name as class name.
- If a user doesn't create a constructor implicitly a default constructor will be created.
- The constructor can be overloaded.
- If the user created a constructor with a parameter then he should create another constructor explicitly without a parameter.

Q #6) What is meant by Local variable and Instance variable?

Ans: Local variables are defined in the method and scope of the variables that have existed inside the method itself.
An instance variable is defined inside the class and outside the method and scope of the variables exist throughout the class.

Q #7) What is a Class?

Ans: All Java codes are defined in a class. A Class has variables and methods.
Variables are attributes which define the state of a class.
Methods are the place where the exact business logic has to be done. It contains a set of statements (or) instructions to satisfy the particular requirement.

Example:

```
public class Addition{ //Class name declaration
int a = 5; //Variable declaration
int b= 5;
public void add(){ //Method declaration
int c = a+b;
}
}
```

Q #8) What is an Object?

Ans: An instance of a class is called object. The object has state and behavior.
Whenever the JVM reads the "new()" keyword then it will create an instance of that class.

Example:

```
public class Addition{
public static void main(String[] args){
Addion add = new Addition();//Object creation
}
}
```

The above code creates the object for the Addition class.

Q #9)What are the Oops concepts?

Ans: Oops concepts include:

- Inheritance
- Encapsulation
- Polymorphism
- Abstraction
- Interface

Q #10) What is Inheritance?

Ans: Inheritance means one class can **extend** to another class. So that the codes can be reused from one class to another class. Existing class is known as Super class whereas the derived class is known as a sub class.

Example:

```
Super class:
public class Manupulation(){
}
Sub class:
public class Addition extends Manipulation(){
}
```

Inheritance is applicable for public and protected members only. Private members can't be inherited.

Q #11) What is Encapsulation?

Ans: Purpose of Encapsulation:

- Protects the code from others.
- Code maintainability.

Example:

We are declaring 'a' as an integer variable and it should not be negative.

```
public class Addition(){
int a=5;
}
```

If someone changes the exact variable as "***a = -5"*** then it is bad.

In order to overcome the problem we need to follow the below steps:

- We can make the variable as private or protected one.
- Use public accessor methods such as set<property> and get<property>.

So that the above code can be modified as:

```
public class Addition(){
private int a = 5; //Here the variable is marked as private
}
```

Below code shows the getter and setter.

Conditions can be provided while setting the variable.

```
get A(){
}
set A(int a){
if(a>0){// Here condition is applied
.....
}
}
```

For encapsulation, we need to make all the instance variables as private and create setter and getter for those variables. Which in turn will force others to call the setters rather than access the data directly.

Q #12) What is Polymorphism?

Ans: Polymorphism means many forms.

A single object can refer the super class or sub-class depending on the reference type which is called polymorphism.

Example:

```
Public class Manipulation(){ //Super class
public void add(){
}
}
public class Addition extends Manipulation(){ // Sub class
public void add(){
}
public static void main(String args[]){
Manipulation addition = new Addition();//Manipulation is reference type and Addition is reference type
addition.add();
}
}
```

Using Manipulation reference type we can call the Addition class "add()" method. This ability is known as Polymorphism.

Polymorphism is applicable for **overriding** and not for **overloading**.

Q #13) What is meant by Method Overriding?

Ans: Method overriding happens if the sub class method satisfies the below conditions with the Super class method:

- Method name should be same
- Argument should be same
- Return type also should be same

The key benefit of overriding is that the Sub class can provide some specific information about that sub class type than the super class.

Example:

```
public class Manipulation{ //Super class
public void add(){
.....
}
}
```

```
Public class Addition extends Manipulation(){
Public void add(){
……..
}
Public static void main(String args[]){
Manipulation addition = new Addition(); //Polimorphism is applied
addition.add(); // It calls the Sub class add() method
}
}
```

addition.add() method calls the add() method in the Sub class and not the parent class. So it overrides the Super class method and is known as Method Overriding.

Q #14) What is meant by Overloading?

Ans: Method overloading happens for different classes or within the same class.

For method overloading, subclass method should satisfy the below conditions with the Super class method (or) methods in the same class itself:

- Same method name
- Different argument type
- May have different return types

Example:

```
public class Manipulation{ //Super class
public void add(String name){ //String parameter
……..
}
}

Public class Addition extends Manipulation(){
Public void add(){//No Parameter
……..
}
Public void add(int a){ //integer parameter

}
Public static void main(String args[]){
Addition addition = new Addition();
addition.add();
}
}
```

Here the add() method having different parameters in the Addition class is overloaded in the same class as well as with the super class.

Note: Polymorphism is not applicable for method overloading.

Q #15) What is meant by Interface?

Ans: Multiple inheritance cannot be achieved in java. To overcome this problem Interface concept is introduced.
An interface is a template which has only method declarations and not the method implementation.

Example:

```
Public abstract interface IManupulation{ //Interface declaration
Public abstract void add();//method declaration
public abstract void subtract();
}
```

- All the methods in the interface are internally **public abstract void**.
- All the variables in the interface are internally **public static final** that is constants.
- Classes can implement the interface and not extends.
- The class which implements the interface should provide an implementation for all the methods declared in the interface.

```
public class Manupulation implements IManupulation{ //Manupulation class uses the interface
Public void add(){
……..
}
Public void subtract(){
```

```
……..
}
}
```

Q #16) What is meant by Abstract class?

Ans: We can create the Abstract class by using "Abstract" keyword before the class name. An abstract class can have both "Abstract" methods and "Non-abstract" methods that are a concrete class.

Abstract method:

The method which has only the declaration and not the implementation is called the abstract method and it has the keyword called "abstract". Declarations are the ends with a semicolon.

Example:

```
public abstract class Manupulation{
public abstract void add();//Abstract method declaration
Public void subtract(){
}
}
```

- An abstract class may have a Non- abstract method also.
- The concrete Subclass which extends the Abstract class should provide the implementation for abstract methods.

Q #17) Difference between Array and Array List.

Ans: The Difference between Array and Array List can be understood from the below table:

Array	Array List
Size should be given at the time of array declaration. String[] name = new String[2]	Size may not be required. It changes the size dynamically. ArrayList name = new ArrayList
To put an object into array we need to specify the index. name[1] = "book"	No index required. name.add("book")
Array is not type parameterized	ArrayList in java 5.0 are parameterized. Eg: This angle bracket is a type parameter which means a list of String.

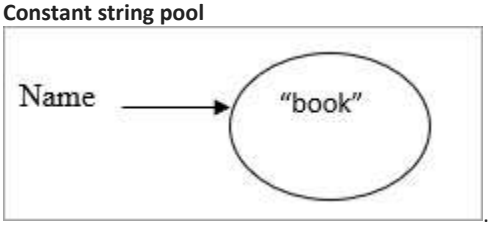
Q #18) Difference between String, String Builder, and String Buffer.

Ans: String: String variables are stored in "constant string pool". Once the string reference changes the old value that exists in the "constant string pool", it cannot be erased.

Example:

String name = "book";

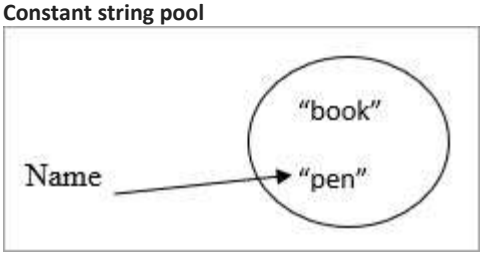
Constant string pool



.

If the name value has changed from "book" to "pen".

Constant string pool



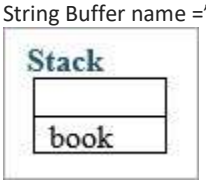
Then the older value retains in the constant string pool.

String Buffer:

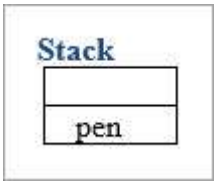
- Here string values are stored in a stack. If the values are changed then the new value replaces the older value.
- The string buffer is synchronized which is thread-safe.
- Performance is slower than the String Builder.

Example:

String Buffer name =”book”;



Once the name value has been changed to “pen” then the “book” is erased in the stack.



String Builder:

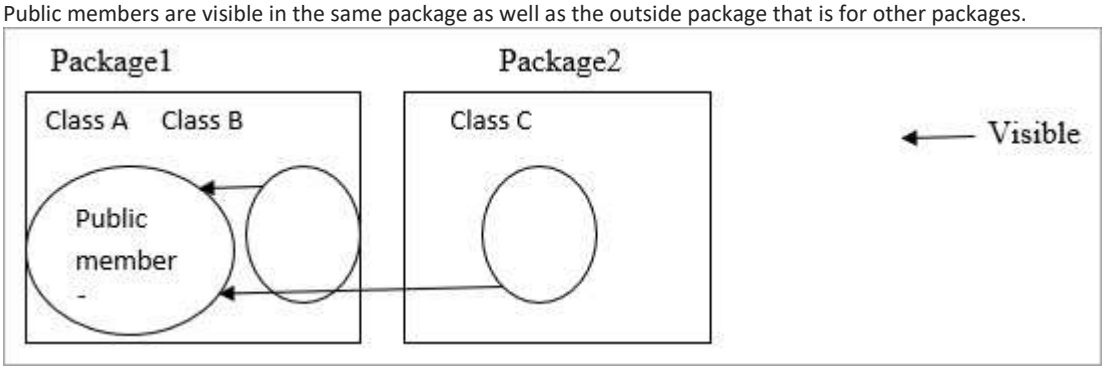
This is same as String Buffer except for the String Builder which is not threaded safety that is not synchronized. So obviously performance is fast.

Q #19) Explain about Public and Private access specifiers.

Ans: Methods and instance variables are known as members.

Public:

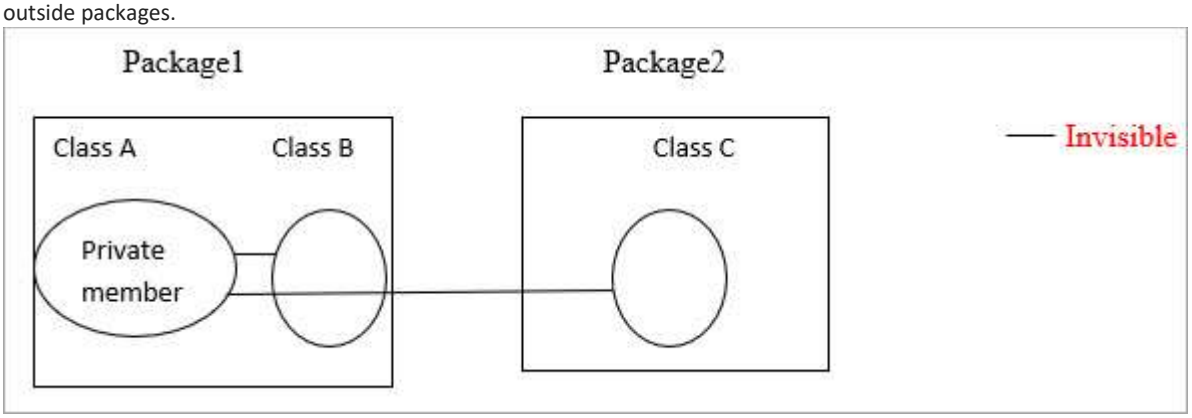
Public members are visible in the same package as well as the outside package that is for other packages.



Public members in Class A are visible to Class B (Same package) as well as Class C (Different package).

Private:

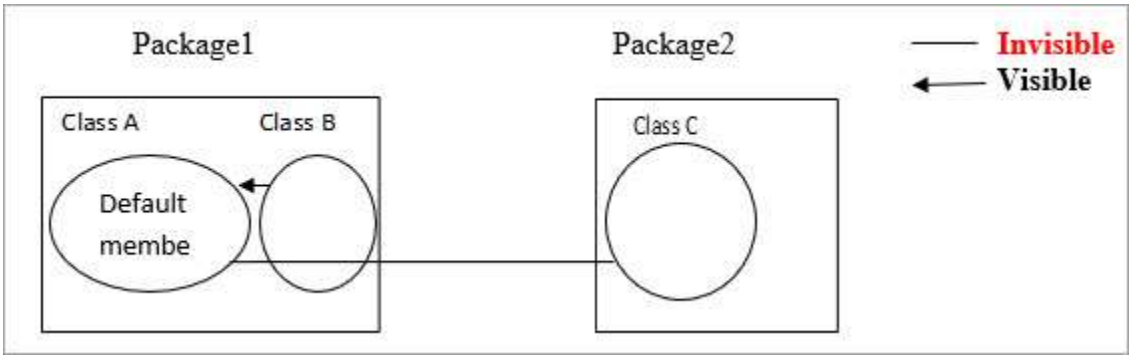
Private members are visible in the same class only and not for the other classes in the same package as well as classes in the outside packages.



