**\*\*\*\*\*\*\*\*\*Concepts important for the Interview (Selenium with Java):-**

**🡪From Core Java:-(OOP’s concepts, String, Static keyword, Final keyword, Collections, Exception Handling, Methods, Constructors, Basic Programs of Java (based on oops concepts).**

**🡪From Selenium:-Real time scenario of Selenium(Basic & Advance level Scenario),how to find element?, xpath , How to debug automation fixes?, about test script…,exception types with examples, difference between findelement & findelements ,POM Class, POI Concept, selenium Grid Concept, TestNG annotation cases, Parallel cases execution, how to check element display on webpage,(questions about html, css, javascript also)..SQL queries, Joins in SQL..**

**🡪From Manual Testing:-Definition of Manual Testing, types of black box & white box, difference between ((Smoke & Sanity testing),(Retesting & Regression testing),(System & Acceptance testing),RTM, Defect/Bug life cycle, Severity & Priority with example, For preparing test documentation(Test scenario, Test cases, Defect Report),SDLC ,STLC ,Agile Methodology, Roles & Responsibility of Software Tester.**

**\*\*\*The 12 principles behind the Agile Methodologies are:-**

**1.** Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

**2.** Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.

**3.** Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

**4.** Business people and developers must work together daily throughout the project.

**5.** Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

**6.** The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

**7.** Working software is the primary measure of progress.

**8.** Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

**9.** Continuous attention to technical excellence and good design enhances agility.

**10.** Simplicity the art of maximizing the amount of work not done is essential.

**11.** The best architectures, requirements, and designs emerge from self-organizing teams.

**12.** At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

**\*\*\*\*\*IMP things for Telephonic Interview:-**

1. Always start with maintaining your name & don’t forget to sound pleasant & Friendly. But not too friendly, just be nice.

2. Be clear in your voice , pronounce each & everything very clear. Be loud enough that the other person on the phone can hear you.

3. It is not someone you known, so be careful about the kind of language you use. Use professional & respectful words. It will impress others very fast.

4. Listen & understand what is being asked, whether you understood the person or not have patience. Let the other person finish other than interrupting them in between. Ask whatever your queries are after that person is done with their part.

\*\*\*\*\*\*\*\***Version Control Tool in selenium:-**

It is also called “**Revision Control Tool” or “**[**Source Control**](https://www.perforce.com/blog/vcs/what-source-control) **Tool”**. It’s an important component of software configuration management; you can use version control to version code, binary files,and digital assets.

Version Control Systems(VCS) allow multiple developers, designers, and team members to work together on the same project. VCS systems are critical to ensure everyone has access to the latest code. As development gets more complex, there's a bigger need to manage multiple versions of entire products.

Version control allows you to manage changes to files over time and store these modifications in a database.

Version control is important to keep track of changes — and keep every team member working off the latest version. You should use version control software for all code, files, and assets that multiple team members will collaborate on.

**\*Continuous Integration/Continuous Deployment Tools(CI/CD Tools):-Jenkins, Bitbucket, Sonar, GitLab, Bamboo, JetBrains TeamCity, Codefresh , Travis CI, AWS CodeDeploy.**

**\*\*\*\*Testing Techniques:-Junit, Jbehave.**

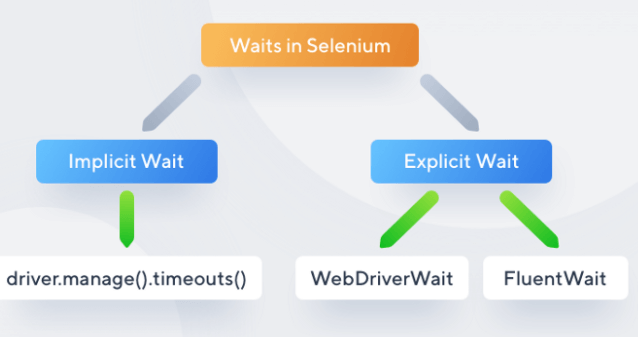
\*\*\*\***Implicit, Explicit and Fluent Wait in Selenium WebDriver:-**

In Selenium, “Waits” play an important role in executing tests.

There are different types of waits in Selenium:-

1. **Implicit Wait in Selenium:-**

## Types of Waits in Selenium



### \*\*\*\*\*\*\*\*\*Implicit Waits:-

The main function of implicit Wait is to tell the web driver to wait for some time before throwing a **"No Such Element Exception".** Its default setting is knocked at zero. Once the time is set, the driver automatically will wait for the amount of time defined by you before throwing the above given exception.

**\*\*\*\*\*\*\*Syntax:**

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

->Implicit wait will accept 2 parameters, the first parameter will accept the time as an integer value and the second parameter will accept the time measurement in terms of SECONDS, MINUTES, MILISECOND, MICROSECONDS, NANOSECONDS, DAYS, HOURS, etc.

->Suppose we are trying to find an element which has some “ExpectedConditions “ (Explicit Wait), If the element is not located within the time frame defined by the Explicit wait(10 Seconds), It will use the time frame defined by implicit wait(20 seconds) before throwing an “ElementNotVisibleException“.

**2.Explicit Wait in Selenium:-**

**Explicit Waits** also known as **Dynamic Waits,** because it is highly specific conditioned. It is implemented by WebDriverWait class. To understand why you need Explicit Wait in Selenium, you must go through the basic knowledge of the wait statements in a program. In simple terms, you must know some conditions. Such conditions have been created to give you a list of the Explicit Waits and why they are important.

**Condition 1:**

Suppose you have a web page consisting of a login form that takes input and loads the Home or Main page content. This page is dynamic because of the time constraints and network frequency, sometimes taking 10 seconds or maybe 15 seconds to load completely. Explicit Wait comes in handy in such cases and allows you to wait until the page is not present to display.

**Condition 2:**

Consider that you are working on an application that is travel themed and users fill the web form and submit it using submit button. Now, you might need to wait until and unless the specific data is not displayed. In such a case, Explicit Wait becomes helpful by waiting until a specific period for the set of elements that are not displayed yet.

## \*\*Difference between Implicit and Explicit:-

1. Implicit Wait applies to all the elements in the script, while Explicit Wait is applicable only for those values which are to be defined by the user.
2. Implicit Wait needs specifying "ExpectedConditions" on the located element, while Explicit Wait doesn't need to be specified with this condition.
3. Implicit Wait needs time frame specification in terms of methods like element visibility, clickable element, and the elements that are to be selected. In contrast, Explicit Wait is dynamic and needs no such specifications.

