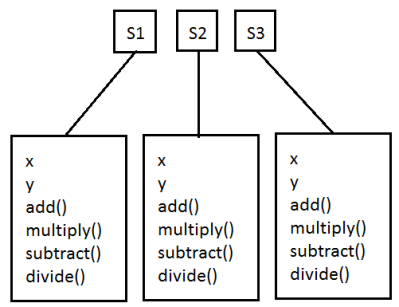
**Java Interview Questions**

**1. What is a class?**

Classes in java are simply bindings which bind/encapsulated data and operations (variables and methods)

* A class is declared by use of the **class** keyword. (please refer above example)
* The data, or variables, defined within a class are called instance variables.
* A typical class may contain methods, data, or both.
* the methods and variables defined within a class are called members of the class.
* We can access class members by creating an object of the class.
* In example-2, ‘s’ is the object of class Sample.
* To access class variables and methods use dot ( . ) operator just after the object name.
* Every object of a class have their own copy of methods and data
* In above example S1, S2, S3 are objects of class *Sample.*
* **Allowed modifiers:** *public, final and abstract* only.
* Class doesn’t have return type.

**2. What is the difference between heap and stack?**

Difference between Java Heap Space and StackMemory. ... Heap memory is used by all the parts of the application whereas stack memory is used only by one thread of execution. Whenever an object is created, it's always stored in the Heap space and stack memory contains the reference to it.

**3. What is the difference between instance variable and local variable?**

There are three types of variables available in Java.

* Instance Variable
* Local Variable
* Static Variable

1. *Instance Variable:*

Instance variable is defined inside class but outside method. All methods inside class can access this variable. Instance variable is not a static variable.

2. *Local Variable:*

Local Variable is defined inside individual method. Its scope is limited to method in which it is declared and defined.

3. *Static Variable:*

A Variable which is declared as ‘Static’ is known as static. It can be a local variable. You do not require to make an object to access static variable. Static variable is accessible using class-name only.  Static variables are loaded in memory at the time of class loading.

**4. What is a Constructor? Types of Constructor?**

It is a special method to initialize the object. Constructor name must be same as Class name.

Constructor must have no explicit return value.

1) Default constructor: No parameters

2) Parameterized Constructor: It contains arguments and parameters.

3) No Argument Constructor: No parameters provided programmer will create that constructor

**5. Difference between Break and Continue?**

Break leaves the loop completely and executes the statements after the loop. Whereas Continue leaves the current iteration and executes with the next value in the loop. ... A break statement results in the termination of the statement to which it applies ( switch , for , do , or while ).

**6. To exit the system from the current execution what command is used in java?**

**7. Addition features in Java 8?**

**8. Difference between for and for each loops in java and use of it?**

**9. Can we have multiple public classes within a class?**

You can have multiple classes within a class. They are called Inner Classes or nested classes. You can even have multiple class definitions in a single .java file without one being nested in another (provided that only one is public, because a public class has to be declared in a file named after it).

**10. What is inheritance? Types of inheritance? Does multiple inheritance allowed in java. If not, why?**

Inheritance is a mechanism where in a new class is derived from an existing class. In Java, classes may inherit or acquire the properties and methods of other classes. A class derived from another class is called a subclass, whereas the class from which a subclass is derived is called a superclass.

1) Single level inheritance 2) Multilevel inheritance 3) Multiple inheritance 4) Hierarchical inheritance 5) Hybrid Inheritance

To avoid ambiguity error java does not support multiple inheritance through class. But through the interface, multiple inheritance is possible in java. No java doesn't support multiple inheritance directly because it leads to overiding of methods when both extended class have a same method name.

**11. What is polymorphism? How we can achieve it?**

Static polymorphism or compile time polymorphism is obtained by using method overloading. When you overload a method, the compiler at the time of compilation knows which method to invoke. ... But, in case of overriding, which method to call is decided at the runtime.

**12. Difference between method overloading and method overriding?**

|  |  |  |
| --- | --- | --- |
| **No.** | **Method Overloading** | **Method Overriding** |
| 1) | Method overloading is used *to increase the readability* of the program. | Method overriding is used *to provide the specific implementation* of the method that is already provided by its super class. |
| 2) | Method overloading is performed *within class*. | Method overriding occurs *in two classes* that have IS-A (inheritance) relationship. |
| 3) | In case of method overloading, *parameter must be different*. | In case of method overriding, *parameter must be same*. |
| 4) | Method overloading is the example of *compile time polymorphism*. | Method overriding is the example of *run time polymorphism*. |
| 5) | In java, method overloading can't be performed by changing return type of the method only. *Return type can be same or different* in method overloading. But you must have to change the parameter. | *Return type must be same or covariant* in method overriding. |

**13. Can we achieve method overloading when two methods have only difference in return type.**

Different ways of Method Overloading in Java. ... The compiler does not consider the return type while differentiating the overloaded method. But you cannot declare two methods with the same signature and different return type. It will throw a compile time error.

**14. Method overloading and overriding examples in Selenium project?**

**15. What is encapsulation?**

* Encapsulation is binding a data and methods into a single entity called ‘[Class](https://testingshastra.wordpress.com/2017/09/28/introduction-to-class/)‘
* In Java, everything is bounded inside a class. You cannot write a single variable or method outside the class
* We can achieve security using Encapsulation.
* Encapsulation gives you more control on data and methods of the class.
* We can restrict access of data and methods using Encapsulation.
* All object oriented languages like, C++, Java, .net etc follows the concept of class.
* Encapsulation allows you to achieve modularity. We never use to write the entire code in a same class. It is a good practice to write different class for different functionalities and then call the required functionalities from the respective class.
* [Class](https://testingshastra.wordpress.com/2017/09/28/introduction-to-class/) has the ownership of each and everything (data, [methods](https://testingshastra.wordpress.com/2017/12/12/methods-classes/), constructor etc) written inside it.