Private members in class A is visible only in that class. It is invisible for class B as well as class C.

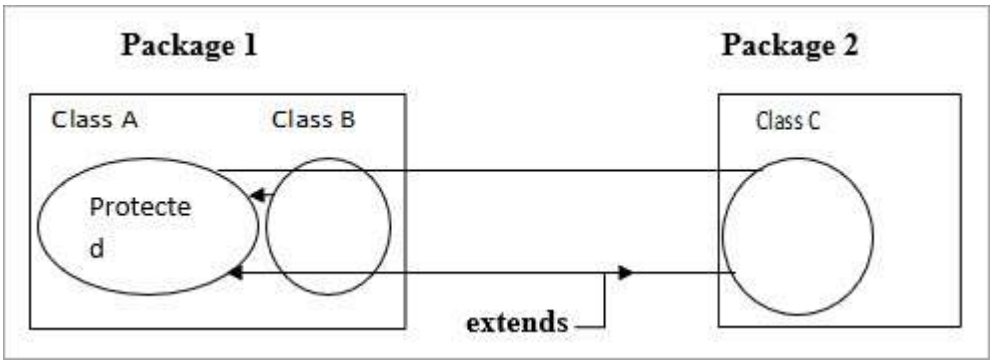
Q #20) Difference between Default and Protected access specifiers.

Ans: Default: Methods and variables declared in a class without any access specifiers are called default.



Default members in Class A are visible to the other classes which are inside the package and invisible to the classes which are outside the package.
So Class A members are visible to the Class B and invisible to the Class C.

Protected:



Protected is same as Default but if a class extends then it is visible even if it is outside the package.
Class A members are visible to Class B because it is inside the package. For Class C it is invisible but if Class C extends Class A then the members are visible to the Class C even if it is outside the package.

Q #21) Difference between HashMap and HashTable.

Ans: Difference between HashMap and HashTable can be seen below:

HashMap	HashTable
Methods are not synchronized	Key methods are synchronized
Not thread safety	Thread safety
Iterator is used to iterate the values	Enumerator is used to iterate the values
Allows one null key and multiple null values	Doesn't allow anything that is null
Performance is high than HashTable	Performance is slow

Q #22) Difference between HashSet and TreeSet.

Ans: Difference between HashSet and TreeSet can be seen below:

HashSet	TreeSet
Inserted elements are in random order	Maintains the elements in the sorted order
Can able to store null objects	Couldn't store null objects
Performance is fast	Performance is slow

Q #23) Difference between Abstract class and Interface.

Ans: Difference between Abstract Class and Interface are as follows:

Abstract Class:

- Abstract classes have a default constructor and it is called whenever the concrete subclass is instantiated.
- Contains Abstract methods as well as Non-Abstract methods.
- The class which extends the Abstract class shouldn't require implementing all the methods, only Abstract methods need to be implemented in the concrete sub-class.
- Abstract Class contains instance variables.

Interface:

- Doesn't have any constructor and couldn't be instantiated.
- Abstract method alone should be declared.
- Classes which implement the interface should provide the implementation for all the methods.
- The interface contains only constants.

Q #24) What is mean by Collections in Java?

Ans: Collection is a framework that is designed to store the objects and manipulate the design to store the objects.

Collections are used to perform the following operations:

- Searching
- Sorting
- Manipulation
- Insertion
- Deletion

A group of objects is known as collections. All the classes and interfaces for collecting are available in Java utile package.

Q #25) What are all the Classes and Interfaces that are available in the collections?

Ans: Given below are the Classes and Interfaces that are available in Collections:

Interfaces:

- Collection
- List
- Set
- Map
- Sorted Set
- Sorted Map
- Queue

Classes:

- Lists:
- Array List
- Vector
- Linked List

Sets:

- Hash set
- Linked Hash Set
- Tree Set

Maps:

- Hash Map
- Hash Table
- Tree Map
- Linked Hashed Map

Queue:

- Priority Queue

Q #26) What is meant by Ordered and Sorted in collections?

Ans:

Ordered:

It means the values that are stored in a collection is based on the values that are added to the collection. So we can iterate the values from the collection in a specific order.

Sorted:

Sorting mechanism can be applied internally or externally so that the group of objects sorted in a particular collection is based on properties of the objects.

Q #27) Explain about the different lists available in the collection.

Ans: Values added to the list is based on the index position and it is ordered by index position. Duplicates are allowed.

Types of Lists are:

Array List:

- Fast iteration and fast Random Access.
- It is an ordered collection (by index) and not sorted.
- It implements Random Access Interface.

Example:

```
public class Fruits{
public static void main (String [ ] args){
ArrayList <String>names=new ArrayList <String>();
names.add (“apple”);
names.add (“cherry”);
names.add (“kiwi”);
names.add (“banana”);
names.add (“cherry”);
System.out.println (names);
}
}
```

Output:

[Apple, cherry, kiwi, banana, cherry]

From the output, Array List maintains the insertion order and it accepts the duplicates. But not sorted.

Vector:

It is same as Array List.

- Vector methods are synchronized.
- Thread safety.
- It also implements the Random Access.
- Thread safety usually causes a performance hit.

Example:

```
public class Fruit {
public static void main (String [ ] args){
Vector <String> names = new Vector <String> ( );
 names.add (“cherry”);
names.add (“apple”);
names.add (“banana”);
names.add (“kiwi”);
names.add (“apple”);
System.out.println (“names”);
}
}
```

Output:

[cherry,apple,banana,kiwi,apple]

Vector also maintains the insertion order and accepts the duplicates.

Linked List:

- Elements are doubly linked to one another.
- Performance is slow than Array list.
- Good choice for insertion and deletion.
- In Java 5.0 it supports common queue methods peek( ), Pool ( ), Offer ( ) etc.

Example:

```
public class Fruit {
public static void main (String [ ] args){
Linkedlist <String> names = new linkedlist <String> ( ) ;
names.add(“banana”);
names.add(“cherry”);
names.add(“apple”);
names.add(“kiwi”);
names.add(“banana”);
```

```
System.out.println (names);
}
}
```

Output

[banana,cherry,apple,kiwi,banana]

Maintains the insertion order and accepts the duplicates.

Q #28) Explain about Set and their types in a collection?

Ans: Set cares about uniqueness. It doesn't allow duplications. Here "equals ( )" method is used to determine whether two objects are identical or not.

Hash Set:

- Unordered and unsorted.
- Uses the hash code of the object to insert the values.
- Use this when the requirement is "no duplicates and don't care about the order".

Example:

```
public class Fruit {
public static void main (String[ ] args){
HashSet<String> names = new HashSet <=String>( ) ;
names.add("banana");
names.add("cherry");
names.add("apple");
names.add("kiwi");
names.add("banana");
System.out.println (names);
}
}
```

Output:

[banana, cherry, kiwi, apple]

Doesn't follow any insertion order. Duplicates are not allowed.

Linked Hash set:

- An ordered version of the hash set is known as Linked Hash Set.
- Maintains a doubly-Linked list of all the elements.
- Use this when the iteration order is required.

Example:

```
public class Fruit {
public static void main (String[ ] args){
LinkedHashSet<String> names = new LinkedHashSet <String>( ) ;
 names.add("banana");
 names.add("cherry");
 names.add("apple");
 names.add("kiwi");
 names.add("banana");
 System.out.println (names);
 }
}
```

Output:

[banana, cherry, apple, kiwi]

Maintains the insertion order in which they have been added to the Set. Duplicates are not allowed.

Tree Set:

- It is one of the two sorted collections.
- Uses "Read-Black" tree structure and guarantees that the elements will be in an ascending order.
- We can construct a tree set with the constructor by using comparable (or) comparator.

Example:

```
public class Fruits{
public static void main (String[ ]args) {
Treeset<String> names= new TreeSet<String>( ) ;
```

```
names.add(“cherry”);
names.add(“banana”);
names.add(“apple”);
names.add(“kiwi”);
names.add(“cherry”);
System.out.println(names);
}
}
```

Output:

[apple, banana, cherry, kiwi]

TreeSet sorts the elements in an ascending order. And duplicates are not allowed.

Q #29). Explain about Map and their types.

Ans: Map cares about unique identifier. We can map a unique key to a specific value. It is a key/value pair. We can search a value, based on the key. Like set, Map also uses “equals ( )” method to determine whether two keys are same or different.

Hash Map:

- Unordered and unsorted map.
- Hashmap is a good choice when we don’t care about the order.
- It allows one null key and multiple null values.

Example:

```
Public class Fruit{
Public static void main(String[ ] args){
HashMap<Sting,String> names =new HashMap<String,String>( );
names.put(“key1”,“cherry”);
names.put (“key2”,“banana”);
names.put (“key3”,“apple”);
names.put (“key4”,“kiwi”);
names.put (“key1”,“cherry”);
System.out.println(names);
}
 }
```

Output:

{key2 =banana, key1=cherry, key4 =kiwi, key3= apple}

Duplicate keys are not allowed in Map.

Doesn’t maintain any insertion order and is unsorted.

Hash Table:

- Like vector key, methods of the class are synchronized.
- Thread safety and therefore slows the performance.
- Doesn’t allow anything that is null.

Example:

```
public class Fruit{
public static void main(String[ ]args){
Hashtable<Sting,String> names =new Hashtable<String,String>( );
names.put(“key1”,“cherry”);
names.put(“key2”,“apple”);
names.put(“key3”,“banana”);
names.put(“key4”,“kiwi”);
names.put(“key2”,“orange”);
System.out.println(names);
}
 }
```

Output:

{key2=apple, key1=cherry,key4=kiwi, key3=banana}

Duplicate keys are not allowed.

Linked Hash Map:

- Maintains insertion order.
- Slower than Hash map.
- Can expect a faster iteration.

Example:

```
public class Fruit{
public static void main(String[ ] args){
LinkedHashMap<Sting,String> names =new LinkedHashMap<String,String>( );
 names.put("key1","cherry");
 names.put("key2","apple");
 names.put("key3","banana");
 names.put("key4","kiwi");
 names.put("key2","orange");
 System.out.println(names);
 }
 }
```

Output:

{key2=apple, key1=cherry,key4=kiwi, key3=banana}

Duplicate keys are not allowed.

TreeMap:

- Sorted Map.
- Like Tree set, we can construct a sort order with the constructor.

Example:

```
public class Fruit{
public static void main(String[ ]args){
TreeMap<Sting,String> names =new TreeMap<String,String>( );
names.put("key1","cherry");
names.put("key2","banana");
names.put("key3","apple");
names.put("key4","kiwi");
names.put("key2","orange");
System.out.println(names);
}
}
```

Output:

{key1=cherry, key2=banana, key3 =apple, key4=kiwi}

It is sorted in ascending order based on the key. Duplicate keys are not allowed.

Q #30) Explain the Priority Queue.

Ans: Queue Interface

Priority Queue: Linked list class has been enhanced to implement the queue interface. Queues can be handled with a linked list. Purpose of a queue is "Priority-in, Priority-out".

Hence elements are ordered either naturally or according to the comparator. The elements ordering represents their relative priority.

Q #31) What is mean by Exception?

Ans: An Exception is a problem that can occur during the normal flow of an execution. A method can throw an exception when something wails at runtime. If that exception couldn't be handled, then the execution gets terminated before it completes the task.

If we handled the exception, then the normal flow gets continued. Exceptions are a subclass of java.lang.Exception.

Example for handling Exception:

```
try{
//Risky codes are surrounded by this block
}catch(Exception e){
//Exceptions are caught in catch block
}
```

Q #32) What are the types of Exceptions?

Ans: Two types of Exceptions are explained below in detail.

Checked Exception:

These exceptions are checked by the compiler at the time of compilation. Classes that extend Throwable class except Runtime exception and Error are called checked Exception.

Checked Exceptions must either declare the exception using throes keyword (or) surrounded by appropriate try/catch.

E.g. ClassNotFound Exception

Unchecked Exception:

These exceptions are not checked during the compile time by the compiler. The compiler doesn't force to handle these exceptions.

It includes:

- Arithmetic Exception
- ArrayIndexOutOfBounds Exception

Q #33) What are the different ways to handle exceptions?

Ans: Two different ways to handle exception are explained below:

#1) Using try/catch:

A risky code is surrounded by try block. If an exception occurs, then it is caught by the catch block which is followed by the try block.

Example:

```
class Manipulation{
public static void main(String[] args){
add();
}
Public void add(){
try{
addition();
}catch(Exception e){
e.printStacktrace();
}
}
}
```

#2) By declaring throws keyword:

At the end of the method, we can declare the exception using throws keyword.

Example:

```
class Manipulation{
public static void main(String[] args){
add();
}
public void add() throws Exception{
addition();
}
}
```

Q #34) What are the Advantages of Exception handling?

Ans: Given below are the advantages:

- The normal flow of the execution won't be terminated if exception got handled
- We can identify the problem by using catch declaration

Q #35) What are Exception handling keywords in Java?