### Fluent Wait

Fluent Wait is quite similar to explicit Wait. It is similar in terms of management and functioning. In Fluent Wait, you can perform wait for action for an element only when you are unaware of the time it might take to be clickable or visible.

To understand how implicit wait works, let's consider an example.

**package** JavaTpoint;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**public** **class** ImplicitWait{

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

{

System.setProperty("webdriver.chrome.driver", "C:Selenium-java-

javaTpointchromedriver\_win32chromedriver.exe");

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

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

driver.manage().deleteAllCookies();

driver.manage().timeouts().pageLoadTimeout(40,TimeUnit.SECONDS); // pageload timeout

driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

// Implicit Wait for 20 seconds

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

driver.findElement(By.xpath("//input[@id='login- username']")).sendKeys("JavaTpoint.com");

//Finding element and

sending values

Thread.sleep(1000);

driver.findElement(By.xpath("//input[@id='login-signin']")).click(); //Clicking on the next button **if** element is located

}

}

In the code snippet given above, the Implicit Wait is defined for only **20 seconds,** implying that the output will load or arrive within the maximum waiting time of 20 seconds for the particular element.

#### Note: Implicit Wait is globally applied. It is readily available for driver instance. It also means that if the driver is having interaction with One thousand elements in the meantime, the implicit Wait will be applicable only for those 1000 elements. It cannot go beyond that.

**Syntax:**

1. WebDriverWait wait=**new**
2. WebDriverWait(WebDriveReference,TimeOut);

The above syntax justifies an object of WebDriver Wait and is passed to the driver's preference, and the timeout is taken as a parameter. To understand this more broadly, consider the below sample application.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Most of the web applications are developed using [Ajax](https://www.guru99.com/php-ajax.html) and [Javascript](https://www.guru99.com/interactive-javascript-tutorials.html). When a page is loaded by the browser the elements which we want to interact with may load at different time intervals.

Not only it makes this difficult to identify the element but also if the element is not located it will throw an “**ElementNotVisibleException**” exception. Using Selenium Waits, we can resolve this problem.

Let’s consider a scenario where we have to use both implicit and explicit waits in our test. Assume that implicit wait time is set to 20 seconds and explicit wait time is set to 10 seconds.

Suppose we are trying to find an element which has some  **“ExpectedConditions** “(Explicit Wait), If the element is not located within the time frame defined by the Explicit wait(10 Seconds), It will use the time frame defined by implicit wait(20 seconds) before throwing an “**ElementNotVisibleException**“.

**Selenium Web Driver Waits**

1. Implicit Wait
2. Explicit Wait

## \*\*\*\*\*Implicit Wait in Selenium

The **Implicit Wait in Selenium** is used to tell the web driver to wait for a certain amount of time before it throws a “No Such Element Exception”. The default setting is 0. Once we set the time, the web driver will wait for the element for that time before throwing an exception.

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

In the below example we have declared an implicit wait with the time frame of 10 seconds. It means that if the element is not located on the web page within that time frame, it will throw an exception.

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Explicit Wait in Selenium

The **Explicit Wait in Selenium** is used to tell the Web Driver to wait for certain conditions (Expected Conditions) or maximum time exceeded before throwing “ElementNotVisibleException” exception. It is an intelligent kind of wait, but it can be applied only for specified elements. It gives better options than implicit wait as it waits for dynamically loaded Ajax elements.

Once we declare explicit wait we have to use “**ExpectedConditions**” or we can configure how frequently we want to check the condition using **Fluent Wait**. These days while implementing we are using **Thread.Sleep()**generally it is not recommended to use

In the below example, we are creating reference wait for “**WebDriverWait**” class and instantiating using “**WebDriver**” reference, and we are giving a maximum time frame of 20 seconds.

**Explicit Wait syntax:**

WebDriverWait wait = new WebDriverWait(WebDriverRefrence,TimeOut);

\*\*\*\*\*In this WebDriver wait example, wait for the amount of time defined in the “**WebDriverWait**” class or the “**ExpectedConditions**” to occur whichever occurs first.

The above[Java](https://www.guru99.com/java-tutorial.html)code states that we are waiting for an element for the time frame of 20 seconds as defined in the “**WebDriverWait**” class on the webpage until the “**ExpectedConditions**” are met and the condition is “**visibilityofElementLocated**“.

The following are the Expected Conditions that can be used in Selenium Explicit Wait

1. alertIsPresent()
2. elementSelectionStateToBe()
3. elementToBeClickable()
4. elementToBeSelected()
5. frameToBeAvaliableAndSwitchToIt()
6. invisibilityOfTheElementLocated()
7. invisibilityOfElementWithText()
8. presenceOfAllElementsLocatedBy()
9. presenceOfElementLocated()
10. textToBePresentInElement()
11. textToBePresentInElementLocated()
12. textToBePresentInElementValue()
13. titleIs()
14. titleContains()
15. visibilityOf()
16. visibilityOfAllElements()
17. visibilityOfAllElementsLocatedBy()
18. visibilityOfElementLocated()

## \*\*\*\*\*\*\*\*Fluent Wait in Selenium

The **Fluent Wait in Selenium** is used to define maximum time for the web driver to wait for a condition, as well as the frequency with which we want to check the condition before throwing an “ElementNotVisibleException” exception. It checks for the web element at regular intervals until the object is found or timeout happens.

**Frequency:**Setting up a repeat cycle with the time frame to verify/check the condition at the regular interval of time

Let’s consider a scenario where an element is loaded at different intervals of time. The element might load within 10 seconds, 20 seconds or even more then that if we declare an explicit wait of 20 seconds. It will wait till the specified time before throwing an exception. In such scenarios, the fluent wait is the ideal wait to use as this will try to find the element at different frequency until it finds it or the final timer runs out.

**Fluent Wait syntax:**

Wait wait = new FluentWait(WebDriver reference)

.withTimeout(timeout, SECONDS)

.pollingEvery(timeout, SECONDS)

.ignoring(Exception.class);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**\*\*\*\*\*\*\*\*\*\*\*\*Web Application Testing - Techniques:**

**1.Verify there is no dead page or invalid redirects/valid URL or not.**

**2.First check all the validations on each field.**

**3.Wrong inputs to perform negative testing.**

**4.Verify the workflow of the system.**

**5.Verify the data integrity.**

### ****\*\*\*\*\*Selenium Components****

The Selenium test suite comprises four main components:-

* **Selenium IDE**
* **Selenium RC**
* **Selenium Webdriver**
* **Selenium Grid**

**\*\*\*\*\*Webdriver :-WebDriver is an API. It is written in more than one language which and they are called language bindings. The API has functions to control a browser.**

To receive the HTTP requests, every Browser Driver uses an HTTP server. Once the browser driver receives the URL, it processes the request by passing it to the real browser over HTTP. And then all your commands in the Selenium scripts will be executed.

