**How to Download & Install Selenium WebDriver**

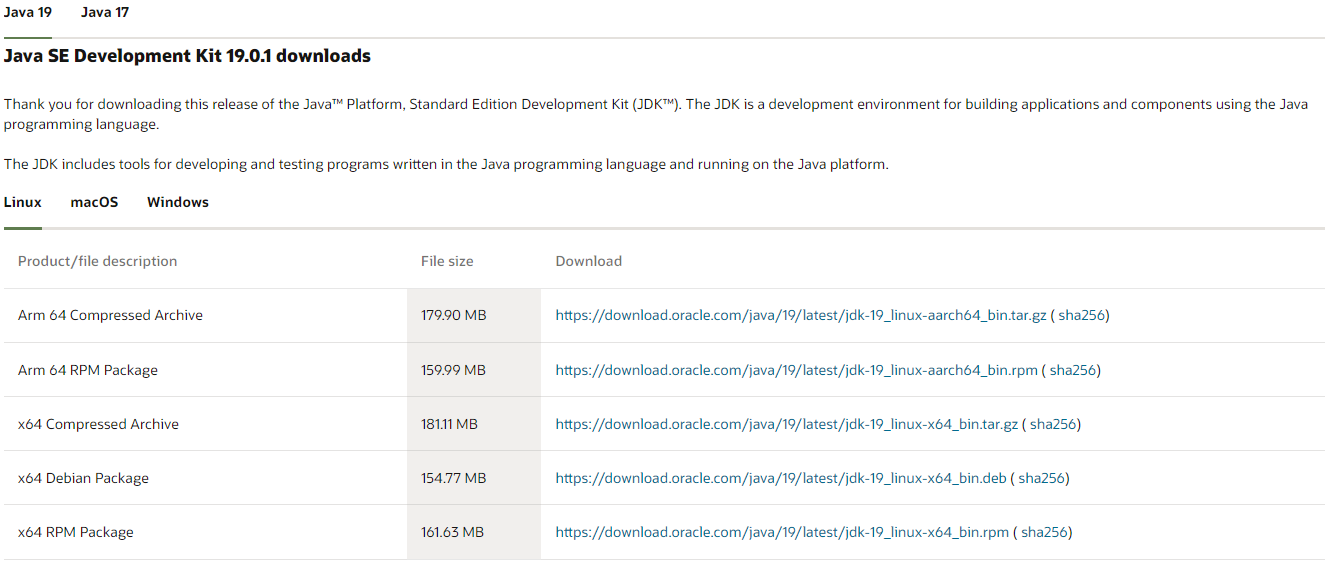
## Selenium WebDriver Installation

Selenium installation is a 3-step process:

**Step 1:** Install Java SDK  
**Step 2:** Install Eclipse  
**Step 3:** Install Selenium Webdriver Files  
**Step 4:** Configure Eclipse IDE with WebDriver

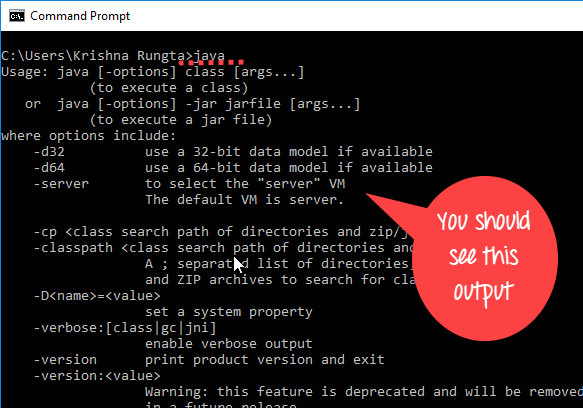
## Step 1 – Install Java Software Development Kit (JDK)

Download and install the **Java Software Development Kit (JDK)** [here](https://www.oracle.com/java/technologies/downloads/).

