



Date: 10-10-2018

DEMONSTRATION

ON

**"Selenium testing tool"**

CIRCULAR


COSMOS, department forum of CS&E, has arranged a demonstration on "Selenium testing tool". This event is for pre-final year students of the department.

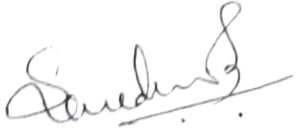
Resource Person Details:


Mr. Mohan H. G.,  
Assistant Professor,  
Dept. of Computer Science and Engg.,  
JNNCE, Shivamogga

**Date and Time:** 12<sup>th</sup> October 2018 at 02.30 pm.

**Venue:** AD Block Seminar Hall,  
JNNCE.

  
**Mr. Narendra Kumar S.**  
Cosmos Co-ordinator  
Asst. Professor  
Dept. of CS & E  
JNNCE, Shivamogga

  
**Mrs. Sreedevi S.**  
Cosmos Co-ordinator  
Asst. Professor  
Dept. of CS & E  
JNNCE, Shivamogga

  
**Dr. Nirmala Shivananda**  
Professor and Head  
Dept. of CS & E  
JNNCE, Shivamogga



## DEMONSTRATION OF "SELENIUM TESTING TOOL"

### STUDENT ATTENDANCE FORM

Date:	12-10-2018	Resource	Mr. Mohan H. G.
Time:	02.30 PM – 4.30 PM	Person:	Assistant Professor Dept. of CS & E, JNNCE
Venue:	AD Block Seminar Hall, JNNCE.		

Si. No.	USN	NAME	Signature
1.	4JN14CS002	AIMAN FATHIMA	Aiman Fathima
2.	4JN14CS092	SERWESH KUMAR SHIROLE Y	Ab
3.	4JN15CS025	GAHANA P	Ab
4.	4JN15CS041	MANOJ N	Manoj N
5.	4JN15CS049	NAVEENANAİK L	Naveen
6.	4JN15CS067	PRIYAMVADA L KALAMDANI	Priyamvada
7.	4JN16CS003	ADITYA B IYER	Aditya B Iyer
8.	4JN16CS008	AMRUTHA M V	Amrutha M V
9.	4JN16CS010	ANUSHREE P	Anushree P
10.	4JN16CS011	APOORVA G	P Anu
11.	4JN16CS012	APOORVA K R	Apoorva K R
12.	4JN16CS013	APOORVA V S	K. R. Apoorva
13.	4JN16CS014	ARPITHA J A	Arpitha JA
14.	4JN16CS015	AVINASH H R	Avinash H R
15.	4JN16CS016	BHOOMIKA S	Bhoomika S
16.	4JN16CS019	BINDU N L	Bindu N. L
17.	4JN16CS020	CHETHANA B H	Chethana B H
18.	4JN16CS021	CHIRANTHAN K S	Chiranthan K S
19.	4JN16CS022	CHOMINI D K	Chomini D K
20.	4JN16CS023	DEEPA SURENDRA RAYAKAR	DK Chomni
21.	4JN16CS024	DEEPIKA H R	Ab



Jawaharlal Nehru National College of Engineering, Shivamogga  
Department of Computer Science and Engineering

22.	4JN16CS027	HALIMA ERUM	Halima Erum
23.	4JN16CS028	HARSHITHA M H	Harshitha
24.	4JN16CS029	JYOTHI J R	Jyothi J.R
25.	4JN16CS030	JYOTHI T S	Jyothi T.S
26.	4JN16CS032	KARTHIK MALODE	Karthik Malode
27.	4JN16CS034	KARTHIK P	Karthik P
28.	4JN16CS036	KAVANA K P	Kavana K.P
29.	4JN16CS037	MADHURA V M	Madhura V.M
30.	4JN16CS040	MANJUNATHA V AMBORE	Manjunatha V.
31.	4JN16CS041	MARELLA JESSICA	Jessica
32.	4JN16CS044	NAMRUTA VASU BHAT	Namruta
33.	4JN16CS046	NEELESH V S	Neelesh V.S
34.	4JN16CS051	PALLAVI N G	Pallavi N.G
35.	4JN16CS053	PANNAGA Y S	Pannaga Y.S
36.	4JN16CS054	PAVANA B	Pavana B
37.	4JN16CS055	PAWAN P	Pawan P
38.	4JN16CS057	POOJA M S	Pooja M.S
39.	4JN16CS058	POOJA S	Pooja S
40.	4JN16CS059	POOJA SHASTRY	Pooja Shastri
41.	4JN16CS062	PRAJWAL M	Prajwal M
42.	4JN16CS064	PRATHIBHA K M	Prathibha K.M
43.	4JN16CS046	NEELESH V S	Neelesh V.S
44.	4JN16CS051	PALLAVI N G	Pallavi N.G
45.	4JN16CS078	RASHMI RAO B	Rashmi Rao B
46.	4JN16CS079	REBONI SHAMMA	Reboni Shamma
47.	4JN16CS082	SAHANA G	Sahana G
48.	4JN16CS084	SANJANA S R	Sanjana S.R
49.	4JN16CS096	SINCHANA R RAO	Sinchana R.Rao
50.	4JN16CS097	SINCHANA S J	Sinchana S.J
51.	4JN16CS104	SUPREETHA H V	Supreetha H.V
52.	4JN16CS106	SUPRIYA S P	Supriya S.P
53.	4JN16CS107	SURABHI S R	Surabhi S.R
54.	4JN16CS110	TANYA V	Tanya V
55.	4JN16CS118	VARSHA V BADAMI	Varsha V Badami
56.	4JN16CS122	VIVEK V KINI	Vivek V Kini
57.	4JN16CS127	APOORVA K G	Apoorva K.G