**\*\*\*\*Types of Requests(in Postman,using that requests):-**

There are two types of requests you might be familiar with – **GET** and **POST**.

If it’s a GET request then it results in a response that will be generated at the browser end and it will be sent over HTTP to the browser driver and eventually, the browser driver with the help of JSON wire protocol sends it to the UI (Eclipse IDE).

**\*\*\*\*\*Selenium WebDriver:-**

Selenium WebDriver is a web framework that permits you to execute cross-browser tests. This tool is used for automating web-based application testing to verify that it performs expectedly.

Selenium WebDriver allows you to choose a programming language to create test scripts. Selenium WebDriver is not capable of handling window components, but this drawback can be overcome by using tools like Sikuli, Auto IT, etc.

### ****\*\*\*\*\*Selenium WebDriver Framework Architecture:-****

**WebDriver Architecture is made up of four major components:**

1. Selenium Client library
2. JSON wire protocol over HTTP
3. Browser Drivers
4. Browsers

**\*\*\*\*\*Selenium Client Libraries/Language Bindings:-**

Selenium provides support to multiple libraries such as Ruby, Python, Java, etc as language bindings have been developed by Selenium developers to provide compatibility for multiple languages. For instance, if you want to use the browser driver in Python, use the Python Bindings. You can download all the supported language bindings of your choice from the official site of Selenium.

**\*\*\*\*\*JSON Wire Protocol:-**

JSON is an acronym for JavaScript Object Notation. It is an open standard that provides a transport mechanism for transferring data between client and server on the web. It provides support for various data structures like arrays and objects which makes it easier to read and write data from JSON.

JSON serves as a REST (Representational State Transfer) API that exchanges information between HTTP servers.

**\*\*\*\*\*Browser Drivers:-**

Selenium provides drivers specific to each browser and without revealing the internal logic of browser functionality, the browser driver interacts with the respective browser by establishing a secure connection. These browser drivers are also specific to the language which is used for test case automation like [C#](https://www.browserstack.com/automate/c-sharp), [Python](https://www.browserstack.com/automate/python), [Java](https://www.browserstack.com/automate/java), etc.

You can download the browser driver of your choice as per your language requirements.

When a test script is executed with the help of WebDriver, the following tasks are performed in the background:-

* An HTTP request is generated and it is delivered to the browser driver for every Selenium Command.
* The HTTP request is received by the driver through an HTTP server.
* All the steps/instructions to be executed on the browser is decided by an HTTP server.
* The HTTP server then receives the execution status and in turn sends it back to the automation scripts.

**\*\*\*\*\*Browsers:-**

Selenium provides support for multiple browsers like Chrome, Firefox, Safari, Internet Explorer etc.

### ****\*\*\*\*\*Benefits of Selenium WebDriver:-****

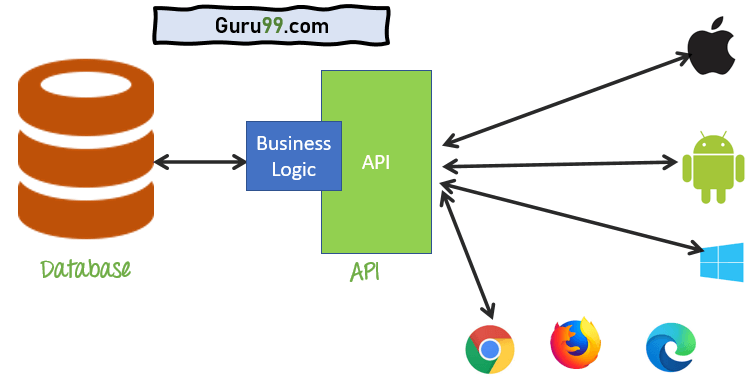
* It is one of the most popular Open-Source tools and is easy to get started with for testing web-based applications. It also allows you to perform [cross browser compatibility testing](https://www.browserstack.com/guide/cross-browser-compatibility-testing-beyond-chrome).
* Supports multiple operating systems like Windows, Mac, Linux, Unix, etc.
* It provides compatibility with a range of languages including Python, Java, Perl, Ruby, etc.
* Provides support for modern browsers like Chrome, Firefox, Opera, Safari, and Internet Explorer.
* Selenium WebDriver completes the execution of test scripts faster when compared to other tools
* More Concise API (Application Programming interface) than Selenium RC’s
* It also provides compatibility with iPhoneDriver, HtmlUnitDriver, and AndroidDriver

**\*\*\*\*\*API meaning:-**

**API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other.**

**Application Programming Interface (API)** is a software interface that allows two applications to interact with each other without any user intervention. API is a collection of software functions and procedures. In simple terms, API means a software code that can be accessed or executed. API is defined as a code that helps two different software’s to communicate and exchange data with each other.

It offers products or services to communicate with other products and services without having to know how they’re implemented.



Application Programming Interface

**\*\*\*\*\*\*\*If there is scenario like that:-**

**Q.1st : 1 website created,& it is tested on different Configuration, different browsers, portability is completed with that, if that site should not accessible for1st time to tester, and after all time it will accessible to tester whenever it will open that, then this issue tester should be add in defect report or not?**

**Ans :-**When that site will not accessible 1st time to tester, then there must be some cookies issue, at that time, this type of problem will be occurred.

**Q.2nd: As a role of QA what are u suggest for your team?**

**Ans :-**First I will check that , there should be SDLC cycle properly followed or not. Then I focused on quality of that Project/Product.

1 Review and analyze system specifications.

2.Collaborate with [**QA Engineers**](https://resources.workable.com/qa-engineer-job-description) to develop effective strategies and test plans.

3.Execute test cases (manual or automated) and analyze results.

4.Evaluate product code according to specifications.

5.Create logs to document testing phases and defects.

6.Report bugs and errors to development teams.

7.Help troubleshoot issues.

8.Conduct post-release/ post-implementation testing.

9.Work with cross-functional teams to ensure quality throughout the software development life cycle.