[](https://www.guru99.com/images/JDK.png)

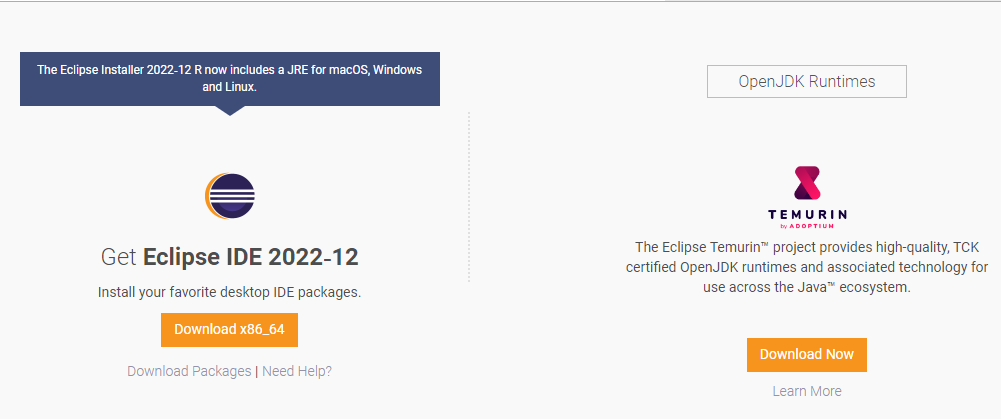
This JDK version comes bundled with Java Runtime Environment (JRE), so you do not need to download and install the JRE separately.

Once installation is complete, open command prompt and type “java”. If you see the following screen you are good to move to the next step.

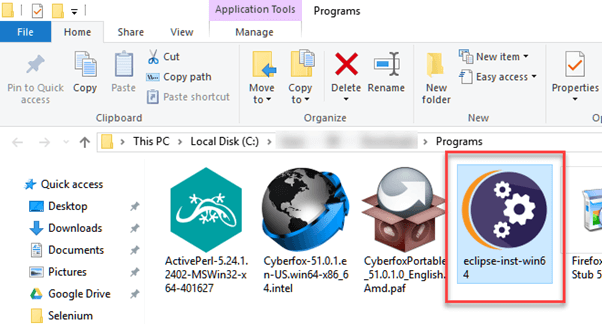
[](https://www.guru99.com/images/2-2017/022017_1205_Guidetoinst_24_7_1.png)

## Step 2 – Install Eclipse IDE

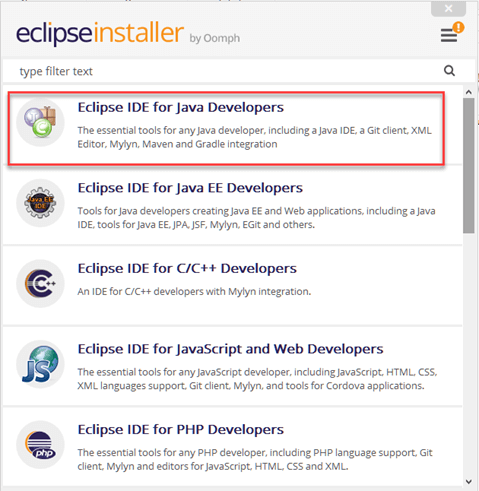
Download the latest version of **“Eclipse IDE for Java Developers”** [here](https://www.eclipse.org/downloads/). Be sure to choose correctly between Windows 32 Bit and 64 Bit versions.

[](https://www.guru99.com/images/eclipse.png)

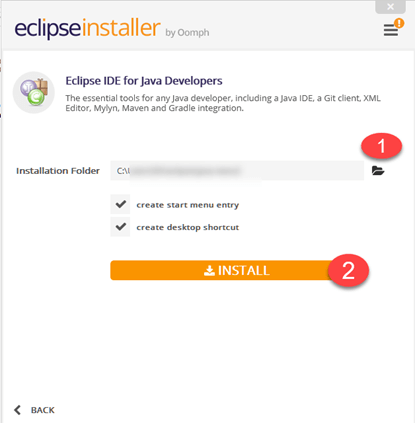
You should be able to download an exe file named “eclipse-inst-win64” for Setup.

[](https://www.guru99.com/images/2-2017/022017_1205_Guidetoinst4.png)

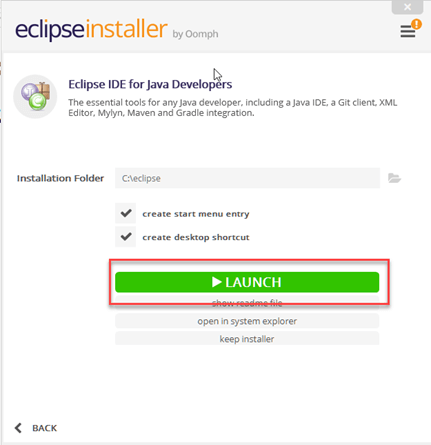
Double-click on a file to Install the Eclipse. A new window will open. Click Eclipse IDE for Java Developers.