Jawaharlal Nehru National College of Engineering, Shivamogga  
Department of Computer Science and Engineering

58.	4JN16CS128	KAVYA KRISHNAMURTY HEGDE	Key
59.	4JN16CS130	CHANDANA B N	Chandana B.N
60.	4JN17CS400	ARUN M	Arun.M
61.	4JN17CS401	BHAVANI V M	Bhavani
62.	4JN17CS404	CHANDRASHEKARA B	Chandra
63.	4JN17CS405	DEEPIKA J	Deepika.J
64.	4JN17CS406	DINESH T	D.S.T
65.	4JN17CS408	GAUTHAM J	Gautham
66.	4JN17CS410	GIRISHA M N	Girish M.N
67.	4JN17CS411	GOURAV G REVANKAR	GOURAV
68.	4JN17CS413	KAVYA M	Kavya.M
69.	4JN17CS414	KISHAN C M	Kishan C.M
70.	4JN17CS417	POOJA B N	Pooja
71.	4JN17CS420	SEVANTHI B C	Sevanti
72.	4JN17CS422	SHILPA H B	Shilpa
73.	4JN17CS424	T A BHARATH	Bharath T.A.
74.	4JN17CS425	VARSHA M H	Varsha M.H.
75.	4JN17CS427	YASHODHA S	Yashodha S.
76.	4JN14CS133	RUMANA	Ab
77.	4JN16CS001	ABHAY K P	Abhay.K.P
78.	4JN16CS005	AISHWARYA V K	Aishwarya
79.	4JN16CS006	AKSHAY YADAV	Akshay
80.	4JN16CS007	AMITH KUMAR H S	Amith
81.	4JN16CS009	AMULYA S R	Ab
82.	4JN16CS017	BI BI ZAKIYA	Zakiya
83.	4JN16CS018	BINDU K R	Bindu K.R
84.	4JN16CS026	DHANYA V	Dhanya
85.	4JN16CS031	K C CHITRA	K.C.Chitra
86.	4JN16CS033	KARTHIK N UDUPA	Karthik
87.	4JN16CS035	KAVANA D M	Kavana
88.	4JN16CS038	MANJUNATH JOGI H K	Manjunath
89.	4JN16CS039	MANJUNATHA D	Ab
90.	4JN16CS042	MEGHANA S	Meghana S
91.	4JN16CS045	NANDINI G M	Nandini
92.	4JN16CS047	NIDA	Nida
93.	4JN16CS048	NIKHIL K A	Nikhil K.A




Jawaharlal Nehru National College of Engineering, Shivamogga  
Department of Computer Science and Engineering

94.	4JN16CS049	NITHIN KUMAR S K	Nithin Kumar S K
95.	4JN16CS052	PANNAGA G R	Pannaga G R
96.	4JN16CS056	POOJA A J	Pooja A J
97.	4JN16CS060	POOJASHREE T	Poojashree T
98.	4JN16CS061	PRAJWAL A P	Prajwal A P
99.	4JN16CS063	PRASHANTH N	Prashanth N
100.	4JN16CS065	PREETHAM KRISHNA	Preetham Krishna
101.	4JN16CS066	PREMARAKSHA CHANDRASHEKHAR MALI	Preetham Krishna
102.	4JN16CS067	PRIYA P SHETTY	Priya P Shetty
103.	4JN16CS070	RAJESH K	Rajesh K
104.	4JN16CS071	RAMACHANDRA ASHOK	Ramachandra Ashok
105.	4JN16CS072	RAMANATH PAI M D	Ramanath Pai M D
106.	4JN16CS073	RAMYA H L	Ramyah L
107.	4JN16CS077	RASHMI B S	Rashmi B S
108.	4JN16CS080	S M SINDHU	S M Sindhu
109.	4JN16CS083	SALMANULLA R	Salmanulla R
110.	4JN16CS085	SANJU B K	Sanju B K
111.	4JN16CS087	SAVITHA G	Savitha G
112.	4JN16CS088	SHARATH C	Sharath C
113.	4JN16CS089	SHARMEEN SADAF	Sharmeen Sadaf
114.	4JN16CS090	SHASHANK L NAGAR	Shashank L Nagar
115.	4JN16CS091	SHASHANK T P	Shashank T P
116.	4JN16CS093	SHREEVATHSA M L	Shreevathsa M L
117.	4JN16CS094	SHUBHAVI B R	Shubhavi B R
118.	4JN16CS095	SINCHANA B	Sinchana B
119.	4JN16CS099	SONIKA V	Sonika V
120.	4JN16CS100	SPOORTHI M P	Spoorthi M P
121.	4JN16CS101	SPOORTHY M	Spoorthy M
122.	4JN16CS102	SUCHETAN J	Suchetan J
123.	4JN16CS103	SUHAS BHARADWAJ K	Suhas Bharadwaj K
124.	4JN16CS105	SUPRIYA S KATTE	Supriya S Katte
125.	4JN16CS108	SURYA S	Surya S
126.	4JN16CS109	SYEDA SABHA SIMRA	Syeda Sabha Simra
127.	4JN16CS112	THANMAYEE T G	Thanmayee T G
128.	4JN16CS113	THANUJA P C	Thanuja P C