**Q 3)What are QA Tester Job Responsibilities and Duties?**

**Ans:-**1.Executes test cases under varying circumstances.

2.Documents and evaluates test results.

3.Detects, logs, and reports program bugs and glitches.

4.Tracks defects and helps troubleshoot errors.

5.Reviews test procedures and develops test scripts.

6.Partners with engineers to drive QA efforts.

**Q 4)What do you do if the bug is not reproducible? Or What do you mean by reproducing the bug?**

**Ans :-**

**Bug**: If there is a variation between the actual result and expected result, it is called as “Bug”.

**Case 1:-**You have found a bug while testing and posted it, if the developer is unable to find the same bug on their environment then they will ask you to reproduce the bug.

**Case 2:-**If the client found a bug on the production environment, they will inform us to fix it. In this case, as a tester we have to reproduce the bug in the test environment.

We all know that sometimes the bugs are not consistent. In the above 2 cases, if we are unable to reproduce the bug, we just mark the bug as inconsistent and close it temporarily with the status not reproducible/fixed. (depends on tool you use in your company and standard you follow in your company).

**Q 5) What are the fields in a bug/Defect report?**

Ans:-Following important fields are included in **Good Bug/Defect Report**:-

**1.A unique ID.**

**2.Defect** **Description**:-a short description what the bug is.

**3.Steps to reproduce**:-Details about how to arrive at the error, exact test data, the time at which defect was found environment :any information that will help reencounter the issue.

**4.Module/Section of the application(if applicable)**

**5.Severity**

**6.Screenshot**

**7.Reproducible QA** : in case of any follow-up questions regarding this issue.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Interview IMP Questions:-**

- Minimum 4 years of hands on experience on Selenium WebDriver, GRID, IDE, Web API,

Jenkins/Hudson CI tools

- Designing and Developing Automation Framework as well as Scripts.

- Strong in Core Java and OOPs concepts

- Proficient in writing well designed, testable, efficient code for Selenium test scripts.

- Good experience in Maven, ANT, TestNG

- Strong experience in working with POM, data driven, keyword driven and hybrid framework

- Strong knowledge and prior experience testing web services/APIs.

- Any experience with TDD and/or BDD will be an added advantage

- Good experience in any of API testing tools like SOAPUI/SOA/ParaSoft etc.

- Experience in Agile methodology and AWS Environment

**\*\*Distributed application:-**

A **distributed application** is a program that runs on more than one computer and communicates through a network. Some distributed applications are actually two separate software programs: the back-end (server) software and the front-end (client) software. Back-end software runs on a shared system (such as a shared Unix or VMS system) and manages shared resources, such as disks, printers, and modems. The back-end software also contains the main processing capability for the application. The front-end (client) software runs on workstations. It is the software you see when you use the application. It handles user interface functions, such as receiving input from a keyboard and displaying output to a screen.

Distributed applications can be relatively simple, requiring a single client computer and a single server, or more complex, allowing many client computers and several servers.

**For example**, web browsers are distributed applications. Browsers require back-end software (servers on the World Wide Web as well as front-end software installed on your workstation (e.g., Netscape Communicator or Internet Explorer).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*Middleware application:-**

Middleware in the context of distributed applications is **software that provides services beyond** those provided by the operating system to enable the various components of a distributed system to communicate and manage data. Middleware supports and simplifies complex distributed applications.

# \*\*TestNG Annotation Attributes:-

TestNG Parameters are the arguments that we pass to the test methods. There are two ways through which we can pass the parameters to the test methods:

**1.TestNG Parameters:-** Suppose we want to set the global variables such url settings, username, password or API Keys, there are some values which are constant in all the test cases, in such case we use the TestNG Parameters.

**2.TestNG Data Providers**:- Similar to TestNG Parameters, Data Providers are a means to pass data to test scripts in TestNG. Using Data Provider in TestNG, we can easily inject multiple values into the same test case. It comes inbuilt in TestNG and is popularly used in data-driven frameworks.

TestNG parameters enable us to pass the values only once per execution cycle. To overcome this, we can use DataProvider in TestNG that allows us to pass multiple parameters to a single test in a single execution. Using DataProviders, we can easily pass multiple values to a test in just one execution cycle.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*To handle the alert in Selenium Program:-**

1.For Clicking on **Ok button**:-

driver.switchTo().alert().accept();

2.For Clicking on **Cancel button**:-

driver().switchTo().alert().dismiss();

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**\*\*Popup Window Handling in Selenium:-**

1.To handle the **main window** when the site opens:-

**🡪driver.getWindowHandle();**

2.To handle all **opened windows(child windows)** by web driver:-

**🡪driver.getWindowHandles();**

3.To handle **popup windows:-**

**🡪 driver.switchTo().window(ChildWindow);**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*A list of differences between throw and throws are given below:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. no.** | **Basis of Differences** | **throw** | **throws** |
| 1. | Definition | Java throw keyword is used throw an exception explicitly in the code, inside the function or the block of code. | Java throws keyword is used in the method signature to declare an exception which might be thrown by the function while the execution of the code. |
| 2. | Type of exception | Using throw keyword, we can only propagate unchecked exception i.e., the checked exception cannot be propagated using throw only. | Using throws keyword, we can declare both checked and unchecked exceptions. However, the throws keyword can be used to propagate checked exceptions only. |
| 3. | Syntax | The throw keyword is followed by an instance of Exception to be thrown. | The throws keyword is followed by class names of Exceptions to be thrown. |
| 4. | Declaration | throw is used within the method. | throws is used with the method signature. |
| 5. | Internal implementation | We are allowed to throw only one exception at a time i.e. we cannot throw multiple exceptions. | We can declare multiple exceptions using throws keyword that can be thrown by the method. For example, main() throws IOException, SQLException. |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\* \*\* Types of Exceptions:-**

There are mainly two types of exceptions: checked and unchecked. An error is considered as the unchecked exception. However, according to Oracle, there are three types of exceptions namely:

1. **Checked Exception**
2. **Unchecked Exception**
3. **Error**

### 1) Checked Exception:-

The classes that directly inherit the **Throwable** class except **RuntimeException** and Error are known as checked exceptions. For example, IOException,SQLException, etc. Checked exceptions are checked at compile-time.

### 2) Unchecked Exception:-

The classes that inherit the **RuntimeException** are known as unchecked exceptions. For example, ArithmeticException, NullPointerException,ArrayIndexOutOfBoundsException, etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime.