Ans: Given below are the two Exception Handling Keywords:

try:

When a risky code is surrounded by a try block. An exception occurring in the try block is caught by a catch block. Try can be followed either by catch (or) finally (or) both. But any one of the blocks is mandatory.

catch:

This is followed by try block. Exceptions are caught here.

finally:

This is followed either by try block (or) catch block. This block gets executed regardless of an exception. So generally clean up codes are provided here.

Q #36) Explain about Exception Propagation.

Ans: Exception is first thrown from the method which is at the top of the stack. If it doesn't catch, then it pops up the method and moves to the previous method and so on until they are got.

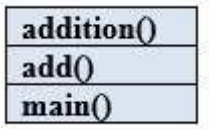
This is called Exception propagation.

Example:

```
public class Manipulation{
public static void main(String[] args){
add();
}
public void add(){
```

```
addition();
}
```

From the above example, the stack looks like as shown below:



If an exception occurred in the **addition()** method is not caught, then it moves to the method **add()**. Then it is moved to the **main()** method and then it will stop the flow of execution. It is called Exception Propagation.

Q #37) What is the final keyword in Java?

Ans:

Final variable:

Once a variable is declared as final, then the value of the variable could not be changed. It is like a constant.

Example:

```
final int = 12;
```

Final method:

A final keyword in a method that couldn't be overridden. If a method is marked as a final, then it can't be overridden by the subclass.

Final class:

If a class is declared as final, then the class couldn't be subclassed. No class can extend the final class.

Q #38) What is a Thread?

Ans: In Java, the flow of a execution is called Thread. Every java program has at least one thread called main thread, the Main thread is created by JVM. The user can define their own threads by extending Thread class (or) by implementing Runnable interface. Threads are executed concurrently.

Example:

```
public static void main(String[] args){//main thread starts here
}
```

Q #39) How do you make a thread in Java?

Ans: There are two ways available in order to make a thread.

#1) Extend Thread class:

Extending a Thread class and override the run method. The thread is available in java.lang.thread.

Example:

```
Public class Addition extends Thread {
public void run () {
}
}
```

The disadvantage of using a thread class is that we cannot extend any other classes because we have already extend the thread class. We can overload the run () method in our class.

#2) Implement Runnable interface:

Another way is implementing the runnable interface. For that we should provide the implementation for run () method which is defined in the interface.

Example:

```
Public class Addition implements Runnable {
public void run () {
}
}
```

Q #40) Explain about join () method.

Ans: Join () method is used to join one thread with the end of the currently running thread.

Example:

```
public static void main (String[] args){
Thread t = new Thread ();
t.start ();
t.join ();
}
```

From the above code, the main thread started the execution. When it reaches the code ***t.start()***then 'thread t' starts the own stack for the execution. JVM switches between the main thread and 'thread t'.

Once it reaches the code ***t.join()*** then 'thread t' alone is executed and completes its task, then only main thread started the execution.
It is a non-static method. Join () method has overloaded version. So we can mention the time duration in join () method also ".s".

Q #41) What does yield method of the Thread class do?

Ans: A yield () method moves the currently running thread to a runnable state and allows the other threads for execution. So that equal priority threads have a chance to run. It is a static method. It doesn't release any lock.
Yield () method moves the thread back to the Runnable state only, and not the thread to sleep (), wait () (or) block.

Example:

```
public static void main (String[] args){
Thread t = new Thread ();
t.start ();
}
public void run(){
Thread.yield();
}
}
```

Q #42) Explain about wait () method.

Ans: wait () method is used to make the thread to wait in the waiting pool. When a wait () method is executed during a thread execution then immediately the thread gives up the lock on the object and goes to the waiting pool. Wait () method tells the thread to wait for a given amount of time.
Then the thread will wake up after notify () (or) notify all () method is called.
Wait() and the other above-mentioned methods do not give the lock on the object immediately until the currently executing thread completes the synchronized code. It is mostly used in synchronization.

Example:

```
public static void main (String[] args){
Thread t = new Thread ();
t.start ();
Synchronized (t) {
Wait();
}
}
```

Q #43) Difference between notify() method and notifyAll() method in Java.

Ans: Given below are few differences between notify() method and notifyAll() method

notify()	notifyAll()
This method is used to send a signal to wake up a single thread in the waiting pool.	This method sends the signal to wake up all the threads in a waiting spool.

Q #44) How to stop a thread in java? Explain about sleep () method in a thread?

Ans: We can stop a thread by using the following thread methods.

- Sleeping
- Waiting
- Blocked

Sleep:

Sleep () method is used to sleep the currently executing thread for the given amount of time. Once the thread is wake up it can move to the runnable state. So sleep () method is used to delay the execution for some period.
It is a static method.

Example:

Thread. Sleep (2000)

So it delays the thread to sleep 2 milliseconds. Sleep () method throws an uninterrupted exception, hence we need to surround the block with try/catch.

```
public class ExampleThread implements Runnable{
public static void main (String[] args){
Thread t = new Thread ();
t.start ();
}
public void run(){
```

```
try{
Thread.sleep(2000);
}catch(InterruptedException e){
}
}
```

Q #45) When to use Runnable interface Vs Thread class in Java?

Ans: If we need our class to extend some other classes other than the thread then we can go with the runnable interface because in java we can extend only one class.

If we are not going to extend any class then we can extend the thread class.

Q #46) Difference between start() and run() method of thread class.

Ans: Start() method creates new thread and the code inside the run () method is executed in the new thread. If we directly called the run() method then a new thread is not created and the currently executing thread will continue to execute the run() method.

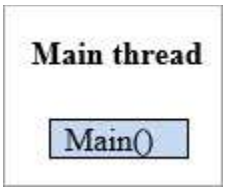
Q #47) What is Multi-threading?

Ans: Multiple threads are executed simultaneously. Each thread starts their own stack based on the flow (or) priority of the threads.

Example Program:

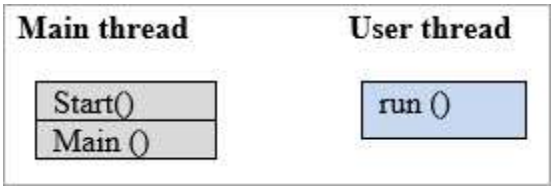
```
public class MultipleThreads implements Runnable
{
public static void main (String[] args){//Main thread starts here
Runnable r = new runnable ();
Thread t=new thread ();
t.start ();//User thread starts here
Addition add=new addition ();
}
public void run(){
go();
}//User thread ends here
}
```

On the 1st line execution, JVM calls the main method and the main thread stack looks as shown below.

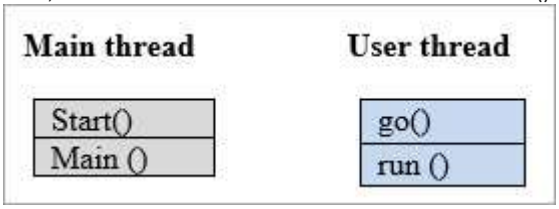


Once the execution reaches, **t.start ()** line then a new thread is created and the new stack for the thread is also created. Now JVM switches to the new thread and the main thread are back to the runnable state.

The two stacks look as shown below.



Now, the user thread executed the code inside the run() method.



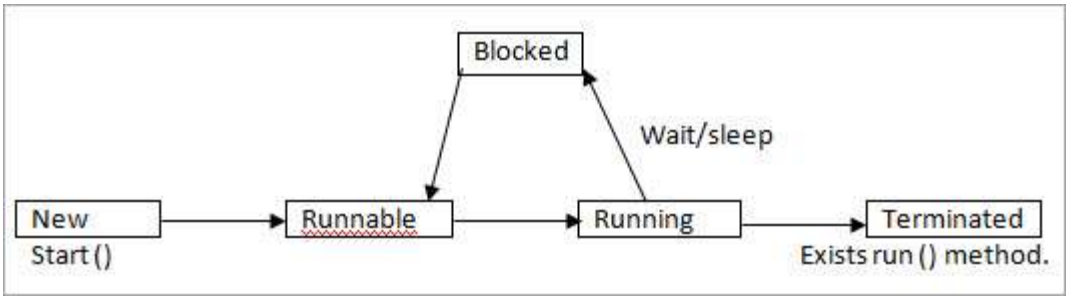
Once the run() method has completed, then JVM switches back to the main thread and the User thread has completed the task and the stack was disappeared.

JVM switches between each thread until both the threads are completed. This is called Multi-threading.

Q #48) Explain thread life cycle in Java.

Ans: Thread has the following states:

- New
- Runnable
- Running
- Non-runnable (Blocked)
- Terminated



- **New:**

In New state, Thread instance has been created but start () method is not yet invoked. Now the thread is not considered alive.

- **Runnable**:

The Thread is in runnable state after invocation of the start () method, but before the run () method is invoked. But a thread can also return to the runnable state from waiting/sleeping. In this state the thread is considered alive.

- **Running**:

The thread is in running state after it calls the run () method. Now the thread begins the execution.

- **Non-Runnable**(Blocked):

The thread is alive but it is not eligible to run. It is not in runnable state but also, it will return to runnable state after some time.
Example: wait, sleep, block.

- **Terminated** :

Once the run method is completed then it is terminated. Now the thread is not alive.

Q #49) What is Synchronization?

Ans: Synchronization makes only one thread to access a block of code at a time. If multiple thread accesses the block of code, then there is a chance for inaccurate results at the end. To avoid this issue, we can provide synchronization for the sensitive block of codes.
The synchronized keyword means that a thread needs a key in order to access the synchronized code.
Locks are per objects. Every Java object has a lock. A lock has only one key. A thread can access a synchronized method only if the thread can get the key to the objects lock.
For this, we use “Synchronized” keyword.

Example:

```
public class ExampleThread implements Runnable{
 public static void main (String[] args){
 Thread t = new Thread ();
 t.start ();
 }
 public void run(){
 synchronized(object){
 {
 }
}
```

Q #50) What is the disadvantage of Synchronization?

Ans: Synchronization is not recommended to implement all the methods. Because if one thread accesses the synchronized code then the next thread should have to wait. So it makes slow performance on the other end.

Q #51) What is meant by Serialization?

Ans: Converting a file into a byte stream is known as Serialization. The objects in the file is converted to the bytes for security purposes. For this, we need to implement java.io.Serializable interface. It has no method to define.
Variables that are marked as transient will not be a part of the serialization. So we can skip the serialization for the variables in the file by using a transient keyword.

Q #52) What is the purpose of a transient variable?

Ans: Transient variables are not part of the serialization process. During deserialization, the transient variables values are set to default value. It is not used with static variables.

Example:
transient int numbers;

Q #53) Which methods are used during Serialization and Deserialization process?

Ans: ObjectOutputStream and ObjectInputStream classes are higher level java.io. package. We will use them with lower level classes FileOutputStream and FileInputStream.
ObjectOutputStream.writeObject **—->**Serialize the object and write the serialized object to a file.
ObjectInputStream.readObject **—>** Reads the file and deserializes the object.
To be serialized, an object must implement the serializable interface. If superclass implements Serializable, then the subclass will automatically be serializable.

Q #54) What is the purpose of a Volatile Variable?

Ans: Volatile variable values are always read from the main memory and not from thread's cache memory. This is used mainly during synchronization. It is applicable only for variables.

Example:
volatile int number;

Q #55) Difference between Serialization and Deserialization in Java.

Ans: These are the difference between serialization and deserialization in java:

Serialization	Deserialization
Serialization is the process which is used to convert the objects into byte stream	Deserialization is the opposite process of serialization where we can get the objects back from the byte stream.
An object is serialized by writing it an ObjectOutputStream.	An object is deserialized by reading it from an ObjectInputStream.

Q #56) What is SerialVersionUID?Ans: Whenever an object is Serialized, the object is stamped with a version ID number for the object class. This ID is called the SerialVersionUID. This is used during deserialization to verify that the sender and receiver that are compatible with the Serialization.

Softway Solutions

JAVA

1.what is constructor?use of constructor in class?
2.why we need to compile a program?
3.upcasting in java?
4.polymorphism
5.singleTon class
6.overloading method

Selenium

1.what is testng?use of testng in framework?
2.how to handle alert,file upload popups
3.method overloading in Selenium?
4.access modifiers public,private,protected use in selenium
5.What is POM.XML
6.difference between .xml & .html
7.handling frames in selenium

Indeecom Global Services Interview questions

JAVA

1.what is polymorphism?types of polymorphism?

2.Access Modiefiers(Private,public,protected)

3.Object life cycle architecture?

4.singleTON Class?

5.Innerclass &Anonymous class?

6.how to print without main method "hello"?

7.Method overloading & overriding & Inheritance?

Selenium

1.what is selenium webdriver?advantage of against RC,IDE?

2.How to handle javascript popup?

3.How to handle iframes?