**16. What is IS-A and HAS-A relation in java With examples?**

IS-A = Inheritance

HAS-A = Aggregation and Composition (New Keyword is used)

**17. What is final and super keyword? Difference between them?**

final keyword is used in different contexts. First of all, final is a non-access modifier applicable only to a variable, a method or a class. Following are different contexts where final is used. When a variable is declared with final keyword, its value can't be modified, essentially, a constant.

super is a keyword. It is used inside a sub-class method definition to call a method defined in the super class. Private methods of the super-class cannot be called. Only public and protected methods can be called by the super keyword. It is also used by class constructors to invoke constructors of its parent class.

**18. Explain runtime polymorphism and compile time with examples?**

Overloading is **compile time polymorphism** where more than one methods share the same name with different parameters or signature and different return type. Overriding is **run time polymorphism** having same method with same parameters or signature, but associated in a class & its subclass.

**19. Can final/Static method be overloaded?**

Yes we can overload.

**20. Can final/Static methods be overridden?**

We cannot **overridden**.

**21. Can we overload main method?**

Yes, we overload main method but at the time of execution JVM will always call JVM main Method

**22. Can we execute a class without a main method?**

Yes You can compile and execute without main method By using static block. But after static block executed (printed) you will get an error saying no main methodfound. ... But this will not execute with JAVA 7 version.

**23. What is Package?**

A package in Java is used to group related classes. Think of it as a folder in a file directory. We use packages to avoid name conflicts, and to write a better maintainable code. Packages are divided into two categories:

* Built-in Packages (packages from the Java API)
* User-defined Packages (create your own packages)

**24. What is an Abstract Class? Write an example code?**

**25. What is an Interface? What is the difference from Abstract class?**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Abstract Class** | **Interface** |
| **1** | Abstract class is achieve 0 to 100% abstraction | It is used to achieve 100% abstraction |
| **2** | All methods should be public or protected | All methods should be public or protected |
| **3** | We cannot make abstract method as final but we can make concrete method as final | We cannot make abstract method as final |
| **4** | We have to write abstract keyword for abstract method | By Defaults all methods are abstract |
| **5** | We cannot create instance of abstract class | We cannot create instance of interface |
| **6** | It is not compulsory to initialize abstract class variables | We should compulsory initialize variables in interface |
| **7** | Inside abstract class we can take constructor | Inside interface we cannot define constructor |

**26. Can we use private and protect access modified inside a Interface?**

It doesn't help that **you** declare these fields public , or even public static . The Java**access** modifiers **private and protected** cannot be assigned to a class. Only to constructors, methods and fields **inside** classes. Classes **can** only have the default (package) and public **access modifier** assigned to them.

**27. Can multiple inheritances support in Interface?**

Yes, class name Implements A,B,C

**28. Examples of Abstract and Interface used in selenium project?**

Abstract class is not used in my project but interface (iTestListner) is used in my project

**29. What is Exception and what is the base class of it?**

- Exception is runtime phenomena.

- Exceptions are objects thrown to the JVM

- Throwable is a parent class of Exception

**30. What is Final, Finally, Finalize?**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **final** | **finally** | **finalize** |
| 1) | Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed. | Finally is used to place important code, it will be executed whether exception is handled or not. | Finalize is used to perform clean up processing just before object is garbage collected. |
| 2) | Final is a keyword. | Finally is a block. | Finalize is a method. |

**31. What is done in finally block?**

Finally is used to place important code, it will be executed whether exception is handled or not.

**32. What is garbage collection java? How it is done?**

It makes java memory efficient because garbage collector removes the unreferenced objects from heap memory. It is automatically done by the garbage collector (a part of JVM) so we don't need to make extra efforts.

**33. What is the difference between Throws and Throw?**

|  |  |  |
| --- | --- | --- |
| **No.** | **throw** | **throws** |
| 1) | Java throw keyword is used to explicitly throw an exception. | Java throws keyword is used to declare an exception. |
| 2) | Checked exception cannot be propagated using throw only. | Checked exception can be propagated with throws. |
| 3) | Throw is followed by an instance. | Throws is followed by class. |
| 4) | Throw is used within the method. | Throws is used with the method signature. |
| 5) | You cannot throw multiple exceptions. | You can declare multiple exceptions e.g. public void method()throws IOException, SQLException. |
| 6) | Throw is used to throw the exception Explicitly | Throws is used to propagate the exception |

**34. Gives some examples of java and Selenium?**

**35. What is Java Reflection, Singleton?**

**36. What is threading? How does multi threading achieved? How to initiate a thread in java? What do you mean by thread safe?**

**37. What is the difference between collection and collections?**

* Collection is a root level interface of the Java Collection Framework. Most of the classes in Java Collection Framework inherit from this interface. List, Set and Queue are main sub interfaces of this interface.
* Collections are an utility class in java.util package. It consists of only static methods which are used to operate on objects of type Collection. For example, it has the method to find the maximum element in a collection, it has the method to sort the collection, it has the method to search for a particular element in a collection.

**38. Collection are what type?**

* It is not fixed in size
* It is variable in size
* Collection is a heterogeneous-multiple types of objects in same collection.
* Predefined support in collection for methods.
* Types

List

Set

Queue

**39. Difference between Array and ArrayList?**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Array** | **ArrayList** |
| **1** | **It is fixed in size** | **It is variable in size** |
| **2** | **It Should not grow at runtime** | **It should grow at runtime** |
| **3** | **Array is Homogenous- same type of array like int type** | **Duplicates are allowed** |
| **4** | **Predefined method support in array** | **Predefined method support in array** |
| **5** | **Insertion order is not preserved** | **Insertion order is preserved** |
| **6** | **It is not variable in size** | **It is variable in size** |
| **7** | **Backend data structure is array** | **Backend data structure is array** |
| **8** | **Random access is possible** | **Random access is possible** |
| **9** | **Null insertion is possible** | **Null insertion is possible** |

**40. Difference between Set and HashSet?**

Set is an interface, HashSet - implementation of interface. It is recommended to use interface instead of implementation when you declaring variables.