# \*\*Features of TestNG:-

### 1.Multiple Before and After annotation options:-

Before and after annotations are used to execute a certain set of code before and after executing the test methods. Some of the Before and After annotations are @BeforeSuite, @BeforeTest, @BeforeGroups, @BeforeClass, etc.

### 2.XML-based test configuration:-

Test suites in a Testng are mainly configured by using the XML-based file. Testng.xml file is used to organize and run the test suites.

### 3.Dependent methods:-

Dependency works on the principle "depend-on-method" which must be either in the same class or in the inherited base class. This is the most important feature in TestNG that tells the TestNG to run the dependent test method after the execution of a given test method.

### 4.Groups/group of groups:-

TestNG groups allow you to group the test methods. By using TestNG groups, you can declare the methods in a group as well as you can declare the groups within a group.

### 5.Dependent groups:-

Similar to the Dependent methods, test methods in a group can depend on the test methods of another group.

### 6.Parameterization of test methods:-

One of the most important feature of TestNG is Parameterization. This feature allows you to pass the arguments as parameters and this achieved by using testng@Parameters annotation

### 7.Data-driven testing:-

TestNG allows users to perform data-driven testing. This testing allows users to execute the same test multiple times with multiple sets of data. To achieve the data-driven testing, DataProvider feature is used. DataProvider is a data feeder method that executes the test method with multiple sets of data.

### 8.Multithreaded execution:-

Multithreaded execution is the parallel execution of tests. Multithreading means the execution of multiple parts of software at the same time. Based on the configuration in the XML file, multiple threads are started, and test methods are executed in them. Multithreaded execution saves a lot of execution time.

### 9.Better reporting:-

Testng provides XML and HTML reports by default for test execution.

### 10.Open API:-

TestNG contains the open API means API is publicly available to the developers.

**\*\*Concept of Stack Memory & Heap Memory:-**



As you can see in the above figure, object gets the memory in heap memory area. The reference variable refers to the object allocated in the heap memory area. Here, s1 and s2 both are reference variables that refer to the objects allocated in memory.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*new keyword in Java:-

The new keyword is used to allocate memory at runtime. All objects get memory in Heap memory area.

we have created a Student class which has two data members id and name. We are creating the object of the Student class by new keyword and printing the object's value.

**//Creating an object or instance**

**Student s1=new Student();//creating an object of Student**

**//Printing values of the object**

**System.out.println(s1.id);//accessing member through reference variable**

**System.out.println(s1.name);**

### \*\*Instance variable in Java:-

A variable which is created inside the class but outside the method is known as an instance variable. Instance variable doesn't get memory at compile time. It gets memory at runtime when an object or instance is created. That is why it is known as an instance variable.

# \*\*Parallel Execution of test methods in TestNG:-

TestNG provides multiple ways to execute tests in separate threads. In testng.xml, if we set**'parallel'** attribute on the tag to 'tests', testNG will run all the ‘@Test’ methods in tag in the same thread, but each tag will be in a separate thread.

This helps us to run test methods / classes / tests in parallel. By using parallel execution, we can reduce the 'execution time' as tests are executed simultaneously in different threads.

In testNG we can achieve parallel execution by two ways. One with testng.xml file and we can configure an independent test method to run in multiple threads.

## \*\*HashSet in Java collection:-

HashSet class implements Set Interface. It represents the collection that uses a hash table for storage. Hashing is used to store the elements in the HashSet. It contains unique items.

**import** java.util.\*;

**public** **class** TestJavaCollection7{

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

//Creating HashSet and adding elements

HashSet<String> set=**new** HashSet<String>();

set.add("Ravi");

set.add("Vijay");

set.add("Ravi");

set.add("Ajay");

//Traversing elements

Iterator<String> itr=set.iterator();

**while**(itr.hasNext()){

System.out.println(itr.next());  }

}

}

**Output:-**

**Vijay**

**Ravi**

**Ajay**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# \*\*Java HashMap:-

Java **HashMap** class implements the Map interface which allows us to store key and value pair, where keys should be unique. If you try to insert the duplicate key, it will replace the element of the corresponding key. It is easy to perform operations using the key index like updation, deletion, etc. HashMap class is found in the java.util package.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **No.** | **StringBuffer** | **StringBuilder** | | 1) | StringBuffer is synchronized i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously. | StringBuilder is non-synchronized i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously. | | 2) | StringBuffer is less efficient than StringBuilder. | StringBuilder is more efficient than StringBuffer. | | 3) | StringBuffer was introduced in Java 1.0 | StringBuilder was introduced in Java 1.5 | |

**\*\*\*\*\*\*\*\*\*\*\*Difference between HashMap & Hashtable:-**

|  |  |
| --- | --- |
| **HashMap** | **Hashtable** |
| 1) HashMap is **non synchronized**. It is not-thread safe and can't be shared between many threads without proper synchronization code. | Hashtable is **synchronized**. It is thread-safe and can be shared with many threads. |
| 2) HashMap **allows one null key and multiple null values**. | Hashtable **doesn't allow any null key or value**. |
| 3) HashMap is a **new class introduced in JDK 1.2**. | Hashtable is a **legacy class**. |
| 4) HashMap is **fast**. | Hashtable is **slow**. |
| 5) We can make the HashMap as synchronized by calling this code Map m = Collections.synchronizedMap(hashMap); | Hashtable is internally synchronized and can't be unsynchronized. |
| 6) HashMap is **traversed by Iterator**. | Hashtable is **traversed by Enumerator and Iterator**. |
| 7) Iterator in HashMap is **fail-fast**. | Enumerator in Hashtable is **not fail-fast**. |
| 8) HashMap inherits **AbstractMap** class. | Hashtable inherits **Dictionary** class. |

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Dataprovider in TestNG:-**

To achieve the data-driven testing, DataProvider feature is used. DataProvider is a data feeder method that executes the test method with multiple sets of data.

**\*\* Principles of Software Testing:-**

The seven different testing principles, one by one:-

**1.Testing shows the presence of defects:-**Through testing we can find out defect present in application.

**2.Exhaustive Testing is not possible:-**Details/Overall/whole project testing is not possible.

**3.Early Testing:-**By reading/referring req. specification documents, before development of application, start testing process.

**4.Defect Clustering:-**We can detect/find out the no. of bugs in small modules in application.

**5.Pesticide Paradox:-**New & different test cases are necessary,for multiple parts of application, which helps to find out the bugs.