4.how to hadle filedownload popup?

5.what is the use of TestngListeners?

6.what is POM.XML?

7.what is ant & use in framework?

HCL -Automation testing and Manual testing

Automation Testing

---

1. Write a program to print your name in Java?

2. In the above program what does public static void main signify?

3. Write a program to add elements to a list?

4. What datatype can be added to a List?

5. How we can specify datatype of specific type in List?

6. Write a program to iterate through a List?

7. What is the difference between List and Map?

9. In Selenium, what is WebDriver - a class or interface?

10. Why we can't create instance of WebDriver directly?

11. What is the difference between findElement and findElements? Explain scenario when it is used?

12. Do you know to create a Framework for test execution ?

13. What is the version of Selenium used?

14. In Selenium write code to access FirefoxDriver?

Manual Testing

---

1. What is the difference between Whitebox and Blackbox testing?

2. What is Agile Methodology? What is Scrum,Sprint?Have you used it?

3. Who will be involved in Agile testing meeting?

4. What is DSR ,WSR?(status reports)

5. Have you done monthly Test plan?

6. Do you know Web testing?

7. Write a program to print to select all columns from Employees table and Departments table where Department name is "Testing"

8. How many years you have in Manual testing,Automation testing?

9. Have you done testing for Redbus application?

10. Write out some positive and negative scenarios to test "Add Friend" feature in Facebook?

11. What is the test design you have used to write test cases?

12. Imagine you are given a Bank Account Login Page. (userid,password,account no, Submit button). List out two positive scenarios to test it?

EBay:

What all collections you have used

What is the approch for HybridFW you used to develop.

Why KeywordDriven framework you have used.

Why Set<Iterator> you have used for handling window Handles

Internal Implementation of HashMap,ArrayList and Set Operator if you know ?

Have you written any Utilities.?

What will PageObject model will consists of ,how do you specify the elements in it.

Where all you have used ArrayList,Set and HashMap in your application.

What kind of dependencies you have added in the Maven and for what

What is your roles and responsibilities for automation.

Explain driver.getWindowHandles() in detail,how it will function.

Explain each usage of collection used in your project

what is smoke testing?

how can you test Mouse?

expline about defect tracking?

differance between sanity & smoke testing?

what is compatibility testing?

What is acceptance testing ?when we do?

difference between bug & defect?s?

what is automation test life cycle?

what is usability testing?

3. write functional test cases for " you have three field A, B, C and one ok button field can take only two character if by using the fields the triangle is formed then ok button must display valid triangle else invalid triangle.

5. some logical puzzles he asked like :-

a> how to measure 4 lt. if u have 2 container of 5 and 3 lt.

b> 3 bulb and 3 switch puzzle

c> car parking no. puzzle

6. what is the use of cookies and cache.

7. how to remove cookies

9. if you have opened any web application and it is broken means it has changed its layout and other thing which type of testing you will perform to check this

1)Which application you work on : Web Application or Windows Application?

2)WAP on java: fibonacci series, reverse a string, reverse a string using recursion?

3)Maven - What is Maven , Why do we use it and How do we use it

4)what is maven life cycle?

5)What is StringBuffer and StringBuilder

6)Explain Parallel exection using grid ?

7)Give examples of Polymorphism,Abstraction(simple code explaination)

8)why do we use singleton class?

9)Explain your framework

10)what is the use of PageFactory.initElements?Can we implement it using PageFactory.initElements or @FindBy?

11)How do we clean the Maven environment?

12)A particular list of buttons are there?Write the xpath for it?

13)How do you configure testNG?what do you write in the .bat file?

LATEST HAPPIEST MINDS:

Happiest Minds:

1.Explain your prevoius project, ur role

2.how u rate ur self in java?

3.Overloading and Overriding ...when u use overLoading and Overriding.

4.What is singleton and have used singleton concept in ur project ??

5.What is TestNG framework and what annotations u have worked on.

6.Write a syntax for the dataprovider in TestNg?

7.What are the frameworks u are aware of?

8.What are issues u faced in ur project while automation

9.Have u followed the agile methodology??What will u do if u find the bug at the end of the sprint and how fast will u make sure that the bug is fixed and verified.

10.Have u done compatibility testing ?? what all browsers u have used?

11.How do u run the scripts on other browser like IE,Chrome,write the code

12.what are exceptions u have handled in selenium webdriver

****

****

Cognizant Interview Question

1.Wat is the diff b/w WebDriver Listerner and TestNG Listener

2.diff b/w Assert and verify

3.count only the non duplicate words in the string

4.Frame Work and advantage of PageFactory classes

****

****

HCL Interview Questions

1.which frame work ur using and explain Framework

2.Questions on Nested frames Concept (Scenario given)

3.How ur using Agile Methodology in ur Project?

4.xpath Syntax

****

****

HCL interview Question

==========

1st technical

From Java

==========

1.What is the Difference between final,finally,finalize

2.what is the difference between Call by value and call by referance

3.How to find out the length of the string without using length function

4.How to find out the part of the string from a string

5.difference between throw & throws

6.What is binding(Early and Late binding) **

He give Programes

1.Reverse a number

2.1,2,3,4,5,65,76,5,,4,33,4,34,232,3,2323,

find the biggest number among these

simple string programe.

what is exception, types of exception

From manual

----

what is the testcase technique

why we write test case.

bug life cycle

what are the different status of bug

what is the different between functional and smoke testing

what is STLC.

from Selenium

==========

what is testng and its advantage

how to handle SSl/

how to handle alert

how to take screenshot

give the diagram write a scrpt..

tell me about Project .What are the challenge face during project

what is the difference between RC and webdriver

what is freamwork explain it.

why we use wait statement.

2nd technical

he gives a application & tell to write the scenario

some manual testing concepts.

==========

CaterPiller

1.Diff b/w final finally and finalize

2.how you will avoid object cloneable

3.wat is synchronize methods

4.wat finalize method will do

5.Quesion on Joints

6.Defect life Cycle

7.wat is Smoke Testing and Regression testing

8.Test case Design Techniques

9.how to run tests parallely in different browsers

10.Explain hashing technique***

11.How u will return multiple values in a method

EF Systems

1.how to run tests parallely in different browsers

2.java program to reverse String

3.pgm to sort array in ascending order

4.diff b/w implicit wait and explicit wait

5.Test case Design Techniques

6.dif b/w delete drop and truncate

7.wat is inheritence polymorphism and encapsulation

8.wat is interface

Happiest Minds

1.Explain ur FramwWork

2.How u will run tests parallely

3.pgm to search ipod in flipkart and sort the results in ascending order based on price

4.give examples for Checked exception and Runtime Exception.

5.Some time application takes 2 sec to load some time 30 sec some times more than 5 min how u will handle this.**

6.wat is maven and wat is ant

COGNIZANT Questions:

1. tell me abt urself

2. what is stlc

3. what is test planning

4. defect life cycle

5. what is inheritance

6. method overloading and overriding

7. framework & framework used in the company

8. what is Parametisation and how do you use it in TestNG

9. how do you execute the tests based on priority

10. what are the annotations

11. what is the order of executing the annotations

12. when a test suite is executed in the order

13. Tell me anything about collections

14. interfaces

15. what is the difference between interface and a class

16. what are arrays and list difference between them

17. Is parameterisation possible and how(Tell the coding)

18. how do you check whether a test script is passed or not

19. if a test case fails how will you log the failure and the exception that caused. how will you report the result to your team lead, prj manager and the customer

20. what is the framework you are using for logging the bugs

21. How do you submit all the test scripts, execution results to you TL, PM

22. what is the process followed in you company

23. what is the difference between agile methodology and spiral modal

24. what is requirement traceability matrix

25. how do hover a link which is present on web page which opens a pop up containing a link.how do you click on the link that is in th pop up

26. prgm for reading a excel file

27. how do you initialise a chrome browser

28. SQl-joins,difference between inner joins and outer joins

29. i am having a table which consists of student details,id, marks and how do i fetch the second maximum value from the table

30. I am having two tables one having student id, student name,and the college id and the other table having college id,college name, location now i want to get map these two tables and get the students who are studying in a specific college

31. i want the count of all the students who belongs to same location

---interview questions in cognizant

Infosys Interview Questions:

1.Brief Description

2.How will you rate yourself in Selenium.

3.SIDE

4.How will you handle pop-up.

5.Which framework you are using in your project & why.

6.How will you handle java script pop-up.

7.How will you handle SSL issue

8.How will you take screenshot.

9.How will you handle AJAX?Explain it with code.

10.Explain your framework.

11.What is STLC,Test Plan & Defect Life Cycle(Procedure to file a defect)

12.traceability matrix

13.Selenium Grid concepts with code ?RC

14.Project Description

15.Diff between Implicitly Wait and Explicitly Wait

16. When will you stop testing

17.Thread 7 Synchronization in java

18.why you switched three companies in 3 years.

1. What difficulty you faced in Automation?

2. What you do with your code when a new version of browser is launched?

3. What are the features in your automation code?

4. How to integrate HPALM/HPQC with Selenium?

5. How you are sending the emails from framework, as .rar files are blocked in most of the companies?

6. How code review is done for your code?

Erricson:

How selenium will select/click elements on the webpage.

How you will idenity elements.

How your testscript will look for login page.

If you have 10 testscripts,and in each testscript there are 5 TestNGMethods how you will execute.

If you want to verify data you selected in one Testmethod ,is same displayed in the another testmethod/webpage

(Ex:Shopping Cart)

What is Axes in the X path

How do you run tests using Testng (answer:xml)

How do you execute your fraamework from command prompt

java -cp bin;jars/* org.testng.TestNG <suitename>.xml

Tavant:

1) Tell me about ur current project and ur role it?

2) Java Oops Concepts?

3) We have two dropdown boxes if we select 1st same will be displaed in 2nd how u test it?

4) We have a table with different colomns if u click on title it will be sorted how u test it is sorted or not?

5) u don't have d.b access if u give data then it will take how u test that data is stored in d.b or not?(Regestration page)

6) i have 8 marbles of same size and color and same weight excepet one..how u find the odd one.?

7) how do u find a perticular word in a statement "Hi how 'are' you"?

8) Testng

9) Jenkins

10)Grid

11)parallel execution in diff browers ?

12) write complete program for..while login it shows "sucess" is displayed or not?

13)frame work?

14)defect life cycle.

HCL Interview Questions

1.which frame work ur using and explain Framework

2.Questions on Nested frames Concept (Scenario given)

3.How ur using Agile Methodology in ur Project?

4.xpath Syntax

Happiest Minds

1.Explain ur FrameWork

2.How u will run tests parallely

3.pgm to search ipod in flipkart and sort the results in ascending order based on price

4.give examples for Checked exception and Runtime Exception.

5.Some time application takes 2 sec to load some time 30 sec some times more than 5 min how u will handle this.

6.wat is maven and wat is ant

Maven is dependance tool (used to download latest jar from internet before running) ANT is Build tool

7.In a String remove duplicate words and duplicate characters.

8.what is checked exception and gve example

Exception which are verified by the compiler before the compilation and it makes it mandatory to try-catch or throws.

are checked exception example InvalidFormatException, IOException, InterruptedException

9.Script to find ipad in flipkart and the results should store in ascending order

Tavant

1..write a code for pagination testing.(need to identify one records in a table)

2.how will u handle frames?

3.how will u handle calendar popup scenario?

4.Explain about TestNG.

5.Explain about POM.

6.what are the factors should consider before doing automation?

CaterPiller

1.Diff b/w final finally and finalize

2.how you will avoid object cloneable

3.wat is synchronize methods

4.wat finalize method will do

5.Quesion on Joints

6.Defect life Cycle

7.wat is Smoke Testing and Regression testing

8.Test case Design Techniques

9.how to run tests parallely in different browsers

10.Explain hashing technique

11.How u will return multiple values in a method

EF Systems

1.how to run tests parallely in different browsers

2.java program to reverse String

3.pgm to sort array in ascending order

4.diff b/w implicit wait and explicit wait

5.Test case Design Techniques

6.dif b/w delete drop and truncate

7.wat is inheritence polymorphism and encapsulation

8.wat is interface

Cognizant Interview Question

1.Wat is the diff b/w WebDriver Listerner and TestNG Listener

Both are interface but WebDriver listener are triggered when you call webdriver methods like findElement,

click etc example-afterClickOn; beforeFindBy.... TestNG Listener are triggered when testng methods are executed such as

onTestFailure onFinish..

2.diff b/w Assert and verify

3.count only the non duplicate words in the string

4.Frame Work and advantage of PageFactory classes

3D PLM:

1. difference between smoke testing and adhoc testing.

2. difference between integration testing and system testing with examples.

3. write functional test cases for " you have three field A, B, C and one ok button field can take only two character if by using the fields the triangle is formed then ok button must display valid triangle else invalid triangle.

4. difference between http and https?