[](https://www.guru99.com/images/2-2017/022017_1205_Guidetoinst5.png)

After that, a new window will open which click button marked 1 and change path to “C:\eclipse”. Post that Click on the Install button marked 2

[](https://www.guru99.com/images/2-2017/022017_1205_Guidetoinst6.png)

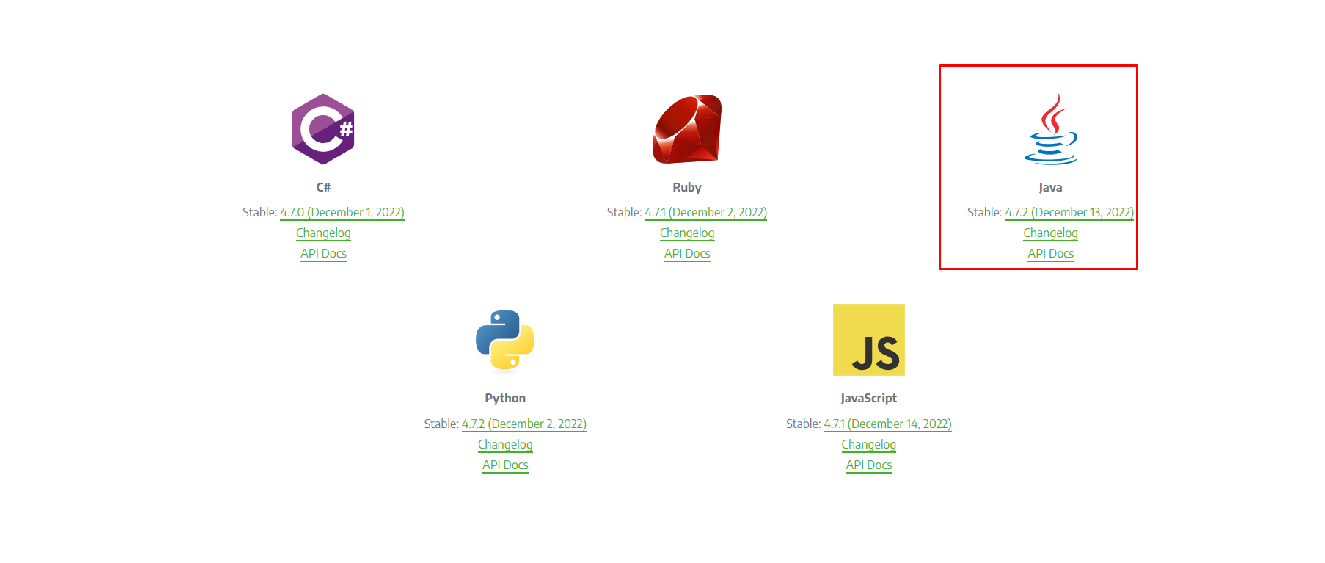
After successful completion of the installation procedure, a window will appear. On that window click on Launch.

[](https://www.guru99.com/images/2-2017/022017_1205_Guidetoinst7.png)

This will start eclipse neon IDE for you.

## Step 3 – Selenium WebDriver Installation

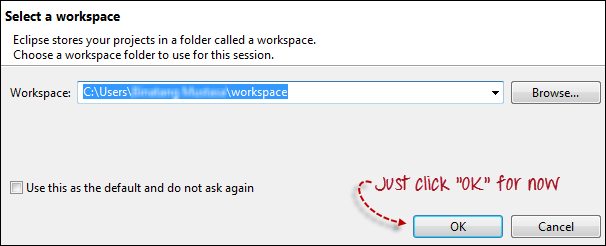
You can download **Selenium Webdriver for Java Client Driver** [here](https://selenium.dev/downloads/). You will find client drivers for other languages there, but only choose the one for Java.