**6.Testing is context-dependent:-** Every site has its own needs, features, functionality then use various testing ,different testing approaches, multiple methods to test that data.

**7.Absence of errors fallacy:-**Identifying & fixing the bugs not help ,if application is impractical/not in practical use & not able to accomplish the clients requirements & needs.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**\*\* FindElement and FindElements in Selenium WebDriver:-**

**1)FindElement command syntax:-**

**\*\*FINDELEMENT() METHOD:-**

\*\*findElement method is used to access a single web element on a page. It returns the first matching element. It throws a  NoSuchElementException exception when it fails to find the element.

Syntax:

WebElement elementName = driver.findElement(By.LocatorStrategy("LocatorValue"));

|  |  |
| --- | --- |
| 1 | .driver.findElement(By.xpath("Value of Xpath")); |

**2) FINDELEMENTS() METHOD:**

findElements method returns the list of all matching elements. The findElement method throws a NoSuchElementException exception when the element is not available on the page. Whereas, the findElements method returns  an empty list when the element is not available or doesn’t exist on the page. It doesn’t throw NoSuchElementException.

**Syntax:**

List<WebElement> elementName = driver.findElements(By.LocatorStrategy("LocatorValue"));

|  |  |
| --- | --- |
|  |  |
| **1** | **List link = driver.findElements(By.xpath("Value of Xpath"));** |

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **findElement** | **findElements** |
| **1** | Returns the first most web element if there are multiple web elements found with the same locator. | Returns a list of web elements. |
| **2** | 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. |
| **3** | Find element by XPath will only find one web element. | It will find a collection of elements whose match the locator strategy. |
| **4** | Not Applicable | Each Web element is indexed with a number starting from 0 just like an array. |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

## \*\*\*What is an application programming interface (API)?

An application programming interface, or API, enables companies to open up their applications’ data and functionality to external third-party developers, business partners, and internal departments within their companies. This allows services and products to communicate with each other and leverage each other’s data and functionality through a documented interface.

Developers don't need to know how an API is implemented; they simply use the interface to communicate with other products and services. API use has surged over the past decade, to the degree that many of the most popular web applications today would not be possible without APIs.

An API is a set of defined rules that explain how computers or applications communicate with one another. APIs sit between an application and the web server, acting as an intermediary layer that processes data transfer between systems.

**\*\*\*\*\*\*How an API works:-**

A client application initiates an API call to retrieve information—also known as a request. This request is processed from an application to the web server via the API’s Uniform Resource Identifier (URI) and includes a request verb, headers, and sometimes, a request body.

After receiving a valid request, the API makes a call to the external program or web server.

The server sends a response to the API with the requested information.

The API transfers the data to the initial requesting application.

While the data transfer will differ depending on the web service being used, this process of requests and response all happens through an API. Whereas a user interface is designed for use by humans, APIs are designed for use by a computer or application.

# \*\*\*\*\*What is an API? (Application Programming Interface)

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you’re using an API.

## \*\*\*What Is an Example of an API?

When you use an application on your mobile phone, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your phone. The application then interprets that data and presents you with the information you wanted in a readable way. This is what an API is - all of this happens via API.

Imagine you’re sitting at a table in a restaurant with a menu of choices to order from. The kitchen is the part of the “system” that will prepare your order. What is missing is the critical link to communicate your order to the kitchen and deliver your food back to your table. That’s where the waiter or API comes in. The waiter is the messenger – or API – that takes your request or order and tells the kitchen – the system – what to do. Then the waiter delivers the response back to you; in this case, it is the food.

Here is a real-life API example. You may be familiar with the process of searching flights online. Just like the restaurant, you have a variety of options to choose from, including different cities, departure and return dates, and more. Let us imagine that you’re booking you are flight on an airline website. You choose a departure city and date, a return city and date, cabin class, as well as other variables. In order to book your flight, you interact with the airline’s website to access their database and see if any seats are available on those dates and what the costs might be.

However, what if you are not using the airline’s website––a channel that has direct access to the information? What if you are using an online travel service, such as Kayak or Expedia, which aggregates information from a number of airline databases?

The travel service, in this case, interacts with the airline’s API. The API is the interface that, like your helpful waiter, can be asked by that online travel service to get information from the airline’s database to book seats, baggage options, etc. The API then takes the airline’s response to your request and delivers it right back to the online travel service, which then shows you the most updated, relevant information.

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HTTP vs HTTPS:

## \*\*\*What is HTTP?

Full form of HTTP is **Hypertext Transfer Protocol**. HTTP offers set of rules and standards which govern how any information can be transmitted on the World Wide Web. HTTP provides standard rules for web browsers & servers to communicate.

HTTP is an application layer network protocol which is built on top of TCP.

## \*\*\*What is HTTPS?

HTTPS stands for **Hyper Text Transfer Protocol Secure**. It is highly advanced and secure version of HTTP. It uses the port no. 443 for Data Communication. It allows the secure transactions by encrypting the entire communication with SSL. It is a combination of SSL/TLS protocol and HTTP. It provides encrypted and secure identification of a network server.

HTTP also allows you to create a secure encrypted connection between the server and the browser.

**\*\*\*\*\*KEY DIFFERENCE:-**

* HTTP lacks security mechanism to encrypt the data whereas HTTPS provides SSL or TLS Digital Certificate to secure the communication between server and client.
* HTTP operates at Application Layer whereas HTTPS operates at Transport Layer.
* HTTP by default operates on port 80 whereas HTTPS by default operates on port 443.
* HTTP transfers data in plain text while HTTPS transfers data in cipher text (encrypt text).
* HTTP is fast as compared to HTTPS because HTTPS consumes computation power to encrypt the communication channel.