5. some logical puzzles he asked like :-

a> how to measure 4 lt. if u have 2 container of 5 and 3 lt.

b> 3 bulb and 3 switch puzzle

c> car parking no. puzzle

6. what is the use of cookies and cache.

7. how to remove cookies

8. how to test web application

9. if you have opened any web application and it is broken means it has changed its layout and other thing which type of testing you will perform to check this

What is severity and priority in software testing?

What is Stub?

What is abstraction ?

What is Encapsulation in Java and OOPS with Example

What is Functional Testing? What are the Different Techniques used in it

Black Box Testing: Types and techniques of BBT

SQL Questions

What is the difference between inner and outer join? Explain with example.

What is the difference between JOIN and UNION?

What is the difference between UNION and UNION ALL?

What is the difference between WHERE clause and HAVING clause?

What is the difference among UNION, MINUS and INTERSECT?

What is Self Join and why is it required?

How can we transpose a table using SQL (changing rows to column or vice-versa) ?

How can we transpose a table using SQL (changing rows to column or vice-versa) ?

How to select first 5 records from a table?

What is the difference between ROWNUM pseudo column and ROW_NUMBER() function?

What are the differences among ROWNUM, RANK and DENSE_RANK?

ACI Worldwide Interview Questions

1st Round-Written Test

Java

1. What is Polymorphism?

2. What is Abstraction?

3. Difference between static and non-static members?

4. Difference between super and this?

5. What is a List and Set?

6. What is Encapsulation?

7. What is Access Specifies in java?

8. Diagram and Asked to write Program?

Selenium

1. Code to select all the check boxes?

2. Different Waits in Selenium?

3. How do you make sure that Checkbox/Radio button is selected or not? Button Enabled or not?

4. How do you handle drop down list?

5. Code to select all the options present in multi select list box?

6. How to perform drag and drop of element and Right clicking on element?

7. How do you handle Alert popup?

8. How do you handle element present in Frames? Code to transfer control from frame to main page

9. How do u setup profile in Firefox browser?

10. What are Annotations?

11. Write a code to take screen shot for only failed test scripts?

12. Write a code to print number of links present on a webpage?

13. Code for Login page and how do u verify whether it is successfully logged in or not using assert?

2nd Round – Technical

1. Introduce yourself?

2. STLC?

3. What is Interface and Abstract Class?

4. Can we create an object of Abstract class?

5. What are the types of Automation frameworks we have? Can u explain them?

6. What is a sub Query?

7. What are Joins?

8. Queries on Joins and Sub-Query?

9. What is TestNG? Why we use it in Selenium?

10. Explain Automation Framework?

11. Program on Prime Number?

12. SDLC?

13. Program to Read and Write data in to Excel Sheet?

14. How to Right Click in Selenium?

Manager Round

1. About Yourself?

2. Why Testing U r from Electronics Background right?

3. Qualities of Good Test Engineer?

4. Challenges in Testing?

5. Objectives of Developer and Test Engineer?

6. About Family?

7. Explain frame Work?

QSG interview questions

1.Tell me about yourself

2.What is difference between array and string

3.What are different types you have in Microsoft Office

4.How do you create table in MS Office

5.How do you add column to a table

6.what is framework

7.write a prgm for sorting a string

8.Why we go for inheritance

9.what are different types of annotations expl @Groups

10.tell abt your s/m configuration

I want to select two different values in two different dropdown list(without using xpath). which has same ID . pls help..

```
List<WebElement> allListBox = driver.findElements(By.id("jasa"));
 for(int i=0;i<allListBox.size();i++)
 {
 WebElement listBox=allListBox.get(i);
 Select select=new Select(listBox);
 select.selectByIndex(i);
 }
```

HappyestMind& Ness

Selenium

1.Write the syntax of drop down

2.What is Webdriver-Java interface

3.What is the current Version of Selinumwebdriver

4.How to get the text value from text box

5.StrinG x="ABC";

String x="ab"; does it create two objects?

6.write a program to compare the strings

7.Class a

{

}

class b extends a

{

}

A a= new A();

B b=new B();

A a= new B();

B a=new A();

Which is valid and invalid?

8.How to handle differnt type of pop up.(Wnidow,Alerts,Invisiblepopup)

9.How to handle DropDown menu

10. How to handle SSL certificate

11.How to handle Google search text.

12. How to handle dynamic text box which contains the 3 numbers, get

the number and add the three number and add it other text box.

13.How to handle Ajax Objects

8.Explain webdriver architecture

9.Explain File downloading

10.Explain File attachments other that Auto IT

11.Write the syntax for finding the row count in dynamic web table

12.Differnece between class and Interface

13. What type of class is the string class

14.WHAT are the differnt methods that are used along with Xpath

15.Explain Interface

16 Explain Abstract

17.What is selenum grid

18 what is selenium RC

19.Why is key word drivern frame work only choosen,

1. how to handle dynamic object

2. how to work with button which is in div tag and and u have to click

without using xpath

3. JVM is dependent or independent platform

4.how many Test script you write in day

5. describe your framework

6. how to parameterized your junit

7.how to handle ssl security

8. how to handle window pops

9. diffnct between implicit and explicit

10.what are the types of assertion and what are assertion in junit

11.How to handle ssl certificate

12.What is dom concept

13.What is the challenges u have faced during Automation

14What is genrics

15.What is synchronization

Java

1.JVM is dependent or independent platform

2.diffn bwhashmap and hash set, set and linkedlist, arraylist and

vector list , linkedhash set and hashset

3.abstract and interface

4.throw and throws

5.how to split

6.checked and unchecked exception

7.how to work with ajax aplication

8.why sring is immutable

9.wat is the retruntype of getwindowhandles();

10.what are the types of assertion and what are assertion in java

11.differnce between interface and Abstract classes

12.What is static varaible

13.what is volatile

14. what is trainsient

15.what is the differnece between Final,Finalize and finally

16.what is the differnce between Public,private and protected

Click on Add to Firefox

Steps:

1) Right click on webelement and select Xpath

or

2) Right click on webelement and select Java locators

3) Click on Xpath and paste it into notepad

.

Its very useful to you to develop/create Xpaths and writing the Selenium code.

It saves lot of time

Prowareness Interview Questions

1. If there is a priority bug from production how do you address when the Sprint is in progress?

2. What is the level of involvement of Management in a Scrum team?

3. After a Sprint ends when the planning for next sprint begins?

4. Have you created VBA macros to update status of multiple test cases in QC?

5. I had DB2 exp in resume - so he asked me to write a join query invoving table A & B where only the values from table B be retrieved excluding those rows from B having matching rows in A.

6. What are the challenges you faced while creating framework?

7. Have you used Selenium Grid?

8. Have you run tests in parallel?

9. How will you wait until all the elements in a page gets loaded, the page is completely dynamic and you cannot guess which will be the last element to load.

10. When tests run in parallel how will Webdriver know that it has to work on a particular Browser window only?

11. When you move from one page to another how will you ensure that the proper page is loaded?

12. How will you drag and drop an element in a webpage?

13. If the element for which you are waiting for is not loaded, how will you handle the same?

14. How selenium interacts with Firefox, Chrome and IE (internal aspects)?

15. Have you used Javascript Executor in your scripting?

16. Why is main method Static?

17. Can the main method have return type other than void? (Yes it can, it can have return type int to provide info about successful execution, this info can be used in batch file/shell scripting - interviewers answer)

18. I was given a Unstable system of a Agile toolkit (its like our Jira) and was aked to list down the scenarios for testing Task Creation and editing the same. Also I was asked to find the bugs.

19. For the above scenario (point 18), I was asked to automate Task Creation and drag & drop of task from 'To Do' to 'In Progress'. This has to be done alongside with TestNG framework.

I was given 2hrs for performing point 18 & 19.

Genpact HeadStrong Questions

1. factorial,prime,armstorngnumber,swaping two numbers without temp var--Programs

2. interface,abstraction,encapsulation

3. diff between pagefactory n pageobject model

4. printstacktrace(exception related)

5. diff between string.equals and string1==string2

6. normalisation

7. searching with keyword and copy all the files name into one file

8. how grid works

9. how to debug or fix the issue

10. how to switch from one window to another window

11. mousemover concept

12. explain testNG framework

13. dataprovide,findby annotation

14. how to do exectuion of second method after first method method exectuion only without giving priority

15. what is diff between string,string buffer,string builder

16. diff between method overloading n method overriding

17. exception means? how do u recongnise where to fix in lengthy code if somewhere exception occurs while executing

18. diff between list-iterator and iterator

19. what is hashmap

20. how do u execute testcases using testng

21. how do u run batch execution, which one ur using for tht

22. how to print webtable in same structure on console

23. how do you switch multiple windows

24. how do you switch between tabs

25. How do you find webpage background color

1. How do you write xpath, if last three characters keeps on changing EX : "abc_ _ _".

2. Pagefactory concepts.

3. Collections, list concepts.

4. Have you involved in Framework developement?

5. How do you fetch data from webtable in a webpage?

6. In a webtable, how do you fetch only last column data?

7. How do you handle dynamic element?

8. What is concrete method?

9. Actions, windowhandle and Select concepts.

10 Are you using any build tool?

11. Grid concepts.

12. How much do you rate yourself in Java out of 5?

13. Which API used to fetch data from excel sheet?

Telephonic Questions of Happiest Minds..

1. When do you abstract class and when do you use interfaces.

2. Have you written / are there any interface in your automation frame work.

3. How do you run Java Script directly in Web Driver.

4, If the customer does not want DOM model of automation, what is your suggestion.

5. tell me about your framework.

6. How flexible is your framework.

7. Few Questions on Serialization, Threads and OOPS concepts.

Mind tree

what framework have you worked and used in automation?

How many methods are there in in your framework libraries?

What all methods in libraries?

How many test scripts are there for your application?

How often do you run your automation test?

what is getAttribute and getText?

what is the time taken to run all your scripts?

what all settings do you do before running your automation test?

How do you pass values to method?

who writes the test cases?

write a script to print the multiples of 5?

How do change the run time property values of the element?

How do handle dynamic elements during run time?

what version of selenium do you use and which is the latest version?

Suppose a webtable is there with three columns such as combo box,name and id.I want to search a particular id and also check the checkbox for that particular id.How would you do that?

FICO company technical questions for Core Java +Selenium

.what is the default package in java ?

2.: why we use interface why not abstract class ...what if i implements same method in interface and abstract ....thn ?? any diffrnc??

3. what are inner classes ..name them ?

4.in public static void main(String arr[])... what if i replace public with private ..... remove static .....replace void with string

5.in hash map we have (key and value) pair , can we store inside a value =(key, value) again ??

5. what are variable scope in java (in class , in method , in static block)

6. what are the oops concept ? explain them each with real world examples

7.

8. what is singleton classes ?

9.what is difference between .equals() , (==) and compare-to();

10. what is the difference between hash code and equals

11.write a program to get substring of string ex: javais good ... so result : avais

12.write a program to reverse the string

13. wap for binary search

14.what is the use of package

15. why we use interface and abstract

16.we have 2 interface both have print method , in my class i have implemented the print method , how u wil get to know that i have implemented the first interface and how u will use it .. if u want to use it

17.what is the difference between vector list and arraylist

18. difference between hashmap and hash table, what is synchronization , how it is achieved

19. what is the use of collection, when we use it

20. what is priority queue in collection , what is the use , how u have use in your project

21.where to use hashmap and hashtable

22. where u have use the concept of interface and abstract in your framework

HCL Interview Questions.

1. Code for Login page / how do you find and store the Elements / Locators ?

2. How to use the same Test Script for Localization testing (for a different language browser / different language settings).

3. Why do we upcast FirefoxDriver to Webdriver ?

4. How do you use @parameter Annotation in TestNg ?

5. How do you use load elements through properties file and run using TestNG ?

1. What are the annotations used in TestNG?

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

2. How do you read data from excel?

```
FileInputStream fis = new FileInputStream ("path of excel file");
Workbook wb = WorkbookFactory.create (fis);
```

```
Sheet s = wb.getSheet ("sheetName")
String value = s.getRow (rowNum).getCell (cellNum).getStringCellValue ();
```

3. What is the use of xpath?

It is used to find the WebElement in web page. It is very useful to identify the dynamic web elements.

4. What are different types of locators?

There are 8 types of locators and all are the static methods of the By class.

- By.id ()
- By.name ()
- By.tagName ()
- By.className ()
- By.linkText ()
- By.partialLinkText ()
- By.xpath
- By.cssSelector ()

5. What is the difference between Assert and Verify?

Assert it is used to verify the result. If the test case fails then it will stop the execution of the test case there itself and move the control to other test case.

Verify it is also used to verify the result. If the test case fails then it will not stop the execution of that test case.

6. What is the alternate way to click on login button?

Use submit () method but it can be used only when attribute type=submit.

7. How do you verify if the checkbox/radio is checked or not?

We can use isSelected () method.