Jawaharlal Nehru National College of Engineering, Shivamogga  
Department of Computer Science and Engineering

129	4JN16CS114	TULASI D S	Tulasi
130	4JN16CS115	ULLAS C	Ullas
131	4JN16CS116	VAISHAK M B	Vaishak M B
132	4JN16CS117	VAISHANAVI M SAJJAN	Vaishanavi M Sajjan
133	4JN16CS119	VIDYA H G	Vidya H G
134	4JN16CS120	VIGNESHA B R	Vignesha B R
135	4JN16CS121	VISHWANATH N P	Vishwanath N P
136	4JN16CS123	ZAIBA KHANUM	Zaiba
137	4JN16CS124	KRUPA C	Krupa C
138	4JN16CS125	MADHURA BHAT	Madhura Bhat
139	4JN16CS126	MEGHANA V	Meghana V
140	4JN16CS129	SUSHMA S	Sushma S
141	4JN16CS131	MANAMOHAN G WALI	Manamohan G Wali
142	4JN16CS132	ANAGHA V BHARADWAJ	Anagha V Bharadwaj
143	4JN17CS402	BI BI HAFSA	Bi Bi Hafsa
144	4JN17CS403	CHAITHRASHREE R	Chaithrashree R
145	4JN17CS407	GANGA G	Ganga G
146	4JN17CS412	HUSNA AFREEN	Husna Afreen
147	4JN17CS416	POOJA B N	Pooja B N
148	4JN17CS418	RAMYA G T	Ramya G T
149	4JN17CS419	REVATHI H	Revathi H
150	4JN17CS423	SUPREETHA SAAMAK N R	Supreetha Saamak N R

  
 Coordinator Signature

H.O.D.



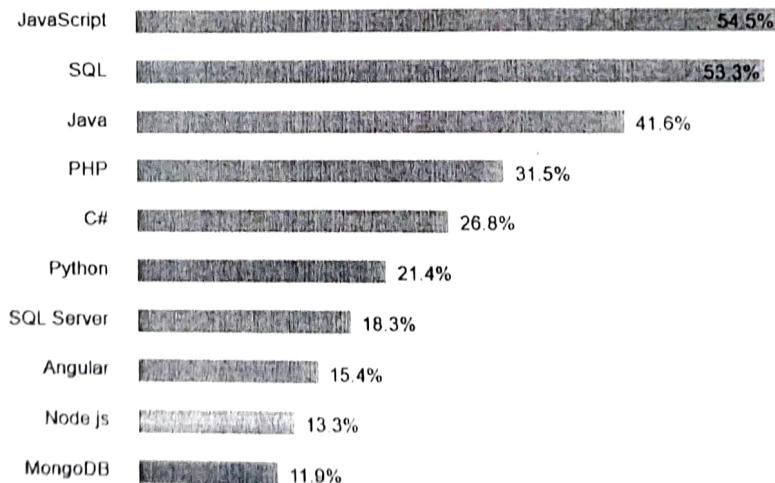
## Selenium with Java : Getting Started to Run Automated Tests

Selenium is the first thing that comes to mind when one is planning to automate the testing of web applications. Selenium is a beneficial tool because it is not only open source but also a portable software testing framework for web applications that support multiple languages like Java, C#, Ruby, Python. Choosing the right language depends on the application under test, the supporting community, available test automation frameworks, usability, elegance, and of course, seamless build integration.

Java is a popular programming language. As per StackOverflow, it is the third most popular back-end technology after JavaScript and SQL.