## Difference Between HTTP and HTTPS:-

|  |  |  |
| --- | --- | --- |
| **Parameter** | **HTTP** | **HTTPS** |
| **Protocol** | It is hypertext transfer protocol. | It is hypertext transfer protocol with secure. |
| **Security** | It is less secure as the data can be vulnerable to hackers. | It is designed to prevent hackers from accessing critical information. It is secure against such attacks. |
| **Port** | It uses port 80 by default. | It was use port 443 by default. |
| **Starts with** | HTTP URLs begin with http:// | HTTPs URLs begin with https:// |
| **Used for** | It’s a good fit for websites designed for information consumption like blogs. | If the website needs to collect the private information such as credit card number, then it is a more secure protocol. |
| **Scrambling** | HTTP does not scramble the data to be transmitted. That’s why there is a higher chance that transmitted information is available to hackers. | HTTPS scrambles the data before transmission. At the receiver end, it descrambles to recover the original data. Therefore, the transmitted information is secure which can’t be hacked. |
| **Protocol** | It operates at [**TCP/IP**](https://www.guru99.com/tcp-ip-model.html) level. | HTTPS does not have any **separate protocol**. It operates using HTTP but uses encrypted TLS/SSL connection. |
| **Domain Name Validation** | HTTP website do not need SSL. | HTTPS requires SSL certificate. |
| **Data encryption** | HTTP website doesn’t use encryption. | HTTPS websites use data encryption. |
| **Search Ranking** | HTTP does not improve search rankings. | HTTPS helps to improve search ranking. |
| **Speed** | Fast | Slower than HTTP |
| **Vulnerability** | It is Vulnerable to hackers. | It Is highly secure as the data is encrypted before it is seen across a network. |
| **Layer** | It operates at Application Layer. | It operates at Transport Layer. |

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Shital concepts about Manual Testing:-**

**Severity & Priority examples(from guru99),Smoke & Sanity difference, Authentication & Authorization difference, http & https difference, retesting & regression testing ,Test scenario, Test Cases, Exploratory Testing, Ad hoc Testing, Practical examples of our project when u r handled that type of testing(for Testing type and for Severity & Priority), Real scenario related to Severity & Priority..**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SQL Concepts:-**

**Difference between Drop, Delete &Truncate Command, Find 2nd highest salary from Emp table, Use of count, Group by with Where Clause ,Order by clause, Aggregation Function, DDL & DML Command, SQL queries from w3c school, Joins, Distinct Command, Count Command…**

# \*\*\*Cookies and Session Description:-

**\*\*\*\*\*Testing done on web application by checking following criteria:-**

**Security Testing:-**

1.Cookie Testing

2.Authentication(Validity of User).

3.Authorization(Level of Access).

4.Password Testing.

**\*\*\*\*Contents of Cookies:-**

1.Server Name, Sent From(Server Information).

2.Cookies Lifetime.

3.A Value.

4.Information or data about user.

**\*\*\*2 Types of Cookies:-**

1. Session Cookies.

2. Persistent Cookies.

## \*\*\*\*\*What is Cookie?

**A Cookie is temporary text file saved on server in the encrypted format to the user side.**

A Cookie is a small file with the maximum size of 4KB that the web server stores on the client computer. Once a cookie has been set, all page requests that follow return the cookie name and value. A cookie can only be read from the domain that it has been issued from.

For example, a cookie set using the domain www.guru99.com cannot be read from the domain career.guru99.com. Most of the websites on the internet display elements from other domains such as advertising. The domains serving these elements can also set their own cookies. These are known as third party cookies. A cookie created by a user can only be visible to them. Other users cannot see its value. Most web browsers have options for disabling cookies, third party cookies or both.

## \*\*\*\*\*Why and when to use Cookies?

* **A Cookie is a small text file that is stored on the user's computer**. The maximum file size of a cookie is **4KB**. It is also known as an HTTP cookie, web cookie, or internet Cookie. Whenever a user visits a website for the first time, the site sends packets of data in the form of a cookie to the user's computer.
* The cookies help the websites to keep track of the user's browsing history or cart information when they visit their sites.
* It stores only the "String" data type.
* The information stored within cookies is not secure because this information is stored in text-format on the client-side, which can be read by anyone.
* We can enable or disable the cookies as per the requirement.
* The cookies generated by a user are only be shown to them, and no other user can see those cookies.
* Cookies are created and shared between the server and browser with the help of an HTTP header.
* The path where the cookies are saved is decided by the browser, as Internet explorer usually stored them in **Temporal Internet File Folder**.
* When we visit YouTube channel and search for some songs, next time whenever we visit YouTube, cookies read our browsing history and shows similar songs or last played songs.

## \*\*\*\*\*What is a Session?

**A session is used to temporarily store the information on the server to be used across multiple pages of the website**.

The session values are automatically deleted when the browser is closed. If you want to store the values permanently, then you should store them in the database.

## \*\*\*\*\*Why and when to use Sessions?

* It is the total time used for an activity. The user session starts when he logs-in to a particular network application and ends when the user logs out from the application or shutdowns the system.
* When we work on an application over the internet, the webserver doesn't know the user because the HTTP protocol does not maintain the state. The information provided by the user on one page of the application (Let's say Home) will not be transferred to another page. To remove this limitation, sessions are used. Session gets started whenever a visitor first enters a website.
* The user information is stored in session variables, and these variables can store any type of value or data type of an Object.
* Session values are much secured as these are stored in binary form or encrypted form and can only be decrypted at the server. The session values are automatically removed when the user shutdowns the system or logout from the application. To store the values permanently, we need to store them in the database.
* Each session is unique for each user, and any number of sessions can be used in an application; there is no limitation to it.
* The user is identified with the help of **sessionID**, which is a unique number saved inside the server. It is saved as a **cookie, form field, or URL.**

## \*\*\*\*\*Key Differences between Session and Cookies:-

* Cookies are client-side files that contain user information on a local computer whereas Sessions are server-side files that store the user information.
* Cookies are not dependent on Session whereas Sessions are cookies dependent.
* Cookies expire at the set time,whereas the Session ends when the user closes the browser or logout from the application.
* A session can store as much data as a user want, whereas Cookies have a limited size of 4KB.

# Difference between Cookie and Session:-

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Cookie** | **Session** |
| **1** | Cookies are client-side files that contain user information | Sessions are server-side files which contain user information. |
| **2** | Cookie ends depending on the lifetime you set for it. | A session ends when a user closes his browser. |
| **3** | Cookies are saved on the user's browser or client-side. | Sessions are saved on the server-side. |
| **4** | The official maximum cookie size is 4KB. | Within-session you can store as much data as you like. The only limits you can reach is the maximum memory a script can consume at one time, which is 128MB by default. |
| **5** | A cookie is not dependent on Session. | A session is dependent on Cookie. |
| **6** | There is no function named unsetcookie(). | Session\_destroy(); is used to destroy all registered data or to unset some. |

**\*\*\*ArrayList and LinkedList:-**

ArrayList and LinkedList both implements List interface and maintains insertion order. Both are non synchronized classes.