Syntax – *driver.findElement (By.xpath ("xpath of the checkbox/radio button")).isSelected ();*

If the return value of this method is true then it is checked else it is not.

8. How do you handle alert popup?

To handle alert popup, we need to 1st switch control to alert popup then click on ok or cancel then move control back to main page.

Syntax:

```
String mainPage = driver.getWindowHandle ();
Alert alt = driver.switchTo ().alert (); // to move control to alert popup
alt.accept (); // to click on ok.
alt.dismiss (); // to click on cancel.
//Then move the control back to main web page
driver.switchTo ().window (mainPage); → to switch back to main page.
```

9. How do you launch IE/chrome browser?

Before launching IE or Chrome browser we need to set the System property.

```
//To open IE browser
System.setProperty ("webdriver.ie.driver","path of the iedriver.exe file");
Web Driver driver = new InternetExplorerDriver ();
//To open Chrome browser → System.setProperty ("webdriver.chrome.driver","path of the chromeDriver.exe file");
WebDriver driver = new ChromeDriver ();
```

10. How to perform right click using WebDriver?

Use Actions class

```
Actions act = new Actions (driver); // where driver is WebDriver type
act.moveToElement(WebElement).perform();
act.contextClick ().perform ();
```

11. How do perform drag and drop using WebDriver?

```
Use Action class
Actions act = new Actions (driver);
WebElement source = driver.findElement (By.xpath (“”)); //source ele which you want to drag
```

```
WebElement target = driver.findElement (By.xpath (“”)); //target where you want to drop
act.dragAndDrop(source, target).perform();
```

12. Give the example for method overload in WebDriver.

Frame (string), frame (int), and frame (WebElement).

13. How do you upload a file?

To upload a file we can use sendKeys() method.

driver.findElement (By.xpath (“input field”)).sendKeys (“path of the file which u want to upload”);

14. How do you click on a menu item in a drop down menu?

If that menu has been created by using select tag then we can use the methods selectByValue () or selectByIndex () or selectByVisibleText (). These are the methods of the Select class.

If the menu has not been created by using the select tag then we can simply find the xpath of that element and click on that to select.

15. How do you simulate browser back and forward?

```
Driver. Navigate ().back ();
Driver. Navigate ().forward ();
```

16. How do you get the current page URL?

driver.getCurrentUrl ();

17. What is the difference between ‘/’ and ‘//’?

// It is used to search in the entire structure.

/ It is used to identify the immediate child.

18. What is the difference between find Element and find Elements?

Both methods are abstract method of WebDriver interface and used to find the WebElement in a web page.

Find Element () – it used to find the one web element. It return only one WebElement type.

FindElements () it used to find more than one web element. It returns List of WebElement.

19. How do you achieve synchronization in WebDriver?

We can use implicit wait.

Syntax driver.manage ().timeouts ().implicitly Wait (10, TimeUnit.SECONDS);

Here it will wait for 10sec if while execution driver did not find the element in the page immediately. This code will attach with each and every line of the script automatically. It is not required to write every time. Just write it once after opening the browser.

20. Write the code for Reading and Writing to Excel through Selenium?

```
FileInputStream fis = new FileInputStream (“path of excel file”);
Workbook wb = WorkbookFactory.create (fis);
Sheet s = wb.getSheet (“sheetName”);
String value = s.getRow (rowNum).getCell (cellNum).getStringCellValue (); // read data
s.getRow (rowNum).getCell (cellNum).setCellValue (“value to be set”); //write data
FileOutputStream FOS = new FileOutputStream (“path of file”);
wb.write (FOS); //save file
```

21. How to get typed text from a textbox?

```
Use get Attribute (“value”) method by passing arg as value.
String typedText = driver.findElement (By.xpath (“xpath of box”)).get Attribute (“value”));
```

22. What are the different exceptions you got when working with WebDriver?

- ElementNotVisibleException,
- ElementNotSelectableException,
- NoAlertPresentException,
- NoSuchAttributeException,
- NoSuchWindowException,
- TimeoutException,
- WebDriverException

23. What are the languages supported by WebDriver?

Python, Ruby, C# and Java are all supported directly by the development team. There are also WebDriver implementations for PHP and Perl.

24. How do you clear the contents of a textbox in selenium?

Use clear () method.
driver.findElement (By.xpath ("xpath of box")).clear ();

25. What is a Framework?

A framework is set of automation guidelines which help in
Maintaining consistency of Testing, Improves test structuring, Minimum usage of code, Less Maintenance of code, Improve reusability, Non Technical testers can be involved in code, Training period of using the tool can be reduced, Involves Data wherever appropriate.
There are five types of framework used in software automation testing:

- Data Driven Automation Framework
- Method Driven Automation Framework
- Modular Automation Framework
- Keyword Driven Automation Framework
- Hybrid Automation Framework, it's basically combination of different frameworks. (1+2+3).

26. What are the prerequisites to run selenium WebDriver?

JDK, Eclipse, WebDriver (selenium standalone jar file), browser, application to be tested.

27. What are the advantages of selenium WebDriver?

a) It supports with most of the browsers like Firefox, IE, Chrome, Safari, Opera etc.
b) It supports with most of the language like Java, Python, Ruby, C# etc.
b) Doesn't require to start server before executing the test script.
c) It has actual core API which has binding in a range of languages.
d) It supports of moving mouse cursors.
e) It supports to test iphone/Android applications.

28. What is WebDriverBackedSelenium?

WebDriverBackedSelenium is a kind of class name where we can create an object for it as below:
Selenium WebDriver= new WebDriverBackedSelenium (WebDriver object name, "URL path of website")
The main use of this is when we want to write code using both WebDriver and Selenium RC , we must use above created object to use selenium commands.

29. How to invoke an application in WebDriver?

driver. get ("url"); or driver. Navigate ().to ("url");

30. What is Selenium Grid?

Selenium Grid allows you to run your tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines, different browsers and operating systems. Essentially, Selenium Grid support distributed test execution. It allows for running your tests in a distributed test execution environment.

31. How to get the number of frames on a page?

List < WebElement> frames List = driver.findElements (By.xpath ("//iframe"));
int numOfFrames = frameList.size ();

32. How do you simulate scroll down action?

```
JavaScript Executor jsx = (JavaScript Executor) driver;
jsx.executeScript ("window.scrollBy (0, 4500)", ""); //scroll down, value 4500 you can change as per your req
jsx.executeScript ("window.scrollBy (450,0)", ""); //scroll up
public class Scroll Down {
public static void main (String[] args) throws Interrupted Exception {
WebDriver driver = new Firefox Driver ();
driver.manage ().timeouts ().implicitly Wait (10, TimeUnit.SECONDS);
}
}
```

33. What is the command line we have to write inside a .bat file to execute a selenium project when we are using TestNG?

Java cp bin; jars/* org.testng.TestNG testng.xml

34. Which is the package which is to be imported while working with WebDriver?

org.openqa.selenium

35. How to check if an element is visible on the web page?

Use is Displayed() method.The return type of the method is Boolean.So if it return true then element is visible else not visible.
driver.findElement (By.xpath ("xpath of element")).is Displayed ();

36. How to check if a button is enabled on the page?

Use is Enabled () method. The return type of the method is Boolean. So if it return true then button is enabled else not enabled.

driver.findElement (By.xpath (“xpath of button”)).is Enabled ();

37. How to check if a text is highlighted on the page ?

To identify weather color for a field is different or not,

```
String color = driver.findElement (By.xpath (“//a [text () ='Shop']”)).getCssValue (“color”);
String backcolor = driver.findElement (By.xpath (“//a [text () ='Shop']”)).getCssValue (“backgroundcolor”);
System.out.println (color);
System.out.println (backcolor);
```

Here if both color and back color different then that me that element is in different color.

38. How to check the checkbox or radio button is selected?

Use isSelected () method to identify. The return type of the method is Boolean. So if it return true then button is selected else not enabled.

driver.findElement (By.xpath (“xpath of button”)).isSelected ();

39. How to get the title of the page?

Use getTitle () method.

Syntax: driver.getTitle ();

40. How does you get the width of the textbox?

```
driver.findElement (By.xpath (“xpath of textbox”)).getSize().getWidth ();
driver.findElement (By.xpath (“xpath of textbox”)).getSize().getHeight();
```

41. How do you get the attribute of the web element?

driver.getElement (By.tagName (“img”)).getAttribute(“src”) will give you the src attribute of this tag. Similarly, you can get the values of attributes such as title, alt etc.

Similarly you can get CSS properties of any tag by using getCssValue (“some property name”).

42. How to check whether a text is underlined or not?

Identify by getCssValue (“borderbottom”) or sometime getCssValue (“text decoration”) method if the CssValue is ‘underline’ for that WebElement or not.

ex : This is for when moving cursor over element that is going to be underlined or not

```
public class UnderLine {
public static void main (String [] args) {
WebDriver driver = new Firefox Driver();
driver.manage ().timeouts ().implicitly Wait (10, TimeUnit.SECONDS);
driver.get (“https://www.google.co.in/?gfe_rd=ctrl&ei=bXAwU8jYN4W6iAf8zIDgDA&gws_rd=cr“);
String cssValue= driver.findElement (By.xpath (“//a[text()='Hindi']”)).getCssValue(“textdecoration”);
System.out.println (“value”+cssValue);
Actions act = new Actions(driver);
act.moveToElement (driver.findElement (By.xpath (“//a[text()='Hindi']”))).perform();
String cssValue1= driver.findElement (By.xpath (“//a[text()='Hindi']”)).getCssValue(“textdecoration”);
System.out.println (“value over”+cssValue1);
driver.close ();
}
}
```

43. How to change the URL on a webpage using selenium web driver ?

driver. get (“url1”);

driver. get (“url2”);

44. How to hover the mouse on an element?

Actions act = new Actions (driver);

act.moveToElement (WebElement); //WebElement on which you want to move cursor

45. What is the use of get Options () method?

get Options () is used to get the selected option from the dropdown list.

46. What is the use of deselect All () method?

It is used to deselect all the options which have been selected from the dropdown list.

47. Is WebElement an interface or a class?

WebDriver is an Interface.

48. Firefox Driver is class or an interface and from where is it inherited ?

Firefox Driver is a class. It implements all the methods of WebDriver interface.

49. What is the difference b/w close () and quit ()?

close () – it will close the browser where the control is.
quit () – it will close all the browsers opened by WebDriver.

50. Can Selenium handle windows based pop up?

Selenium is an automation testing tool which supports only web application testing. Therefore, windows pop up cannot be handled using Selenium.

1. What is Selenium?

Selenium is a robust test automation suite that is used for automating web based applications. It supports multiple browsers, programming languages and platforms.

2. What are different forms of selenium?

Selenium comes in four forms-
Selenium WebDriver – Selenium WebDriver is used to automate web applications using browser's native methods.
Selenium IDE – A firefox plugin that works on record and play back principle.
Selenium RC – Selenium Remote Control(RC) is officially deprecated by selenium and it used to work on javascript to automate the web applications.
Selenium Grid – Allows selenium tests to run in parallel across multiple machines.

Interview Questions on Selenium

3. What are some advantages of selenium?

Following are the advantages of selenium-
Selenium is open source and free to use without any licensing cost.
It supports multiple languages like Java, ruby, python etc.
It supports multi browser testing.
It has good amount of resources and helping community over the internet.
Using selenium IDE component, non-programmers can also write automation scripts
Using selenium grid component, distributed testing can be carried out on remote machines possible.

4. What are some limitations of selenium?

Following are the limitations of selenium-
We cannot test desktop application using selenium.
We cannot test web services using selenium.
For creating robust scripts in selenium webdriver, programming langauge knowledge is required.
We have to rely on external libraries and tools forperforming tasks like – logging(log4J), testing framework-(testNG, JUnit), reading from external files(POI for excels) etc.

5. Which all browsers are supported by selenium webdriver?

Some commonly used browsers supported by selenium are-
Google Chrome – ChromeDriver
Firefox – FireFoxDriver
Internet Explorer – InternetExplorerDriver
Safari – SafariDriver
HtmlUnit (Headless browser) – HtmlUnitDriver
Android – Selendroid/Appium
IOS – ios-driver/Appium

6. Can we test APIs or web services using selenium webdriver?

No selenium webdriver uses browser's native method to automate the web applications. Since web services are headless, so we cannot automate web services using selenium webdriver.