If go further into details, interface in Java it is a set of methods, and if some class wants to implement this interface, it must implement all of this methods.

Set interface represents a set of some objects, non-ordered, without random-element access. HashSet - implementation of Set interface, based on .hashCode() function

**41. Difference between HashMap and HasTable?**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **HashTable** | **HashMap** |
| **1** | HashTable is synchronized | HashMap is not synchronized |
| **2** | HashTable is thread safe. | HashMap is not thread safe |
| **3** | HashMap is slower in execution. | HashMap is faster in execution. |
| **4** | HashTable does not allow null keys or value | HashMap allows one null key and any number of null values. |

**42. Difference between ArrayList and LinkList?**

|  |  |
| --- | --- |
| **ArrayList** | **LinkedList** |
| 1) ArrayList internally uses a **dynamic array** to store the elements. | LinkedList internally uses a **doubly linked list** to store the elements. |
| 2) ArrayList is synchronized. | LinkedList is A synchronized. |
| 2) Manipulation with ArrayList is **slow** because it internally uses an array. If any element is removed from the array, all the bits are shifted in memory. | Manipulation with LinkedList is **faster** than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory. |
| 3) An ArrayList class can **act as a list** only because it implements List only. | LinkedList class can **act as a list and queue** both because it implements List and Deque interfaces. |
| 4) ArrayList is **better for storing and accessing** data. | LinkedList is **better for manipulating** data. |

**43. How do you use Map collection your project?**

**- I didn’t use**

**44. Can we have duplicate key value in HasMap?**

**-** Yes but only values is duplicate not key is duplicate

**45. How to fetch values from a hashmap?**

Use the GET method of HashMap to get what you want. Use the 'string' key of thehashmap, to access its value which is your tab class. You have to follow the following sequence of opeartions: Convert Map to MapSet with map.entrySet();

**46. Difference between String, String Builder And String Buffer?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **String** | **String Buffer** | **String Builder** |
| **1** | String is immutable in nature | StringBuffer is mutable in nature | StringBuilder is mutable in nature |
| **2** | String is secure than the String buffer and String Builder | Objects of string buffer class are synchronized | Objects of string builder class are not synchronized |
| **3** |  | In string buffer it will focus on single thread at a time | In string builder it will focus on multiple thread at a time |
| **4** |  | String Buffer is thread safe. | String builder is not thread safe |
| **5** |  | String Buffer is slower in execution. | String builder is faster in execution. |
| **6** |  | String Buffer does not allow null keys or value | String builder allows one null key and any number of null values. |

**47. How to iterate Hash Map?**

There is no iterator method in HashMap class

* Get all the keys from map into set
* Save keys in set instance
* Apply iterator on set of keys
* Read one by one

**47. Differentiate between Array list and vector?**

|  |  |
| --- | --- |
| **ArrayList** | **Vector** |
| Array list is not synchronized | Vector is synchronized |
| It will focus on multiple thread at a time | It will focus on single thread at a time |
| ArrayList is not thread safe. | Vector is thread safe |
| ArrayList is faster in execution. | Vector is slower in execution. |
| Array list is not belongs to legacy class because it is introduced in 1.2 version of java | Vector is belongs to legacy class because it is introduced in 1.0 Version of java |

**#Java Programs**

1. Write a Code to generate Random numbers.

2. Write code to verify a number is perfect number or not.

3. Fibonacci series from 1 to 10.

4. Write a program to find a Factorial of a number.

5. Swap two numbers without using the third variable.

6. Program to find greatest of three numbers.

7. An Array of numbers given. .Find the largest two number and print it

8. Reverse a number.

9. Verify if a given number is a palindrome or not.(same concept of reversing a number)

10. Armstrong number program.

11. A String is given remove the white spaces, reverse it and print the only odd position characters.

12. Check if a string is an anagram of another string.

13. A string is given make few characters to upper case as asked.

14. In a String print the occurrence of each character.

15. In a Statement check how any duplicate strings are there and remove them.

16. Use split to print each word of a statement.

17. Find the substring of a String.

18. Remove duplicate characters from a String and Print it.

19. Write a code to print the triangle of numbers.

20. Read and Write file program.

21. Calculate power of a number using a while loop

**#SQL**

1. What is primary key and unique key?

2. How many joins are there any difference?

3. Difference between join and union?

4. Select top 3 max salary employees by depts.

5. Difference between where and having and use of it?

6. Different types of Date functions?

7. What are the grouping functions?

8. Which function is used to get the current date?

9. What is a sub query?

10. What is indexing?

**#Selenium Interview Questions**

**1. What are the components of Selenium?**

Selenium has 4 special components that is:

* Selenium IDE (Integrated Development Environment)
* Selenium GRID
* Selenium(RC) 1.0 (Also called as Remote Control)
* Selenium 2.0/3.0 (Also called as Webdriver)

**2. Which version you are using and what addition features you see from the previous version?**

**3. What is a WebDriver?**

WebDriver is a web automation framework that allows you to execute your tests against different browsers, not just Firefox, Chrome (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.

**4. How do you select you drivers to launch a url?**

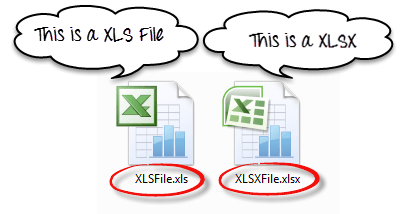
**5. Which method is used to fetch the driver?**

**6. Which framework you are using? Explain in details and its benefit? What is POM?**

**7. What is data driven and keyword driven frameworks?**

Data Driven Test Framework

In data driven framework all of our test data is generated from some external files like[Excel](https://www.guru99.com/excel-tutorials.html), CSV, XML or some database table.