\*\*\*\*Differences between ArrayList and LinkedList classes that are given below.

|  |  |
| --- | --- |
| **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) 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. |

**\*\*\*Java HashSet:-**

Java HashSet class is used to create a collection that uses a hash table for storage. It inherits the AbstractSet class and implements Set interface.

The important points about Java HashSet class are:

* HashSet stores the elements by using a mechanism called **hashing.**
* HashSet contains unique elements only.
* HashSet allows null value.
* HashSet class is non synchronized.
* HashSet doesn't maintain the insertion order. Here, elements are inserted on the basis of their hashcode.
* HashSet is the best approach for search operations.
* The initial default capacity of HashSet is 16, and the load factor is 0.75.

**\*\*\* Java HashMap:-**

\*\*\* Java **HashMap** class implements the Map interface which allows us to store key and value pair, where keys should be unique. If you try to insert the duplicate key, it will replace the element of the corresponding key. It is easy to perform operations using the key index like updation, deletion, etc. HashMap class is found in the java.util package.

HashMap in Java is like the legacy Hashtable class, but it is not synchronized. It allows us to store the null elements as well, but there should be only one null key. Since Java 5, it is denoted as HashMap<K,V>, where K stands for key and V for value. It inherits the AbstractMap class and implements the Map interface.

### Points to remember

* Java HashMap contains values based on the key.
* Java HashMap contains only unique keys.
* Java HashMap may have one null key and multiple null values.
* Java HashMap is non synchronized.
* Java HashMap maintains no order.
* The initial default capacity of Java HashMap class is 16 with a load factor of 0.75.

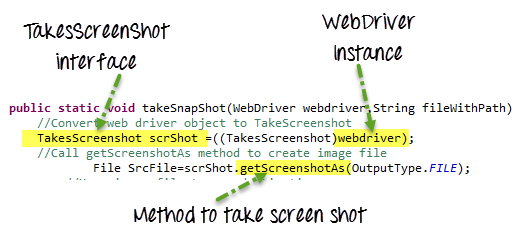
### Hierarchy of HashMap class

As shown in the above figure, HashMap class extends AbstractMap class and implements Map interface.

# \*\*\*How to Take Screenshot in Selenium WebDriver:-

## Screenshot in Selenium Webdriver

A **Screenshot in Selenium Webdriver** is used for bug analysis. Selenium webdriver can automatically take screenshots during the execution. But if users need to capture a screenshot on their own, they need to use the TakeScreenshot method which notifies the WebDrive to take the screenshot and store it in Selenium.



## Capture Screenshot using Selenium WebDriver:-

Taking Screenshot in Selenium is a 3 Step process

**Step 1) Convert web driver object to TakeScreenshot**

TakesScreenshot scrShot =((TakesScreenshot)webdriver);

**Step 2) Call getScreenshotAs method to create image file**

File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

**Step 3) Copy file to Desired Location**

Example: In this example we will take screen capture of http://demo.guru99.com/V4/ & save it as C:/Test.png

package Guru99TakeScreenshot;

import java.io.File;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.Test;

public class Guru99TakeScreenshot {

@Test

public void testGuru99TakeScreenShot() throws Exception{

WebDriver driver ;

System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");

driver = new FirefoxDriver();

//goto url

driver.get("http://demo.guru99.com/V4/");

//Call take screenshot function

this.takeSnapShot(driver, "c://test.png") ;

}

/\*\*

\* This function will take screenshot

\* @param webdriver

\* @param fileWithPath

\* @throws Exception

\*/

public static void takeSnapShot(WebDriver webdriver,String fileWithPath) throws Exception{

//Convert web driver object to TakeScreenshot

TakesScreenshot scrShot =((TakesScreenshot)webdriver);

//Call getScreenshotAs method to create image file

File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

//Move image file to new destination

File DestFile=new File(fileWithPath);

//Copy file at destination

FileUtils.copyFile(SrcFile, DestFile);

}

}

\*\*\*\***Abstract Class Vs Concrete Class:-**

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Abstract Class** | **Concrete Class** |
| **1** | A class declared with an abstract keyword which is a collection of abstract & non-abstract methods. | A class that allows to create an instance or an object using the new keyword. |
| **2** | Programmer cannot create an object using Abstract class. | Programmer can create an object using Concrete class. |
| **3** | An Abstract class can have a method of an unimplemented classes. | All methods in a Concrete class are implemented. |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**\*\*\*\*\*\*Technical challenges with Selenium are:-**

* Selenium supports only web based applications
* It does not support the Bitmap comparison
* For any reporting related capabilities have to depend on third party tools
* No vendor support for tool compared to commercial tools like HP UFT
* As there is no object repository concept in Selenium, maintainability of objects becomes difficult

#### \*\*\*\*features of TestNG and list some of the functionality in TestNG which makes it more effective:-

* Support for annotations
* Support for data-driven testing
* Flexible test configuration
* Ability to re-execute failed test cases

#### \*\*\*\*\* Data driven framework and Keyword driven:-

**\*\*Data driven framework:**  In this framework, the test data is separated and kept outside the Test Scripts, while[**Test Case**](https://www.guru99.com/test-case.html)logic resides in Test Scripts.  Test data is read from the external files ( Excel Files) and are loaded into the variables inside the Test Script.  Variables are used for both for input values and for verification values.

**\*\*Keyword driven framework:** The keyword driven frameworks requires the development of data tables and keywords, independent of the test automation.  In a keyword driven test, the functionality of the application under test is documented in a table as well as step by step instructions for each test.

#### \*\*\*\*\*Capabilities of Selenium WebDriver or Selenium 2.0 :-

WebDriver should be used when requiring improvement support for

* Handling multiple frames, pop ups , multiple browser windows and alerts
* Page navigation and drag & drop
* Ajax based UI elements
* Multi browser testing including improved functionality for browser not well supported by Selenium 1.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **getwindowhandles & getwindowhandle**:-

getwindowhandles(): It is used to get the address of all the open browser and its return type is Set<String>

getwindowhandle(): It is used to get the address of the current browser where the control is and return type is string.

\*\*\*\*\***Limitations of Selenium IDE:-**

1. Exceptional handling is not present.

2. Selenium IDE uses only HTML languages.

3. External databases reading is not possible with IDE.

4. Reading from the external files like .txt, .xls is not possible.

5. Conditional or branching statements execution like if, else, select statements is not possible.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Intellij :-

Intellij is an IDE that helps you to write better and faster code for Selenium. Intellij can be used in the option to Java bean and Eclipse.