7. What are the testing type supported by Selenium WebDriver?
Selenium webdriver can be used for performing automated functional and regression testing.
8. What are various ways of locating an element in selenium?
The different locators in selenium are:
Id
XPath
cssSelector
className
tagName
name
linkText
partialLinkText
9. What is an XPath?
Xpath or XML path is a query language for selecting nodes from XML documents. XPath is one of the locators supported by selenium webdriver.
10. What is an absolute XPath?
An absolute XPath is a way of locating an element using an XML expression beginning from root node i.e. html node in case of web pages. The main disadvantage of absolute xpath is that even with slightest change in the UI or any element the whole absolute XPath fails.
Example – html/body/div/div[2]/div/div/div/div[1]/div/input
11. What is a relative XPath?
A relative XPath is a way of locating an element using an XML expression beginning from anywhere in the HTML document. There are different ways of creating relative XPaths which are used for creating robust XPaths (unaffected by changes in other UI elements).
Example – //input[@id='username']
12. What is the difference between single slash(/) and double slash(//) in XPath?
In XPath a single slash is used for creating XPaths with absolute paths beginning from root node.
Whereas double slash is used for creating relative XPaths.
13. How can we inspect the web element attributes in order to use them in different locators?
Using Firebug or developer tools we can inspect the specific web elements.
Firebug is a plugin of firefox that provides various development tools for debugging applications. From automation perspective, firebug is used specifically for inspecting web-elements in order to use their attributes like id, class, name etc. in different locators.
14. How can we locate an element by only partially matching its attributes value in Xpath?
Using contains() method we can locate an element by partially matching its attribute's value. This is particularly helpful in the scenarios where the attributes have dynamic values with certain constant part.
```
xPath expression = //*[contains(@name,'user')]
```
The above statement will match the all the values of name attribute containing the word 'user' in them.
15. How can we locate elements using their text in XPath?
Using the text() method –
```
xPathExpression = //*[text()='username']
```
16. How can we move to parent of an element using XPath?
Using '..' expression in XPath we can move to parent of an element.
17. How can we move to nth child element using XPath?
There are two ways of navigating to the nth element using XPath-
Using square brackets with index position-
Example – div[2] will find the second div element.
Using position()-
Example – div[position()=3] will find the third div element.
18. What is the syntax of finding elements by class using CSS Selector?
By .className we can select all the element belonging to a particluar class e.g. '.red' will select all elements having class 'red'.
19. What is the syntax of finding elements by id using CSS Selector?
By #idValue we can select all the element belonging to a particluar class e.g. '#userId' will select the element having id – userId.
20. How can we select elements by their attribute value using CSS Selector?
Using [attribute=value] we can select all the element belonging to a particluar class e.g. '[type=small]' will select the element having attribute type of value 'small'.
21. How can we move to nth child element using css selector?
Using :nth-child(n) we can move to the nth child element e.g. div:nth-child(2) will locate 2nd div element of its parent.

22. What is fundamental difference between XPath and css selector?
The fundamental difference between XPath and css selector is using XPaths we traverse up in the document i.e. we can move to parent elements. Whereas using CSS selector we can only move downwards in the document.
23. How can we launch different browsers in selenium webdriver?
By creating an instance of driver of a particular browser-

```
WebDriver driver = new FirefoxDriver();
```

24. What is the use of driver.get("URL") and driver.navigate().to("URL") command? Is there any difference between the two?
Both driver.get("URL") and driver.navigate().to("URL") commands are used to navigate to a URL passed as parameter.
There is no difference between the two commands.
25. How can we type text in a textbox element using selenium?
Using sendKeys() method we can type text in a textbox-

```
WebElement searchTextBox = driver.findElement(By.id("search"));
searchTextBox.sendKeys("searchTerm");
```

26. How can we clear a text written in a textbox?
Using clear() method we can delete the text written in a textbox.

```
driver.findElement(By.id("elementLocator")).clear();
```

27. How to check a checkBox in selenium?
The same click() method used for clicking buttons or radio buttons can be used for checking checkbox as well.
28. How can we submit a form in selenium?
Using submit() method we can submit a form in selenium.

```
driver.findElement(By.id("form1")).submit();
```

Also, the click() method can be used for the same purpose.
29. Explain the difference between close and quit command.
driver.close() – Used to close the current browser having focus
driver.quit() – Used to close all the browser instances
30. How to switch between multiple windows in selenium?
Selenium has driver.getWindowHandles() and driver.switchTo().window("{windowHandleName}") commands to work with multiple windows. The getWindowHandles() command returns a list of ids corresponding to each window and on passing a particular window handle to driver.switchTo().window("{windowHandleName}") command we can switch control/focus to that particular window.

```
for (String windowHandle : driver.getWindowHandles()) {
   driver.switchTo().window(handle);
}
```

31. What is the difference between driver.getWindowHandle() and driver.getWindowHandles() in selenium?
driver.getWindowHandle() returns a handle of the current page (a unique identifier)
Whereas driver.getWindowHandles() returns a set of handles of the all the pages available.
32. How can we move to a particular frame in selenium?
The driver.switchTo() commands can be used for switching to frames.

```
driver.switchTo().frame("{frameIndex/frameId/frameName}");
```

For locating a frame we can either use the index (starting from 0), its name or Id.
33. Can we move back and forward in browser using selenium?
Yes, using driver.navigate().back() and driver.navigate().forward() commands we can move backward and forward in a browser.
34. Is there a way to refresh browser using selenium?
There a multiple ways to refresh a page in selenium-
Using driver.navigate().refresh() command
Using sendKeys(Keys.F5) on any textbox on the webpage
Using driver.get("URL") on the current URL or using driver.getCurrentUrl()
Using driver.navigate().to("URL") on the current URL or driver.navigate().to(driver.getCurrentUrl());
35. How can we maximize browser window in selenium?
We can maximize browser window in selenium using following command-

```
driver.manage().window().maximize();
```

36. How can we fetch a text written over an element?
Using getText() method we can fetch the text over an element.

```
String text = driver.findElement("elementLocator").getText();
```

37. How can we find the value of different attributes like name, class, value of an element?
Using getAttribute("{attributeName}") method we can find the value of different attrbutes of an element e.g.-
String valueAttribute =
driver.findElement(By.id("elementLocator")).getAttribute("value");

38. How to delete cookies in selenium?
Using deleteAllCookies() method-
driver.manage().deleteAllCookies();
39. What is an implicit wait in selenium?
An implicit wait is a type of wait which waits for a specified time while locating an element before throwing NoSuchElementException. As by default selenium tries to find elements immediately when required without any wait. So, it is good to use implicit wait. This wait is applied to all the elements of the current driver instance.
driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
40. What is an explicit wait in selenium?
An explicit wait is a type of wait which is applied to a particular web element untill the expected condition specified is met.
WebDriverWait wait = new WebDriverWait(driver, 10);
WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("elementId")));
41. What are some expected conditions that can be used in Explicit waits?
Some of the commonly used expected conditions of an element that can be used with expicit waits are-
elementToBeClickable(WebElement element or By locator)
stalenessOf(WebElement element)
visibilityOf(WebElement element)
visibilityOfElementLocated(By locator)
invisibilityOfElementLocated(By locator)
attributeContains(WebElement element, String attribute, String value)
alertIsPresent()
titleContains(String title)
titleIs(String title)
textToBePresentInElementLocated(By, String)
42. What is fluent wait in selenium?
A fluent wait is a type of wait in which we can also specify polling interval(intervals after which driver will try to find the element) along with the maximum timeout value.

```
Wait wait = new FluentWait(driver)
    .withTimeout(20, SECONDS)
    .pollingEvery(5, SECONDS)
    .ignoring(NoSuchElementException.class);
  WebElement textBox = wait.until(new Function() {
    public WebElement apply(WebDriver driver) {
    return driver.findElement(By.id("textBoxId"));
    }
}
);
```

43. What are the different keyboard operations that can be performed in selenium?
The different keyboard operations that can be performed in selenium are-
.sendKeys("sequence of characters") – Used for passing charcter sequesnce to an input or textbox element.
.pressKey("non-text keys") – Used for keys like control, function keys etc that ae non text.
.releaseKey("non-text keys") – Used in conjuntion with keypress event to simulate releasing a key from keyboard event.
44. What are the different mouse actions that can be performed?
The different mouse evenets supported in selenium are
click(WebElement element)
doubleClick(WebElement element)
contextClick(WebElement element)
mouseDown(WebElement element)
mouseUp(WebElement element)
mouseMove(WebElement element)
mouseMove(WebElement element, long xOffset, long yOffset)
45. Write the code to double click an element in selenium?
Code to double click an element in selenium-

```
Actions action = new Actions(driver);
WebElement element=driver.findElement(By.id("elementId"));
action.doubleClick(element).perform();
```

46. Write the code to right click an element in selenium?
Code to right click an element in selenium-

Actions action = new Actions(driver);
WebElement element=driver.findElement(By.id("elementId"));
action.contextClick(element).perform();
47. How to mouse hover an element in selenium?
Code to mouse hover over an element in selenium-
Actions action = new Actions(driver);
WebElement element=driver.findElement(By.id("elementId"));
action.moveToElement(element).perform();
48. How to fetch the current page URL in selenium?
Using getCurrentURL() command we can fetch the current page URL-
driver.getCurrentUrl();
49. How can we fetch title of the page in selenium?
Using driver.getTitle(); we can fetch the page title in selenium. This method returns a string containing the title of the webpage.
50. How can we fetch the page source in selenium?
Using driver.getPageSource(); we can fetch the page source in selenium. This method returns a string containing the page source.
51. How to verify tooltip text using selenium?
Tooltips webelements have an attribute of type 'title'. By fetching the value of 'title' attribute we can verify the tooltip text in selenium.
String toolTipText = element.getAttribute("title");
52. How to locate a link using its text in selenium?
Using linkText() and partialLinkText() we can locate a link. The difference between the two is linkText matches the complete string passed as parameter to the link texts. Whereas partialLinkText matches the string parameter partially with the link texts.
WebElement link1 = driver.findElement(By.linkText("artOfTesting"));
WebElement link2 = driver.findElement(By.partialLinkText("artOf"));
53. What are DesiredCapabilities in selenium webdriver?
Desired capabilities are a set of key-value pairs that are used for storing or configuring browser specific properties like its version, platform etc in the browser instances.
54. How can we find all the links on a web page?
All the links are of anchor tag 'a'. So by locating elements of tagName 'a' we can find all the links on a webpage.
List links = driver.findElements(By.tagName("a"));
55. What are some commonly encountered exceptions in selenium?
Some of the commonly seen exception in selenium are-
NoSuchElementException – When no element could be located from the locator provided.
ElementNotVisibleException – When element is present in the dom but is not visible.
NoAlertPresentException – When we try to switch to an alert but the targetted alert is not present.
NoSuchFrameException – When we try to switch to a frame but the targetted frame is not present.
NoSuchWindowException – When we try to switch to a window but the targetted window is not present.
UnexpectedAlertPresentException – When an unexpected alert blocks normal interaction of the driver.
TimeoutException – When a command execution gets timeout.
InvalidElementStateException – When the state of an element is not appropriate for the desired action.
NoSuchAttributeException – When we are trying to fetch an attribute's value but the attribute is not correct
WebDriverException – When there is some issue with driver instance preventing it from getting launched.
56. How can we capture screenshots in selenium?
Using getScreenshotAs method of TakesScreenshot interface we can take the screenshots in selenium.
File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(scrFile, new File("D:\\testScreenShot.jpg"));
57. How to handle dropdowns in selenium?
Using Select class-
Select countriesDropDown = new Select(driver.findElement(By.id("countries")));
dropdown.selectByVisibleText("India");
//or using index of the option starting from 0
dropdown.selectByIndex(1);
//or using its value attribute
dropdown.selectByValue("Ind");
58. How to check which option in the dropdown is selected?
Using isSelected() method we can check the state of a dropdown's option.
Select countriesDropDown = new Select(driver.findElement(By.id("countries")));