### Most Popular Technologies per Dev Type

Full-Stack    Front-End    **Back-End**    Mobile    Math & Data    Students



## Why Developers prefer Selenium With Java?

A good community of developers to create documentation and resolve issues has helped Java to become the most preferred language among the application developers. Thus, writing selenium test cases using Java has multiple benefits:

- Selenium supports Java. So, testers can leverage the active community of contributors and detailed documentation to write test cases
- Programs written in Java are faster than other popular languages like Python
- Java is more widely used in commercial applications as compared to other programming languages like Python and hence integrating Selenium tests it easier

Eventually choosing the right language varies by project, organization, and individuals driving it. An essential criterion is to know the language in-depth when dealing with Selenium.

*ca*

## Getting Started with Selenium using Java

To kickstart Selenium Webdriver with Java, one needs to bring the different components together, to start coding.

Selenium is compatible with multiple operating systems like Windows, Linux, Solaris, Macintosh. It also supports multiple browsers like Chrome, Safari, IE, Edge, Firefox. It is used to automate functional tests. Also, Selenium is easy to integrate with tools like [Jenkins](#), [Maven](#), and Docker to achieve a continuous testing approach. Tools like [TestNG](#) and JUnit further help in structuring the selenium tests for easy maintainability and generating reports.

In this section, we explain how to set up and run a simple test through Selenium with Java bindings.

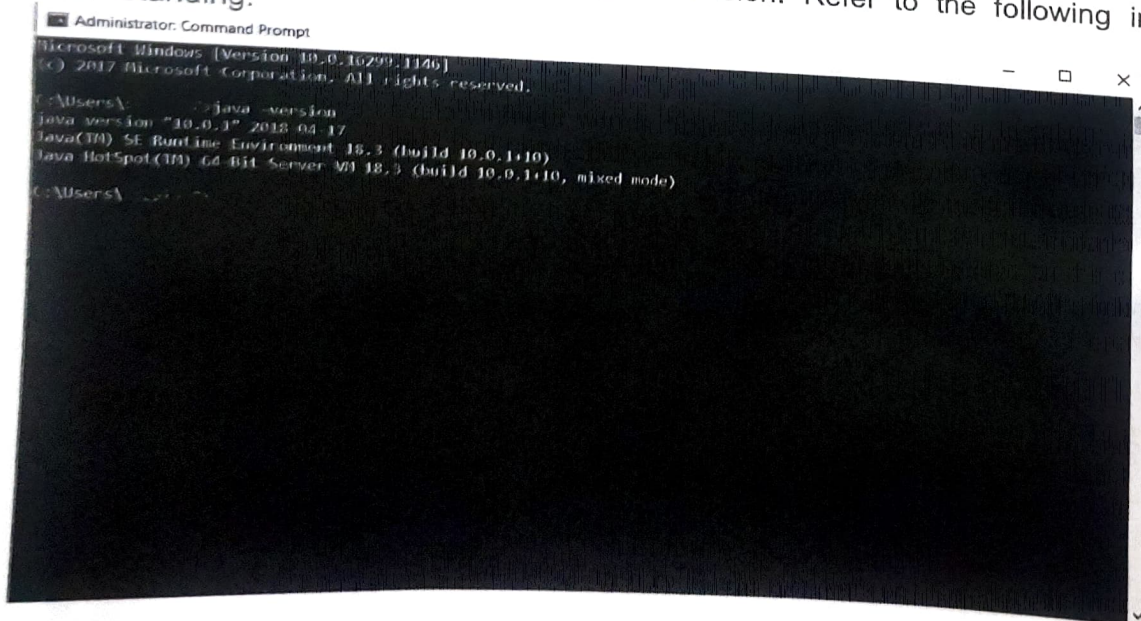
## Pre-requisites for Setup Configuration

The following components are required to get started with automation:

1. Java(JDK)
2. Eclipse
3. Selenium Client and WebDriver Language bindings
4. Configuring Selenium Webdriver with Eclipse
5. Creating and Running the first test

## Step #1 – Install Java

A Java development kit which includes the JRE (Java Runtime Environment) is required to write and run Java programs. JRE is a child of JDK, which comes along with JDK installation. Even for running applications that need the dependency of Java, one needs to install JDK. One such application is Eclipse IDE. [Download Java](#), install it and set the environment path. Once the path is set, one can verify the installation by typing `java -version` on the command prompt, which provides the details of the installed java version. Refer to the following image for better understanding.