[](https://www.guru99.com/images/AdvanceSelenium/071514_0711_AllAboutExc1.png)

To read or write an Excel,Apache provides a very famous library POI. This library is capable enough to read and write both XLS and XLSX file format of Excel.

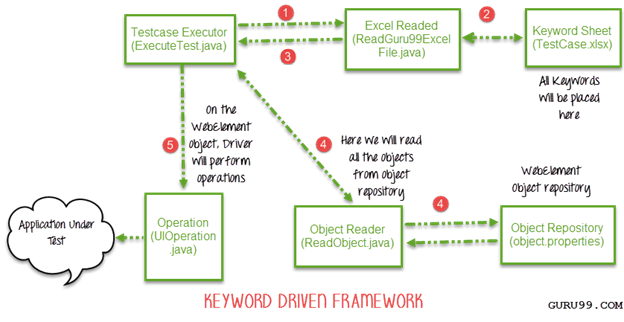
To read XLS files, an HSSF implementation is provided by POI library.

To read XLSX, XSSF implementation of POI library will be the choice. Let's study these implementations in detail.

We already learned about Data Driven Testing in our [previous tutorial](https://www.guru99.com/all-about-excel-in-selenium-poi-jxl.html)

Keyword Driven Test Framework:

In keyword driven test framework, all the operations and instructions are written in some external file like Excel worksheet. Here is how the complete framework looks like

[](https://www.guru99.com/images/AdvanceSelenium/071514_0715_CreatingKey2.png)

**8. What locators you are using?**

* **Xpath**
* **Id**
* **cssSelector**
* **name**
* **linkText**
* **PartialLinkText**
* **tagName**
* **className**

**9. How do you explain the concept of object repository?**

Object Repository is a collection of webElements.

Object Repository is a collection of object and properties with which QTP will be able to recognize the objects and act on it. When a user records a test, the object sand its properties are captured by default. Without understanding objects and its properties, QTP will NOT be able to play back the scripts.

**10. What is the difference between findElement and findElements and what is return type? Give some example where it is used?**

|  |  |
| --- | --- |
| **Find Element** | **Find Elements** |
| Returns the first most web element if there are multiple web elements found with the same locator | Returns a list of web elements |
| Throws exception NoSuchElementException if there are no elements matching the locator strategy | Returns an empty list if there are no web elements matching the locator strategy |
| It will only find one web element | It will find a collection of elements whose match the locator strategy. |
| Not Applicable | Each Web element is indexed with a number starting from 0 just like an array |

### Example: How to use Find Element command

The following application is used for demo purpose

<http://demo.guru99.com/test/ajax.html>

**Scenario:**

1. Open the AUT

2. Find and click radio button

package com.sample.stepdefinitions;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class NameDemo {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver", "D:\\3rdparty\\chrome\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

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

// Find the radio button for “No” using its ID and click on it

System.out.println (By.Name("name"));

}

}

### Example: How to use Find Elements command

**Scenario:**

1. Open the URL for Application Under Test

2. Find the text of radio buttons and print it onto the output console

package com.sample.stepdefinitions;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class NameDemo {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "X://chromedriver.exe");

WebDriver driver = new ChromeDriver();

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

List<WebElement> elements = driver.findElements(By.name("name"));

System.out.println("Number of elements:" +elements.size());

for (int i=0; i<elements.size();i++){

System.out.println("Radio button text:" + elements.get(i).getAttribute("value"));

}

}

}

**Summary:**

* Find Element command returns the web element that matches the first most element within the web page.
* Find Elements command returns a list of web elements that match the criteria.
* Find Element command throws NoSuchElement exception if it does not find the element matching the criteria.
* Find Elements command returns an empty list if there are no elements matching the criteria

**11. Which method is used to get the text of an element?**

**-** getText()

**12. How do you fetch an attribute value of an element?**

**-** by using getAttribute() method of webDriver

**13. When multiple checkboxes are there how do you check which one is checked previously?**

**-** first we have fetch all element and apply isSelected() method

**14. What is the return type of isSelected() method?**

**-Boolean**

**15. What are the methods used to verify the existence of an object in a webpage?**

**-** isVisible, isDisplayed

**16. What is a Xpath difference between relative and absolute xpath?**

**Absolute Xpath**: It uses Complete path from the Root Element to the desire element.

**Relative Xpath**: You can simply start by referencing the element you want and go from there.

**17. Read all axes scenarios of xpath like parent, following, ancestor, child and sibling.**

**18. How to fetch an element when its attributes are changing frequently?**

-by using functions of xpath i.e Contains, starts-with, text, last, position

**19. What are the ways to click on a button?**

**-** webElement.click() method, submit() method.

**20. What are different types of wait (write the syntax) and explain the scenarios where you will use them?**

**Selenium Web Driver Waits**

1. Implicit Wait
2. Explicit Wait
3. Fluent Wait

**Implicit Wait**

It is type of wait which is attach to the driver instance throughout its lifetime. It will wait for all the elements present on webpage. It is not customizable wait.

Selenium Web Driver has borrowed the idea of implicit waits from Watir.

driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);

## Explicit Wait

## It is type of wait in which it will wait for particular web element on web page. It is highly customizable wait with the help of that we can customize polling time. We can ignore exceptions we can also apply timeout.