[](https://www.guru99.com/images/java_driver.png)

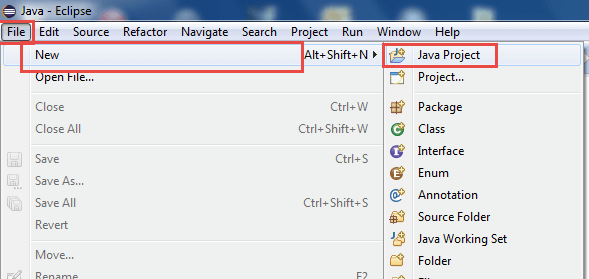
This download comes as a ZIP file named “selenium-3.14.0.zip”. For simplicity of Selenium installation on Windows 10, extract the contents of this ZIP file on your C drive so that you would have the directory “C:\selenium-3.14.0\”. This directory contains all the JAR files that we would later import on Eclipse for Selenium setup.

## Step 4 – Configure Eclipse IDE with WebDriver

1. Launch the “eclipse.exe” file inside the “eclipse” folder that we extracted in step 2. If you followed step 2 correctly, the executable should be located on C:\eclipse\eclipse.exe.
2. When asked to select for a workspace, just accept the default location.

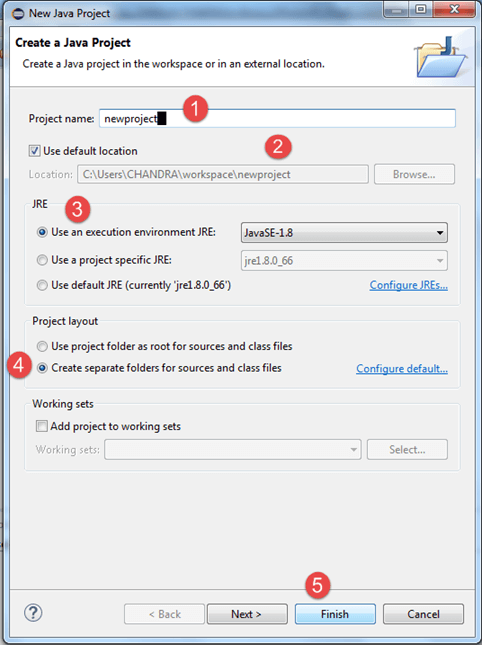
[](https://www.guru99.com/images/configure_eclipse_1b.png)

Create a new project through File > New > Java Project. Name the project as “newproject”.

[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns1.png)

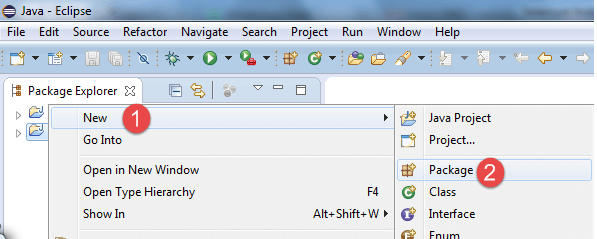
A new pop-up window will open. Enter details as follow

1. Project Name
2. Location to save a project
3. Select an execution JRE
4. Select the layout project option
5. Click on the Finish button

[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns2.png)

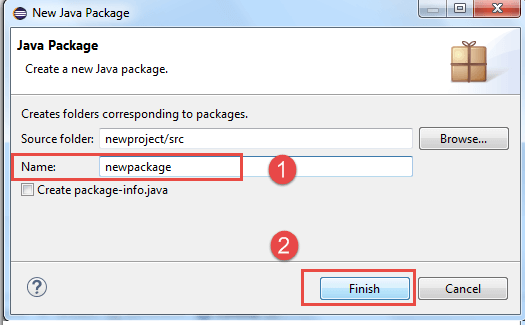
4. In this step,

1. Right-click on the newly created project and
2. Select New > Package, and name that package as “newpackage”.

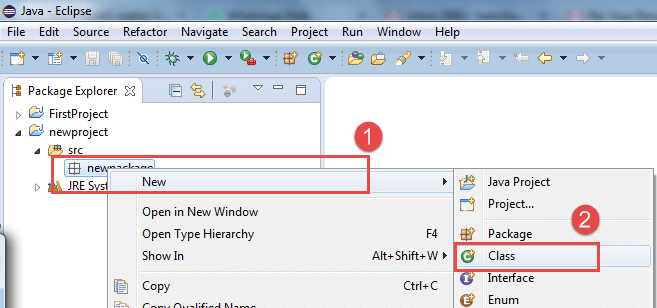
[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns3.png)