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.16299.1146]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\A> java -version
java version "10.0.1" 2018-04-17
Java(TM) SE Runtime Environment 18.3 (build 10.0.1-10)
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.1-10, mixed mode)

C:\Users\A>
```



### Step #2 – Install Eclipse

Eclipse is a platform for Java developers to write their code and run it. [Download Eclipse](#) from their office page. Based on the OS, one can go with the required option. Once downloaded, extract the downloaded file. Once completed, one can see the eclipse .exe in the eclipse folder:

configuration	7/26/2019 6:05 PM	File folder	
dropins	3/13/2019 3:47 PM	File folder	
jre1.8.0_171	7/17/2019 11:47 A...	File folder	
plugins	3/13/2019 3:47 PM	File folder	
readme	3/13/2019 3:47 PM	File folder	
.eclipseproduct	12/22/2017 12:31 ...	ECLIPSEPRODUCT ...	1 KB
eclipse	3/30/2018 5:31 PM	Application	306 KB
eclipse	7/17/2019 12:41 PM	Configuration setti...	1 KB
eclipsec	3/30/2018 5:31 PM	Application	18 KB
javaw	5/8/2019 7:05 PM	Application	204 KB

### Step #3 – Selenium Client and WebDriver Language Bindings

[Selenium Webdriver](#) supports multiple languages, and each language has its client driver. As we are using Selenium with Java, we need to have 'Selenium Java Client Driver'. One can download the client driver from the [official Selenium website](#). One can see the multiple language client drivers provided by Selenium:

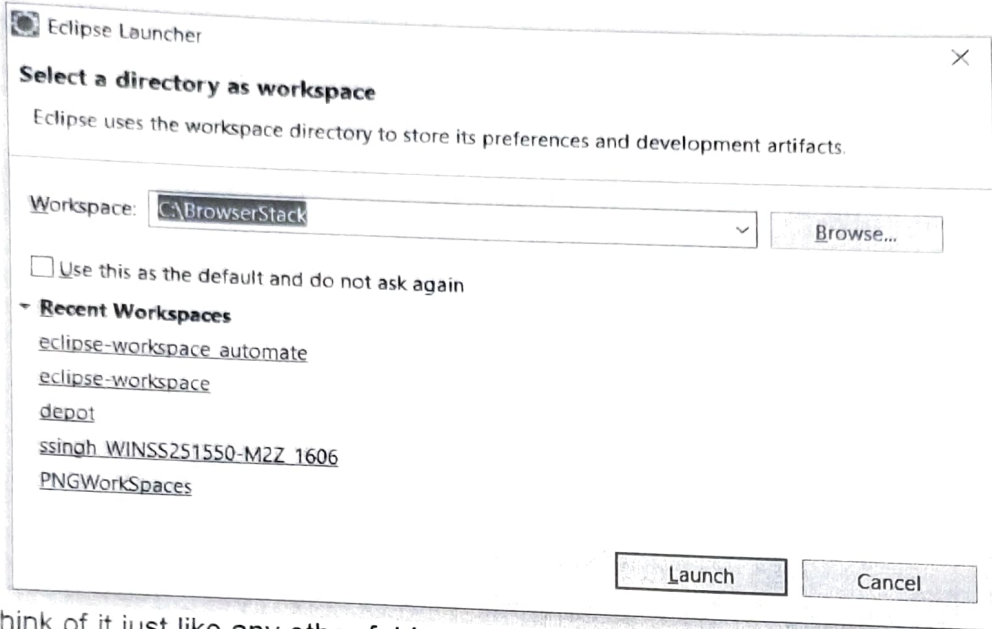
Language	Client Version	Release Date			
Java	3.141.59	2018-11-14	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">Javadoc</a>
C#	3.14.0	2018-08-02	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Ruby	3.14.0	2018-08-03	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Python	3.14.0	2018-08-02	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Javascript (Node)	4.0.0-alpha.1	2018-01-13	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>

Once downloaded, extract the contents of the downloaded file and then move to the next step, which is configuring Selenium Webdriver with Eclipse.

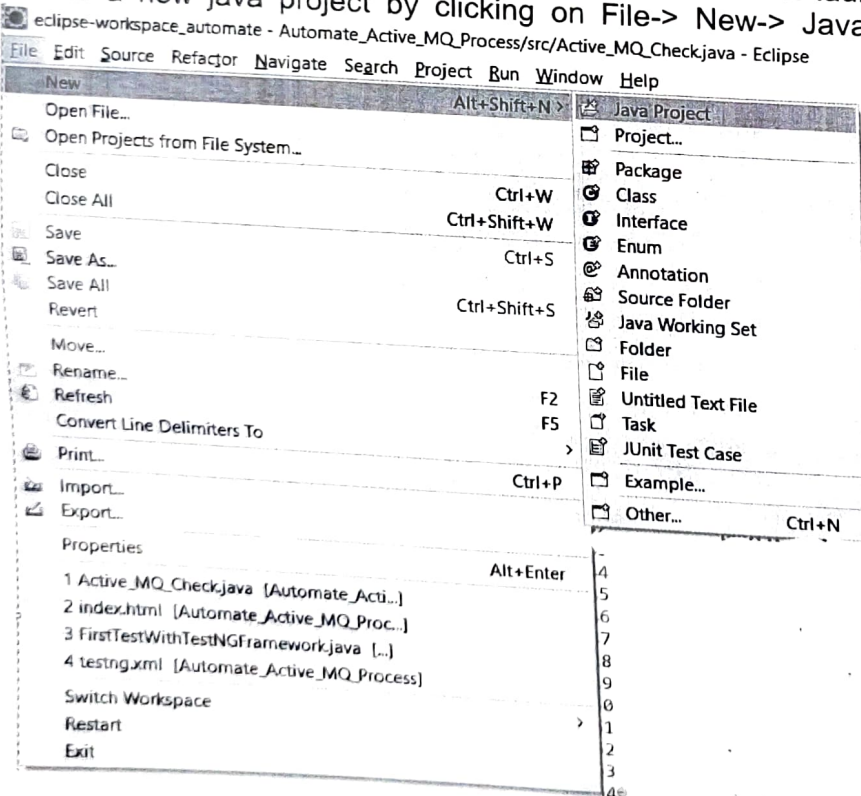
### Step #4 – Configuring Selenium WebDriver With Eclipse

This is a vital step of starting with Selenium. To configure Eclipse with Selenium Webdriver client,

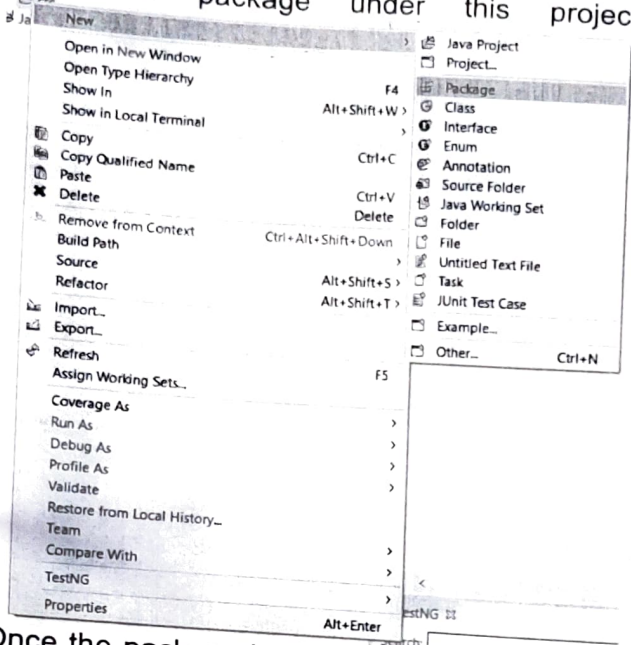
1. Double click on the eclipse.exe file to launch it
2. Create a workspace



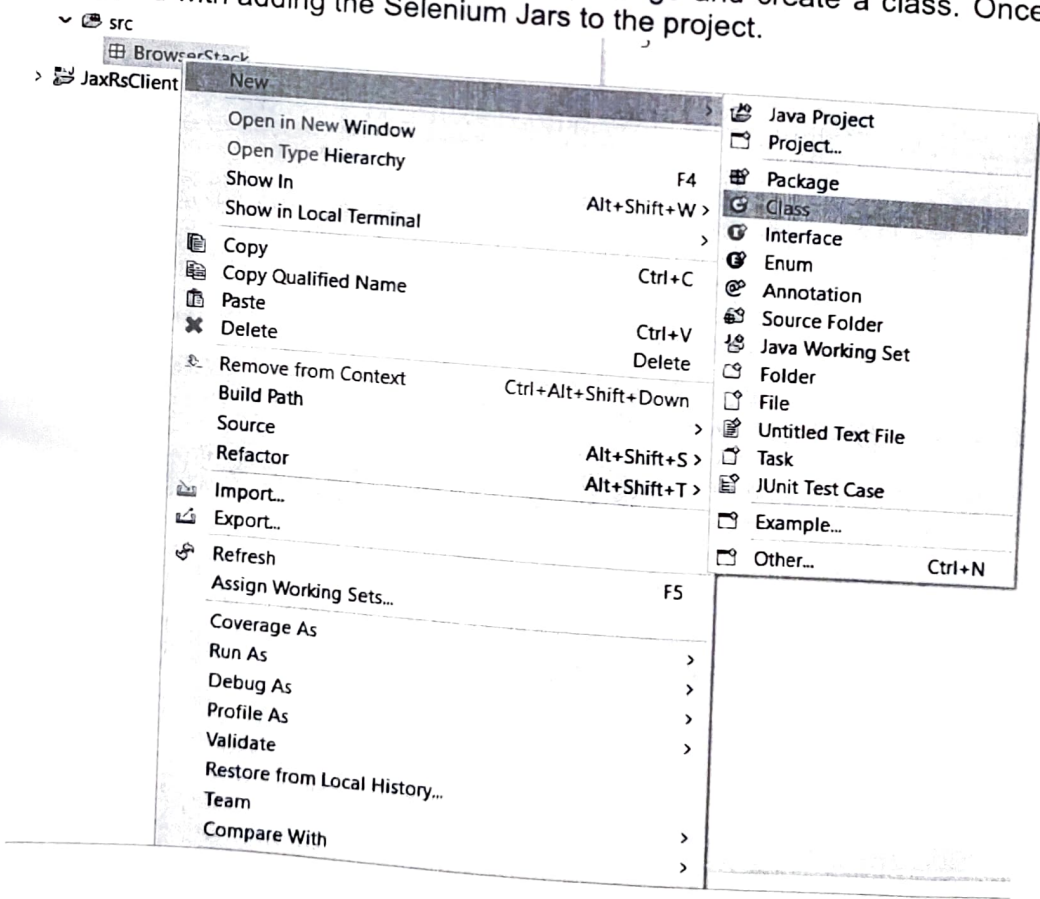
3. Think of it just like any other folder, which stores all the scripts in one place. One can choose to create as many workspaces as required. Click on Launch to launch the workspace.