## There are two types of explicit waits

## 1. WebDriver wait

## Web Driver wait is child class of fluent wait. This doesn’t have its own methods. It uses the method of fluent wait. With the help of this wait we can customize wait with the help of that we can customize polling time. It is more specific wait than fluent wait. Web Driver wait has its constructor to initialize operations

## 2. Fluent wait

## It is a parent class of WebDriver wait. It has its own methods. It is more generalized wait.

## poolingEvery

## Ignoring

## Until

## waitTimeout

WebDriverWait wait = new WebDriverWait(WebDriverRefrence,TimeOut);

**Fluent Wait**

Wait wait = new FluentWait(WebDriver reference)

.withTimeout(timeout, SECONDS)

.pollingEvery(timeout, SECONDS)

.ignoring(Exception.class);

**21. What are different types of exception in selenium?**

IllegalStateException

WebDriverExeptions

StaleElementException

Timeout exception

NoSuchElementException

IllegalArgumentExceptio

NoAlertPresentExceptiom

NowindowPresentException

**22. How do you handle exceptions selenium?**

**23. There are 4 windows in browsers open you have no idea where the required element is present . What will be your approach to find that element?**

**-** windowHandles()

**24. What is the difference between getWindowhandle() and getWindowHandles()? What is its return type?**

**GetWindowHandle Command :-**To get the window handle of the current window.

 String handle= driver.getWindowHandle();

//Return a string of alphanumeric window handle

**GetWindowHandles Command :-** To get the window handle of all the current windows.

Set<String> handle= driver.getWindowHandles();

//Return a set of window handle

**25. Difference between Quit() and Close()?**

- Close() method is used to close the particular window which is open.

- Quit() method is used to Close all the window which are open.

**26. How do you handle an alert pop-up?**

**Handling web-based pop-up box**

WebDriver offers the users with a very efficient way to handle these pop ups using Alert interface.

**There are the four methods that we would be using along with the Alert interface.**

**1) void dismiss()** – The dismiss() method clicks on the “Cancel” button as soon as the pop up window appears.  
**2) void accept()** – The accept() method clicks on the “Ok” button as soon as the pop up window appears.  
**3) String getText()** – The getText() method returns the text displayed on the alert box.  
**4) void sendKeys(String stringToSend)** – The sendKeys() method enters the specified string pattern into the alert box.

Object Creation for Alert class  
Alert alert = driver.switchTo().alert();

We create a reference variable for Alert class and references it to the alert.

**Switch to Alert**  
Driver.switchTo().alert();  
The above command is used to switch the control to the recently generated pop up window.

**Accept the Alert**  
alert.accept();  
The above command accepts the alert thereby clicking on the Ok button.

**Reject the Alert**  
alert.dismiss();  
The above command closes the alert thereby clicking on the Cancel button and hence the operation should not proceed

**27. How do you get the text inside a Alert?**

**- String getText()** – The getText() method returns the text displayed on the alert box.

**28. What is a Alert? Interface/class?**

Alert interface provides the below few methods which are widely used in Selenium Webdriver. 1) void dismiss() // To click on the 'Cancel' button of thealert. ... 2) void accept() // To click on the 'OK' button of the alert. driver.switchTo().alert().accept(); 3) String getText() // To capture the alert message.

**29. How do you handle Frames in selenium?**

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");

**30. Give one example of method overloading concept used in Selenium? Ans : Switching to frames**

**31. How do you select a dropdown value and what are the different methods are there?**

|  |  |
| --- | --- |
| **Method** | **Description** |
| **selectByVisibleText()**and**deselectByVisibleText()** *Example:*[How to Select Option from DropDown using Selenium Webdriver](https://www.guru99.com/images/image017(3).png) | * Selects/deselects the option that displays the text matching the parameter. * **Parameter**: The exactly displayed text of a particular option |
| **selectByValue()** and**deselectByValue()** *Example:*[How to Select Option from DropDown using Selenium Webdriver](https://www.guru99.com/images/image018(3).png) | * Selects/deselects the option whose "value" attribute matches the specified parameter. * **Parameter**: value of the "value" attribute * Remember that not all drop-down options have the same text and "value", like in the example below.   [How to Select Option from DropDown using Selenium Webdriver](https://www.guru99.com/images/image019(2).png) |
| **selectByIndex()** and**deselectByIndex()** *Example:[How to Select Option from DropDown using Selenium Webdriver](https://www.guru99.com/images/image020(2).png)* | * Selects/deselects the option at the given index. * **Parameter**: the index of the option to be selected. |
| **isMultiple()** *Example:[How to Select Option from DropDown using Selenium Webdriver](https://www.guru99.com/images/image021(2).png)* | * Returns TRUE if the drop-down element allows multiple selections at a time; FALSE if otherwise. * **No parameters needed** |
| **deselectAll()** *Example:[How to Select Option from DropDown using Selenium Webdriver](https://www.guru99.com/images/image022(2).png)* | * Clears all selected entries. This is only valid when the drop-down element supports multiple selections. * **No parameters needed** |

**32. When your Xpath is returning more than one matching values how will you handle it?**

**33. Which is better Xpath/CSS? Why?**

CSS selectors perform far better than Xpath and it is well documented in Selenium community. Here are some reasons,

* Xpath engines are different in each browser, hence make them inconsistent
* IE does not have a native xpath engine, therefore selenium injects its own xpath engine for compatibility of its API. Hence we lose the advantage of using native browser features that WebDriver inherently promotes.
* Xpath tend to become complex and hence make hard to read in my opinion

**34. How does u capture the screenshot? What is the best place to have the screenshot code?**

**I** capture screenshot by using three ways 1. takeScreenshot(selenium Method) 2. aShot(API) 3. Robot(Java Class).

Best Place to insert screenshot code is TestNGListner.