A pop-up window will open to name the package,

1. Enter the name of the package
2. Click on the Finish button

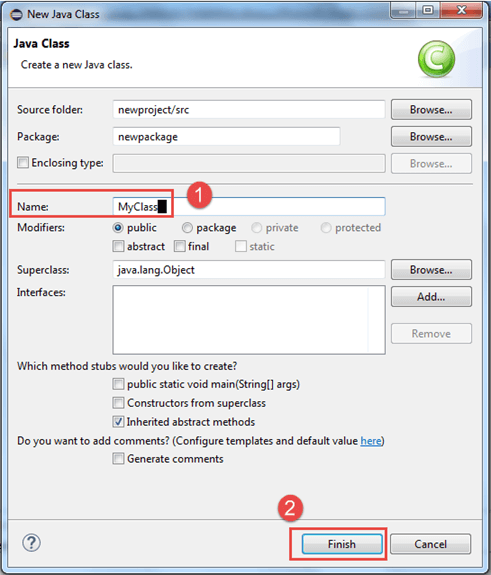
[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns4.png)

5. Create a new Java class under newpackage by right-clicking on it and then selecting- New > Class, and then name it as “MyClass”. Your Eclipse IDE should look like the image below.

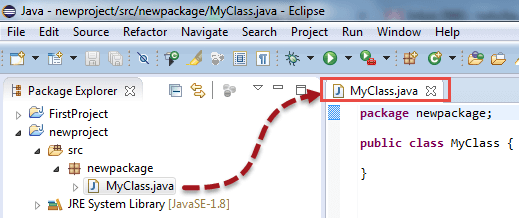
[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns5.png)

When you click on Class, a pop-up window will open, enter details as

1. Name of the class
2. Click on the Finish button

[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns6.png)

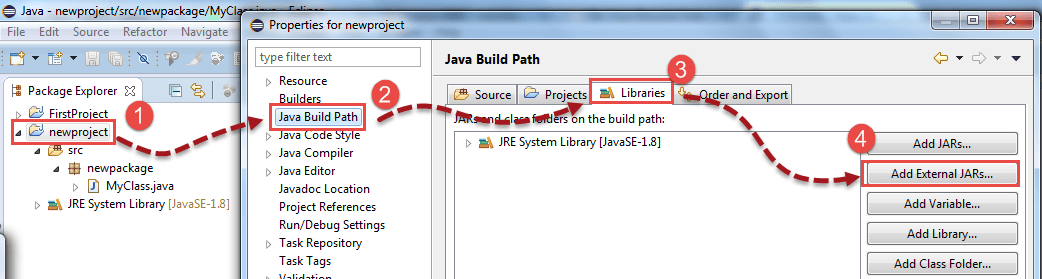
This is how it looks like after creating class.

[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns7.png)

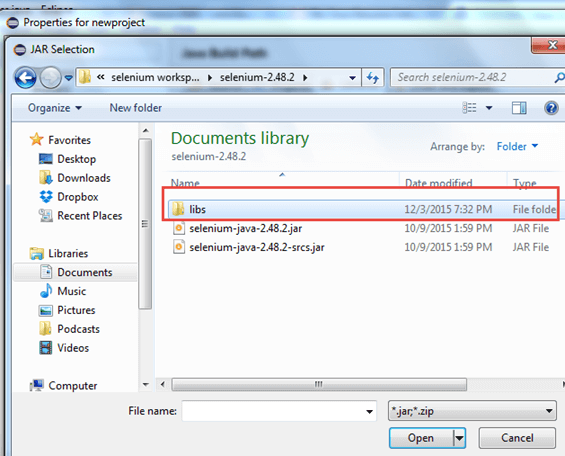
Now selenium WebDriver’s into Java Build Path

In this step,

1. Right-click on “newproject” and select **Properties**.
2. On the Properties dialog, click on “Java Build Path”.
3. Click on the **Libraries** tab, and then
4. Click on “Add External JARs..”

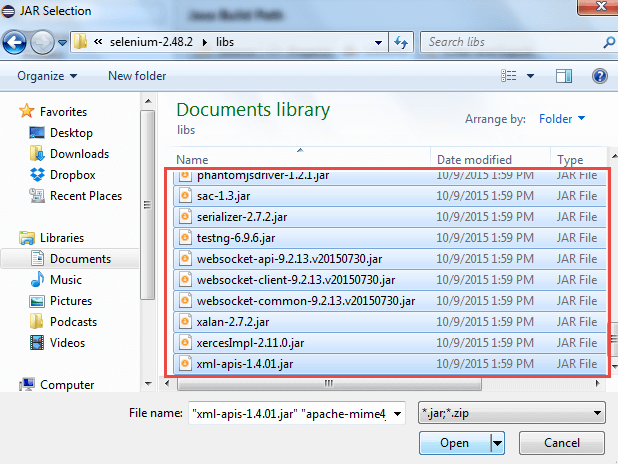
[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns8.png)

When you click on “Add External JARs..” It will open a pop-up window. Select the JAR files you want to add.