4. Create a new java project by clicking on File-> New-> Java Project and name the project



5. Create a package under this project, by right-clicking on the 'src' folder



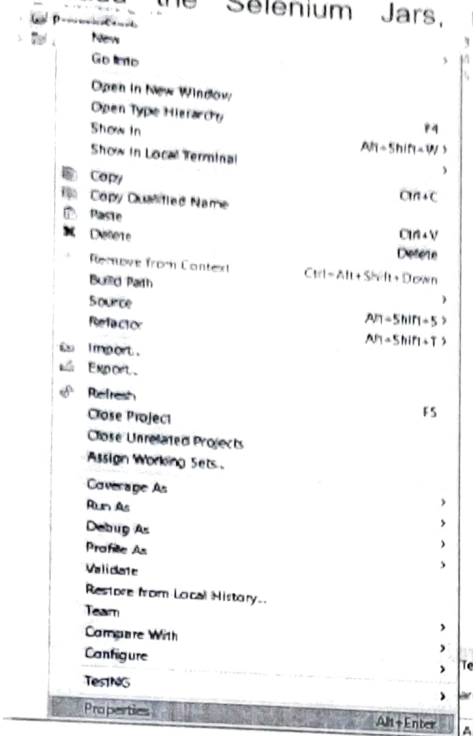
6. Once the package is created, right-click on the package and create a class. Once the class is created, go ahead with adding the Selenium Jars to the project.



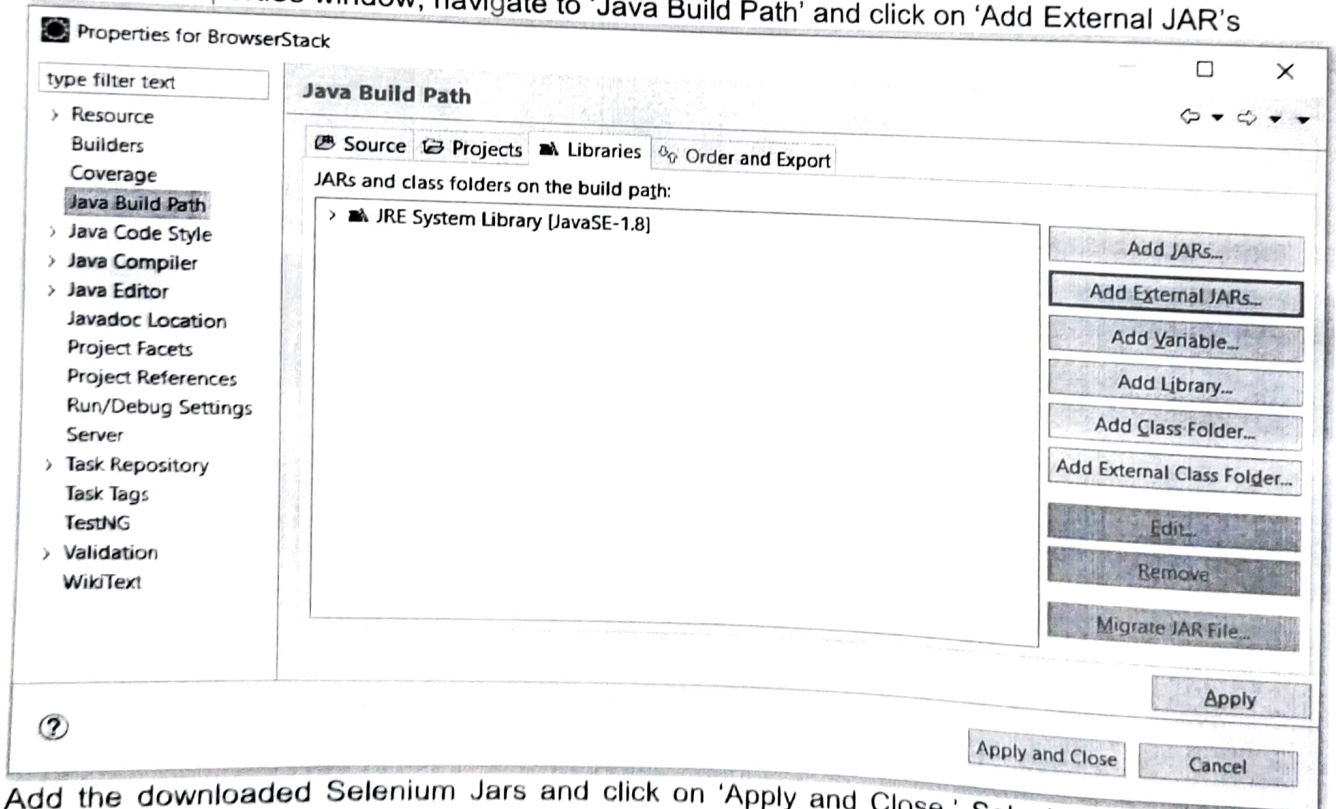
A



7. To add the Selenium Jars, right-click on the project folder and go to Properties:



From the Properties window, navigate to 'Java Build Path' and click on 'Add External JAR's



Add the downloaded Selenium Jars and click on 'Apply and Close.' Selenium with Eclipse is configured now. Now Eclipse is ready to execute the first script.

## Step #5 – Creating and Running the first test using Selenium and Java

As the first test, we will write a script to open 'google.com' on the Chrome browser. To use Chrome, it is mandatory to have the driver executable. To download the driver executable, visit the [Selenium website](#). In the third-party driver browser section, one can download the executable file for a specific browser.

Post downloading, below is the code snippet to run the first test:

```
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class FirstTestInSelenium {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        //setting the driver executable
        System.setProperty("webdriver.chrome.driver", ".\\Driver\\chromedriver.exe");

        //Initiating your chromedriver
        WebDriver driver=new ChromeDriver();

        //Applied wait time
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        //maximize window
        driver.manage().window().maximize();

        //open browser with desired URL
        driver.get("https://www.google.com");

        //closing the browser
        driver.close();

    }
}
```

In the code snippet above, we have used the *Selenium keyword driver.get("URL to open in browser")* to open URL in the desired browser. Other keywords like *driver.close* help to close the browser window, as a cleanup part. So, this was a quick start guide to start using Selenium with Java. One should keep in mind the best practices to be incorporated while writing Selenium tests.

## Selenium with Java: Best Practices

Some of the important aspects to remember while writing Selenium tests with Java are:

- **Using the right locator** – Selecting locators are the building blocks of a selenium script, and using the right one is critical. If incorrect locators are used, they tend to make the script flaky and unreliable. Using 'ID' and 'Name' locators are easy. They also provide faster execution and are more reliable as compared to CSS and XPath
- **Making your way into the test-driven script** – When we talk about testing, it is about testing the software on multiple permutations and combination of data. The same should be incorporated in the Selenium tests. Multiple data points should drive all Selenium tests, and using a data-driven framework helps achieve this

DA

- **Using the right wait and avoiding using 'Thread.sleep'** – For WebElements or a page to load, it is essential to give a specific halt time to the script and avoid the failure of the script. Selenium provides certain waits like '*Implicit*' or '*Explicit*' to achieve this. Both these waits halt the execution of the script until it finds the element. The moment it finds the element, it continues the execution of the script. '*Thread.sleep*' on the other hand, stops the execution for the defined period even when it finds the element in the defined interval. This increases the execution time of the script
- **Don't make scripts specific to a browser or driver** – Cross browser testing plays a vital role in testing. One may expect the scripts to run on multiple browsers or a specific browser depending on business needs. Selenium frameworks like TestNG provide annotations such as `@parameters` and JUnit provides annotations like `@RunWith`, which helps in running tests on multiple browsers and corresponding drivers
- **Validating tests using assertions** – The key to writing a good test is validating the tests. Just like when one writes a test case and mentions the actual and expected results, one needs to assert the tests in Selenium with the help of assertions provided in frameworks like TestNG and JUnit. If the assertions are not used, the testing process is incomplete, as it is not validating the correctness of the test build
- **Taking screenshots and Reporting** – As a QA tester, to provide proof of testing for failures, it is essential to have supportive screenshots. The same stands for Automated Selenium testing. In case a test fails, it is vital to have corresponding screenshots. This helps explain the bug to the developer, who can debug it instantly. Similarly, from a reporting perspective, to provide insight to the stakeholders, it is valuable to share reports with them, to establish the stability of the product. For this, Selenium provides a default reporting system with frameworks like TestNG and provides further customizations to them using listeners