**35. How do you connect Excel sheet? Write the code?**

**36. Write the database connection code.**

**37. How do you read and write a text file?**

**38. How do you read and write a pdf file?**

**39. What are the disadvantages of selenium?**

**40. Give some scenarios where you cannot automate?**

* **CAPTCHA scenarios**: Well, CAPTCHAS are there for purpose. To bypass automation. Best way to handle it is to tell your dev team to disable it or make it static.
* **Video streaming scenarios**: More often that not, Selenium won’t be able to recognise video controls. JavaScript Executor and flex-ui-selenium will work to some extent, but they are not entirely reliable.
* **Code reading scenarios**: If your web app has a functionality which reads barcodes or QR codes, it’s not beneficial to automate it. There may be some tools available for them but I’m not sure how effective they are.
* **Crash recovery scenarios**: You might want to test your application’s crash recovery. This is a scenario best tested manually. I am not saying you won’t be able to test it using Selenium. You may be. But I don’t know how feasible and beneficial it would be.
* **Performance testing**: It can be automated but it’s best not to automate performance testing using Selenium.

**41. What is the base class of exception?**

**-**Throwable

**42. When you execution is failed how do you debug your code?**

**- Using Breaking Points**

**43. How do you fetch the tile of a webpage?**

**-**getTitle() method of webdriver

**44. What are the methods used to verify the end result is achieved?**

**-** assert() method used to verify the end result is achieved

**45. How do you handle dynamic webtable? In cases you have no idea of how many rows of record will come , what should be the approach to handle this case?**

**-** fetch all table data

**46. How to verify the broken images of a page?**

**47. How to clear the cookies of the browser before start the execution?**

**-** deleteAllCookies() method of webdriver

**48. Write the code to maximize the window?**

**-** *driver*.manage().window().maximize();

**49. Implementation of collection in your framework?**

**-** at the time of fetching the data from excel file that data will be saved in ArrayList() method of collection.

**50. Give a scenario where inheritance is used in your framework?**

**-** We have implemented inheritance features in POM\_Base approach.

We have a POM\_Base class and in that class openBrowser(), openUrl(), maxamizeWindow() methods.

**51. Give a scenario where interface is used in your framework?**

**-** iTestListner(is a interface)

**52. Write a code using java script executor to scroll the webpage?**

**- public** **class** JavaScriptExecutorDemo {

**public** **static** **void** main(String[] args) **throws** AWTException {

WebDriverManager.*chromedriver*().setup();

WebDriver driver = **new** ChromeDriver();

driver.get("https://goo.gl/1bvRJu");

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

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeAsyncScript("window.scrollBy(0,500)");

}

}

**53. How do you highlight an web element?**

**54. What is use of property file in selenium?**

.properties files are mainly used in Java programs to maintain **project configuration data**, database config or project settings etc. Each parameter in properties file are stored as a pair of strings, in key-value pair format, where each key is on one line

**55. How do you handle multiple browser selection?**

**-**

**56. What is used for reporting?**

**57. What are the annotations used in TestNG? How it is difference from JUnit?**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Description** | **TestNG** | **JUnit 4** |
| 1 | Test annotation | @Test | @Test |
| 2 | Executes before the first test method is invoked in the current class | @BeforeClass | @BeforeClass |
| 3 | Executes after all the test methods in the current class | @AfterClass | @AfterClass |
| 4 | Executes before each test method | @BeforeMethod | @Before |
| 5 | Executes after each test method | @AfterMethod | @After |
| 6 | annotation to ignore a test | @Test(enable=false) | @ignore |
| 7 | annotation for exception | @Test(expectedExceptions = ArithmeticException.class) | @Test(expected = ArithmeticException.class) |
| 8 | timeout | @Test(timeout = 1000) | @Test(timeout = 1000) |
| 9 | Executes before all tests in the suite | @BeforeSuite | n/a |
| 10 | Executes after all tests in the suite | @AfterSuite | n/a |
| 11 | Executes before a test runs | @BeforeTest | n/a |
| 12 | Executes after a test runs | @AfterTest | n/a |
| 13 | Executes before the first test method is invoked that belongs to any of these groups is invoked | @BeforeGroups | n/a |
| 14 | run after the last test method that belongs to any of the groups here | @AfterGroups | n/a |

**58. Explain all annotations and its use?**

**59. How do you priorities your test cases?**

**-** To set Priority attribute of test Annonation @Test (priority=1)

**60. Suppose you want to skip one test method during execution how you will do it?**

**- @Test(**unable=false)

**61. How cross browser testing is handled?**

**- By using grid by using this we can handle different brewers and different os**

**62. What is the hierarchy of testNG.xml tags?**

<!DOCTYPE suite SYSTEM "http://beust.com/testng/testng-1.0.dtd" >

<suite name="My test suite">

<test name="testing">

<classes>

<class name="com.guru99.SuiteTest1" />

<class name="com.guru99.SuiteTest2" />

</classes>

</test>

</suite>

**63. Write the structure of testNG.xml file and explain?**

<!DOCTYPE suite SYSTEM "http://beust.com/testng/testng-1.0.dtd" >

<suite name="My test suite">

<test name="testing">

<classes>

<class name="com.guru99.SuiteTest1" />

<class name="com.guru99.SuiteTest2" />

</classes>

</test>

</suite>

**64. What is the difference between Assert and verify?**

**-** Assert is used to insert validation points inside scripts when assert is used if condition is not true it throws assertion error and terminates the program.

- Verify is used to insert validation points inside scripts. If condition is false its still allows further steps to get execute. Dose not throw assertion error and marks test as a failure.