dropdown.selectByVisibleText(“India”);
//returns true or false value
System.out.println(driver.findElement(By.id(“India”)).isSelected());
59. How can we check if an element is getting displayed on a web page?
Using isDisplayed method we can check if an element is getting displayed on a web page.
driver.findElement(By locator).isDisplayed();
60. How can we check if an element is enabled for interaction on a web page?
Using isEnabled method we can check if an element is enabled or not.
driver.findElement(By locator).isEnabled();
61. What is the difference between driver.findElement() and driver.findElements() commands?
The difference between driver.findElement() and driver.findElements() commands is- findElement() returns a single WebElement (found first) based on the locator passed as parameter. Whereas findElements() returns a list of WebElements, all satisfying the locator value passed.
Syntax of findElement()-
WebElement textbox = driver.findElement(By.id(“textBoxLocator”));
Syntax of findElements()-
List elements = element.findElements(By.id(“value”));
Another difference between the two is- if no element is found then findElement() throws NoSuchElementException whereas findElements() returns a list of 0 elements.
62. Explain the difference between implicit wait and explicit wait.?
An implicit wait, while finding an element waits for a specified time before throwing NoSuchElementException in case element is not found. The timeout value remains valid throughout the webDriver’s instance and for all the elements.
driver.manage().timeouts().implicitlyWait(180, TimeUnit.SECONDS);
Whereas, Explicit wait is applied to a specified element only-
WebDriverWait wait = new WebDriverWait(driver, 5);
wait.until(ExpectedConditions.presenceOfElementLocated(ElementLocator));
It is advisable to use explicit waits over implicit waits because higher timeout value of implicit wait set due to an element that takes time to be visible gets applied to all the elements. Thus increasing overall execution time of the script. On the other hand, we can apply different timeouts to different element in case of explicit waits.
63. How can we handle window UI elements and window POP ups using selenium?
Selenium is used for automating Web based application only(or browsers only). For handling window GUI elements we can use AutoIT. AutoIT is a freeware used for automating window GUI. The AutoIt scripts follow simple BASIC lanaguage like syntax and can be easily integrated with selenium tests.
64. What is Robot API?
Robot API is used for handling Keyboard or mouse events. It is generally used to upload files to the server in selenium automation.
Robot robot = new Robot();
//Simulate enter key action
robot.keyPress(KeyEvent.VK_ENTER);
65. How to do file upload in selenium?
File upload action can be performed in multiple ways-
Using element.sendKeys(“path of file”) on the webElement of input tag and type file i.e. the elements should be like –
<input type=”file” name=”fileUpload”>
Using Robot API.
Using AutoIT API.
66. How to handle HTTPS website in selenium? or How to accept the SSL untrusted connection?
Using profiles in firefox we can handle accept the SSL untrusted connection certificate. Profiles are basically set of user preferences stored in a file.
FirefoxProfile profile = new FirefoxProfile();
profile.setAcceptUntrustedCertificates(true);
profile.setAssumeUntrustedCertificateIssuer(false);
WebDriver driver = new FirefoxDriver(profile);
67 How to do drag and drop in selenium?
Using Action class, drag and drop can be performed in selenium. Sample code-
Actions builder = new Actions(driver);
Action dragAndDrop = builder.clickAndHold(SourceElement)
.moveToElement(TargetElement)
.release(TargetElement)

```
.build();
dragAndDrop.perform();
```
68. How to execute javascript in selenium?
JavaScript can be executed in selenium using JavaScriptExecuter. Sample code for javascript execution-
```
WebDriver driver = new FireFoxDriver();
if (driver instanceof JavascriptExecutor) {
((JavascriptExecutor)driver).executeScript("{JavaScript Code}");
}
```
69. How to handle alerts in selenium?
In order to accept or dismiss an alert box the alert class is used. This requires first switching to the alert box and than using accept() or dismiss() command as the case may be.
```
Alert alert = driver.switchTo().alert();
//To accept the alert
alert.accept();
Alert alert = driver.switchTo().alert();
//To cancel the alert box
alert.dismiss();
```
70. What is HtmlUnitDriver?
HtmlUnitDriver is the fastest WebDriver. Unlike other drivers (FireFoxDriver, ChromeDriver etc), the HtmlUnitDriver is non-GUI, while running no browser gets launched.
71. How to handle hidden elements in Selenium webDriver?
Using javaScript executor we can handle hidden elements-
```
(JavascriptExecutor(driver)) .executeScript("document.getElementsByClassName(ElementLocator).click();");
```
72. What is Page Object Model or POM?
Page Object Model(POM) is a design pattern in selenium. A design pattern is a solution or a set of standards that are used for solving commonly occuring software problems.
Now coming to POM – POM helps to create a framework for maintaining selenium scripts. In POM for each page of the application a class is created having the web elements belonging to the page and methods handling the events in that page. The test scripts are maintained in seperate files and the methods of the page object files are called from the test scripts file.
73. What are the advantages of POM?
The advantages are POM are-
Using POM we can create an Object Repository, a set of web elements in seperate files along with their associated functions. Thereby keeping code clean.
For any chnage in UI(or web elements) only page object files are required to be updated leaving test files unchanged.
It makes code reusable and maintable.
74. What is Page Factory?
Page factory is an implementation of Page Object Model in selenium. It provides @FindBy annotation to find web elements and PageFactory.initElements() method to initialize all web elements defined with @FindBy annotation.
```
public class SamplePage {
   WebDriver driver;
   @FindBy(id="search")
   WebElement searchTextBox;
   @FindBy(name="searchBtn")
   WebElement searchButton;
   //Constructor
   public samplePage(WebDriver driver){
      this.driver = driver;
      //initElements method to initialize all elements
      PageFactory.initElements(driver, this);
   }
   //Sample method
   public void search(String searchTerm){
      searchTextBox.sendKeys(searchTerm);
      searchButton.click();
   }
}
```

75. What is an Object repository?
An object repository is centralized location of all the object or WebElements of the test scripts. In selenium we can create object repository using Page Object Model and Page Factory design patterns.
76. What is a data driven framework?
A data driven framework is one in which the test data is put in external files like csv, excel etc separated from test logic written in test script files. The test data drives the test cases, i.e. the test methods run for each set of test data values. TestNG provides inherent support for data driven testing using @dataProvider annotation.
77. What is a keyword driven framework?
A keyword driven framework is one in which the actions are associated with keywords and kept in external files e.g. an action of launching a browser will be associated with keyword – launchBrowser(), action to write in a textbox with keyword – writeInTextBox(webElement, textToWrite) etc. The code to perform the action based on a keyword specified in external file is implemented in the framework itself.
In this way the test steps can be written in a file by even a person of non-programming background once all the identified actions are implemented.
78. What is a hybrid framework?
A hybrid framework is a combination of one or more frameworks. Normally it is associated with combination of data driven and keyword driven frameworks where both the test data and test actions are kept in external files(in the form of table).
79. What is selenium Grid?
Selenium grid is a tool that helps in distributed running of test scripts across different machines having different browsers, browser version, platforms etc in parallel. In selenium grid there is hub that is a central server managing all the distributed machines known as nodes.
80. What are some advantages of selenium grid?
The advantages of selenium grid are-
It allows running test cases in parallel thereby saving test execution time.
Multi browser testing is possible using selenium grid by running the test on machines having different browsers.
It is allows multi-platform testing by configuring nodes having different operating systems.
81. What is a hub in selenium grid?
A hub is server or a central point in selenium grid that controls the test executions on the different machines.
82. What is a node in selenium grid?
Nodes are the machines which are attached to the selenium grid hub and have selenium instances running the test scripts. Unlike hub there can be multiple nodes in selenium grid.
83. Explain the line of code Webdriver driver = new FirefoxDriver();.
In the line of code Webdriver driver = new FirefoxDriver(); 'WebDriver' is an interface and we are creating an object of type WebDriver instantiating an object of FirefoxDriver class.
84 What is the purpose of creating a reference variable- 'driver' of type WebDriver instead of directly creating a FireFoxDriver object or any other driver's reference in the statement Webdriver driver = new FirefoxDriver();?
By creating a reference variable of type WebDriver we can use the same variable to work with multiple browsers like ChromeDriver, IEDriver etc.
85. What is testNG?
TestNG(NG for Next Generation) is a testing framework that can be integrated with selenium or any other automation tool to provide multiple capabilities like assertions, reporting, parallel test execution etc.
86. What are some advantages of testNG?
Following are the advantages of testNG-
TestNG provides different assertions that helps in checking the expected and actual results.
It provides parallel execution of test methods.
We can define dependency of one test method over other in TestNG.
We can assign priority to test methods in selenium.
It allows grouping of test methods into test groups.
It allows data driven testing using @DataProvider annotation.
It has inherent support for reporting.
It has support for parameterizing test cases using @Parameters annotation.
87. What is the use of testng.xml file?
testng.xml file is used for configuring the whole test suite. In testng.xml file we can create test suite, create test groups, mark tests for parallel execution, add listeners and pass parameters to test scripts. Later this testng.xml file can be used for triggering the test suite.
88. How can we pass parameter to test script using testNG?
Using @Parameter annotaion and 'parameter' tag in testng.xml we can pass paramters to test scripts.
Sample testng.xml –

```
<suite name="sampleTestSuite">
  <test name="sampleTest">
    <parameter name="sampleParamName" value="sampleParamValue"/>
    <classes>
      <class name="TestFile" />
    </classes>
  </test>
</suite>
```

Sample test script-

```
public class TestFile {
  @Test
  @Parameters("sampleParamName")
  public void parameterTest(String paramValue) {
    System.out.println("Value of sampleParamName is - " + sampleParamName);
  }
```

89. How can we create data driven framework using testNG?

Using @DataProvider we can create a data driven framework in which data is passed to the associated test method and multiple iteration of the test runs for the different test data values passed from the @DataProvider method. The method annotated with @DataProvider annotation return a 2D array of object.

```
//Data provider returning 2D array of 3*2 matrix
 @DataProvider(name = "dataProvider1")
  public Object[][] dataProviderMethod1() {
    return new Object[][] {{"kuldeep","rana"}, {"k1","r1"},{"k2","r2"}};
  }

  //This method is bound to the above data provider returning 2D array of 3*2 matrix
  //The test case will run 3 times with different set of values
  @Test(dataProvider = "dataProvider1")
  public void sampleTest(String s1, String s2) {
    System.out.println(s1 + " " + s2);
  }
```

90. What is the use of @Listener annotation in TestNG?

TestNG provides us different kind of listeners using which we can perform some action in case an event has triggered. Usually testNG listeners are used for configuring reports and logging. One of the most widely used lisetner in testNG is ITestListener interface. It has methods like onTestSuccess, onTestFailure, onTestSkipped etc. We need to implement this interface creating a listner class of our own. After that using the @Listener annotation we can use specify that for a particular test class our customized listener class should be used.

```
@Listeners(PackageName.CustomizedListenerClassName.class)
public class TestClass {
   WebDriver driver= new FirefoxDriver();@Test
   public void testMethod(){
   //test logic
   }
}
```

91. What is the use of @Factory annotation in TestNG?

@Factory annotation helps in dynamic execution of test cases. Using @Factory annotation we can pass parameters to the whole test class at run time. The parameters passed can be used by one or more test methods of that class.

Example – there are two classes TestClass and the TestFactory class. Because of the @Factory annotation the test methods in class TestClass will run twice with the data “k1” and “k2”

```
public class TestClass{
   private String str;

   //Constructor
   public TestClass(String str) {
      this.str = str;
   }

   @Test
```

```
  public void TestMethod() {
    System.out.println(str);
  }
}

public class TestFactory{
  //The test methods in class TestClass will run twice with data "k1" and "k2"
  @Factory
  public Object[] factoryMethod() {
    return new Object[] { new TestClass("K1"), new TestClass("k2") };
  }
}
```

92. What is difference between @Factory and @DataProvider annotation?
@Factory method creates instances of test class and run all the test methods in that class with different set of data.
Whereas, @DataProvider is bound to individual test methods and run the specific methods multiple times.
93. How can we make one test method dependent on other using TestNG?
Using dependsOnMethods parameter inside @Test annotation in testNG we can make one test method run only after successful execution of dependent test method.
@Test(dependsOnMethods = { “preTests” })
94. How can we set priority of test cases in TestNG?
Using priority parameter in @Test annotation in TestNG we can define priority of test cases. The default priority of test when not specified is integer value 0. Example-
@Test(priority=1)
95. What are commonly used TestNG annotations?
The commonly used TestNG annotations are-
@Test- @Test annotation marks a method as Test method.
@BeforeSuite- The annotated method will run only once before all tests in this suite have run.
@AfterSuite-The annotated method will run only once after all tests in this suite have run.
@BeforeClass-The annotated method will run only once before the first test method in the current class is invoked.
@AfterClass-The annotated method will run only once after all the test methods in the current class have been run.
@BeforeTest-The annotated method will run before any test method belonging to the classes inside the tag is run.
@AfterTest-The annotated method will run after all the test methods belonging to the classes inside the tag have run.
96. What are some common assertions provided by testNG?
Some of the common assertions provided by testNG are-
assertEquals(String actual, String expected, String message) – (and other overloaded data type in parameters)
assertNotEquals(double data1, double data2, String message) – (and other overloaded data type in parameters)
assertFalse(boolean condition, String message)
assertTrue(boolean condition, String message)
assertNotNull(Object object)
fail(boolean condition, String message)
true(String message)
97. How can we run test cases in parallel using TestNG?
In order to run the tests in parallel just add these two key value pairs in suite-
parallel=”{methods/tests/classes}”
thread-count=”{number of thread you want to run simultaneously}”.

Check Running Selenium Tests in parallel for details.
98. Name an API used for reading and writing data to excel files.
Apache POI API and JXL(Java Excel API) can be used for reading, writing and updating excel files.
99. Name an API used for logging in Java.
Log4j is an open source API widely used for logging in Java. It supports multiple levels of logging like – ALL, DEBUG, INFO, WARN, ERROR, TRACE and FATAL.
100. What is the use of logging in automation?
Logging helps in debugging the tests when required and also provides a storage of test’s runtime behaviour.