**65. What are the different methods of Assert?**

|  |  |
| --- | --- |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(boolean,%20boolean))(boolean actual, boolean expected)            Asserts that two booleans are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(boolean,%20boolean,%20java.lang.String))(boolean actual, boolean expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two booleans are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(byte[],%20byte[]))(byte[] actual, byte[] expected)            Asserts that two arrays contain the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(byte[],%20byte[],%20java.lang.String))(byte[] actual, byte[] expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two arrays contain the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(byte,%20byte))(byte actual, byte expected)            Asserts that two bytes are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(byte,%20byte,%20java.lang.String))(byte actual, byte expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two bytes are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(char,%20char))(char actual, char expected)            Asserts that two chars are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(char,%20char,%20java.lang.String))(char actual, char expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two chars are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.util.Collection,%20java.util.Collection))([Collection](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Collection.html?is-external=true)<?> actual, [Collection](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Collection.html?is-external=true)<?> expected)            Asserts that two collections contain the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.util.Collection,%20java.util.Collection,%20java.lang.String))([Collection](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Collection.html?is-external=true)<?> actual, [Collection](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Collection.html?is-external=true)<?> expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two collections contain the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(double,%20double,%20double))(double actual, double expected, double delta)            Asserts that two doubles are equal concerning a delta. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(double,%20double,%20double,%20java.lang.String))(double actual, double expected, double delta, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two doubles are equal concerning a delta. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(float,%20float,%20float))(float actual, float expected, float delta)            Asserts that two floats are equal concerning a delta. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(float,%20float,%20float,%20java.lang.String))(float actual, float expected, float delta, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two floats are equal concerning a delta. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(int,%20int))(int actual, int expected)            Asserts that two ints are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(int,%20int,%20java.lang.String))(int actual, int expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two ints are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.lang.Iterable,%20java.lang.Iterable))([Iterable](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Iterable.html?is-external=true)<?> actual, [Iterable](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Iterable.html?is-external=true)<?> expected)            Asserts that two iterables return iterators with the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.lang.Iterable,%20java.lang.Iterable,%20java.lang.String))([Iterable](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Iterable.html?is-external=true)<?> actual, [Iterable](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Iterable.html?is-external=true)<?> expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two iterables return iterators with the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.util.Iterator,%20java.util.Iterator))([Iterator](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Iterator.html?is-external=true)<?> actual, [Iterator](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Iterator.html?is-external=true)<?> expected)            Asserts that two iterators return the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.util.Iterator,%20java.util.Iterator,%20java.lang.String))([Iterator](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Iterator.html?is-external=true)<?> actual, [Iterator](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Iterator.html?is-external=true)<?> expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two iterators return the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(long,%20long))(long actual, long expected)            Asserts that two longs are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(long,%20long,%20java.lang.String))(long actual, long expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two longs are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.util.Map,%20java.util.Map))([Map](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Map.html?is-external=true)<?,?> actual, [Map](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Map.html?is-external=true)<?,?> expected)            Asserts that two maps are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.lang.Object[],%20java.lang.Object[]))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)[] actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)[] expected)            Asserts that two arrays contain the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.lang.Object[],%20java.lang.Object[],%20java.lang.String))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)[] actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)[] expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two arrays contain the same elements in the same order. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.lang.Object,%20java.lang.Object))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) expected)            Asserts that two objects are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.lang.Object,%20java.lang.Object,%20java.lang.String))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two objects are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.util.Set,%20java.util.Set))([Set](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Set.html?is-external=true)<?> actual, [Set](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Set.html?is-external=true)<?> expected)            Asserts that two sets are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.util.Set,%20java.util.Set,%20java.lang.String))([Set](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Set.html?is-external=true)<?> actual, [Set](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Set.html?is-external=true)<?> expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Assert set equals |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(short,%20short))(short actual, short expected)            Asserts that two shorts are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(short,%20short,%20java.lang.String))(short actual, short expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two shorts are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) actual, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) expected)            Asserts that two Strings are equal. |
| static void | [**assertEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEquals(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) actual, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two Strings are equal. |
| static void | [**assertEqualsNoOrder**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEqualsNoOrder(java.lang.Object[],%20java.lang.Object[]))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)[] actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)[] expected)            Asserts that two arrays contain the same elements in no particular order. |
| static void | [**assertEqualsNoOrder**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertEqualsNoOrder(java.lang.Object[],%20java.lang.Object[],%20java.lang.String))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)[] actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)[] expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two arrays contain the same elements in no particular order. |
| static void | [**assertFalse**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertFalse(boolean))(boolean condition)            Asserts that a condition is false. |
| static void | [**assertFalse**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertFalse(boolean,%20java.lang.String))(boolean condition, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that a condition is false. |
| static void | [**assertNotEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotEquals(double,%20double,%20double))(double actual1, double actual2, double delta) |
| static void | [**assertNotEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotEquals(double,%20double,%20double,%20java.lang.String))(double actual1, double actual2, double delta, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message) |
| static void | [**assertNotEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotEquals(float,%20float,%20float))(float actual1, float actual2, float delta) |
| static void | [**assertNotEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotEquals(float,%20float,%20float,%20java.lang.String))(float actual1, float actual2, float delta, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message) |
| static void | [**assertNotEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotEquals(java.lang.Object,%20java.lang.Object))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual1, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual2) |
| static void | [**assertNotEquals**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotEquals(java.lang.Object,%20java.lang.Object,%20java.lang.String))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual1, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual2, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message) |
| static void | [**assertNotNull**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotNull(java.lang.Object))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) object)            Asserts that an object isn't null. |
| static void | [**assertNotNull**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotNull(java.lang.Object,%20java.lang.String))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) object, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that an object isn't null. |
| static void | [**assertNotSame**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotSame(java.lang.Object,%20java.lang.Object))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) expected)            Asserts that two objects do not refer to the same object. |
| static void | [**assertNotSame**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNotSame(java.lang.Object,%20java.lang.Object,%20java.lang.String))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two objects do not refer to the same objects. |
| static void | [**assertNull**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNull(java.lang.Object))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) object)            Asserts that an object is null. |
| static void | [**assertNull**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertNull(java.lang.Object,%20java.lang.String))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) object, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that an object is null. |
| static void | [**assertSame**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertSame(java.lang.Object,%20java.lang.Object))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) expected)            Asserts that two objects refer to the same object. |
| static void | [**assertSame**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertSame(java.lang.Object,%20java.lang.Object,%20java.lang.String))([Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) actual, [Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) expected, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that two objects refer to the same object. |
| static void | [**assertTrue**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertTrue(boolean))(boolean condition)            Asserts that a condition is true. |
| static void | [**assertTrue**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#assertTrue(boolean,%20java.lang.String))(boolean condition, [String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Asserts that a condition is true. |
| static void | [**fail**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#fail())()            Fails a test with no message. |
| static void | [**fail**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#fail(java.lang.String))([String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message)            Fails a test with the given message. |
| static void | [**fail**](https://static.javadoc.io/org.testng/testng/6.8.17/org/testng/Assert.html#fail(java.lang.String,%20java.lang.Throwable))([String](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html?is-external=true) message, [Throwable](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Throwable.html?is-external=true) realCause)            Fails a test with the given message and wrapping the original exception. |

**66. How do you store your testNG reports?**

**67. How do you use parameters in testNG?**

**-**  By using @paramaterTag

**-** @Parameters ({ "sUsername", "sPassword" })

**68. What is use of Grouping in testNG?**

## - TestNG Groups with Example

We use groups in[Testng](https://www.guru99.com/all-about-testng-and-selenium.html)when,

* We don't want to define test methods separately in different classes (depending upon functionality) and
* At the same time want to ignore (not to execute) some test cases as if they does not exist in the code.
* So to carry out this we have to Group them. This is done by using "include" and "exclude" mechanism supported in testNG.

In below example, we have shown the syntax of how to use groups in the XML file.

@Test (groups = { "bonding", "strong\_ties" })

**69. What is the use of Maven?**

**-** Maven is a build tool.

- Maven is used to define project structure, dependencies, build, and test management. Using pom.xml(Maven) you can **configure** dependencies needed for building **testing**and **running** code. Maven automatically downloads the necessary files from the repository while building the project

**70. How does u define dependencies?**

- In POM.xml file we have dependencies tag in which we can define all the required dependencies. Dependencies can be taken from maven dependencies

**71. What are the advantages of using Maven in your project?**

**-** command line we can execute our program , easily transfer our project to one system to another, jars will be downloaded if dependencies add in pom.xml files

**72. Name the folder of maven which contains all libraries?**

- pom.xml

**73. Give some example of implementation of Maps in your project?**

**-** We have not used maps we have used set and list

**74. What are the use of Action class?**

**-** By using action class we can handle all the mouse and keyboards events it belongs to selenium.

**-** keyevents.pgUp , keyevents.pgSown

**75. How do you perform double click?**

**-** By using action class.

Action a = new Action(driver)

a.doubleClick

a.rightClick

**76. Write code for Drag and Drop?**

Action a = new Action(driver)

a.dragAndDrop(source, destination)

**77. How to hover over an element?**

**-** By using action class we can hover of a particular webElement.

**public** **static** **void** mouseHover(String locatorType, String locatorValue) {

Actions bk = **new** Actions(Constants.*driver*);

bk.moveToElement(*getWebElement*(locatorType, locatorValue)).build().perform();

*log*.info("Mouse movement done successfully");

}// End of mouseHover Method

**78. How do enter values using key commands?**

**79. What is the difference between key command and key types?**

**80. How do you fetch all links from a web page?**

**public** **class** FindNoOfLinkOnPage {

**public** **static** **void** main(String[] args) {

System.*setProperty*("webdriver.chrome.driver", "E:\\Selenium\\chromedriver.exe");

WebDriver driver= **new** ChromeDriver();

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

List<WebElement> list=driver.findElements(By.*tagName*("a"));

System.***out***.println(list.size());

**for** (**int** i = 0; i < list.size(); i++) {

String s=list.get(i).getText();

System.***out***.println(s); }}}

**81. If a page containing 1000 of images how you select the 100th image?**

**82. How do you upload images into webpage?**

**83. What is @DataProvider?**

**-** DataProvider is an annotation when we want to perform data driven testing then dataProvider annotation is used it is method which is provide a data. Test annotation has data provider as an attribute which receives a data from the data provider.

**public** **class** DataProviderDemo {

@Test(dataProvider = "SecondDataProvider")

**private** **void** m1(**int** i) {

System.***out***.println("Hello="+i);

}

@DataProvider(name = "SecondDataProvider")

**public** Integer[] dataProvder1() {

Integer[] i= {10,20,30};

**return** i;

}

**84. How to execute failed test cases?**

**-** In testNG we have TestNG output folder in that failed.xml file is present simply run that file then all failed test cases will be executed.

**85. How to skip Test cases?**

**- @ignore, @Test (unable=false), @Test (exclude), @Test (invocation Time = 0), //comment**

**#Other Questions**

1. Give a short introduction about yourself with your roles and responsibilities in the team?

2. How do you approach to develop Automation test cases?

3. Explain your agile work and your role in it?

4. How do you estimate your automation cases?

5. What are the principles of agile?

6. Why do we follow Fibonacci series in pointing?

7. How do you track your automation cases?

8. How do you execute your cases?

9. Why do you choose selenium over other automation tool?

10. Do you have any experience in framework development?

11. Have you worked in any BDD framework?

12. Tell me the difficulties you faced in developing automation scripts? Any complex scenarios handled?

13. Dou you have experience in API Testing

14. What are the structures used in SOAP and REST services?

15. What is the major difference REST and SOAP?

16. What is the difference methods used in REST?

17. How you pass your data in SOAP request?

18. How do verify the responses?

19. What is WSDL?

20. What is the use of Jenkins?

21. What are the commands of jenkin?

22. How do you set-up your maven project in jenkin?

23. How do you start jenkin?