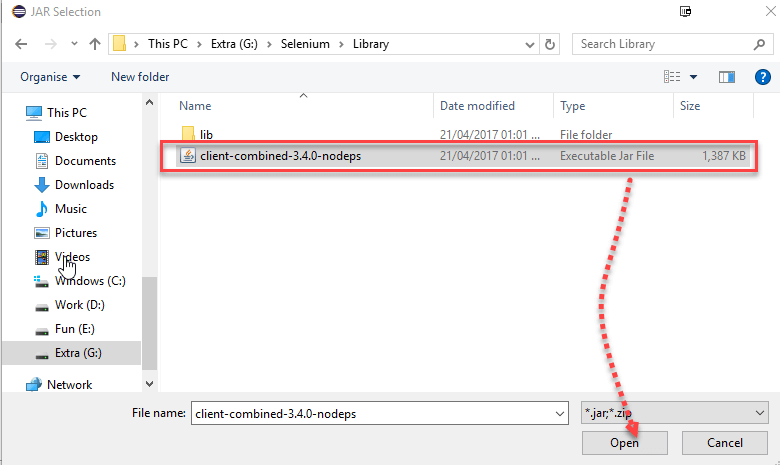
[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns9.png)

After selecting jar files, click on OK button.

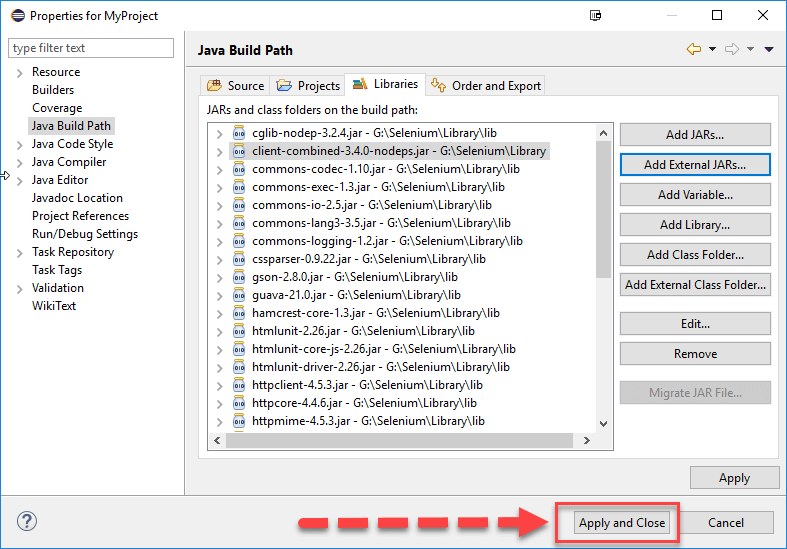
Select all files inside the lib folder.

[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns10.png)

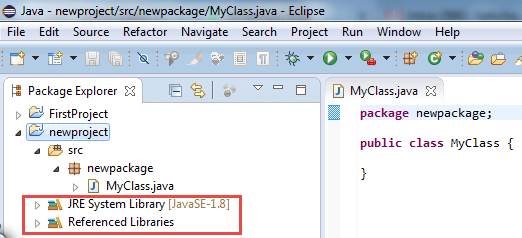
Select files outside lib folder

[](https://www.guru99.com/images/2-2017/022017_1205_Guidetoinst_24_7_2.png)

Once done, click “Apply and Close” button

[](https://www.guru99.com/images/2-2017/022017_1205_Guidetoinst_24_7_3.png)

6. Add all the JAR files inside and outside the “libs” folder. Your Properties dialog should now look similar to the image below.

[](https://www.guru99.com/images/cassandra/021316_1152_SeleniumIns11.png)

7. Finally, click OK and we are done importing Selenium libraries into our project.

# Selenium WebDriver- First Test Case

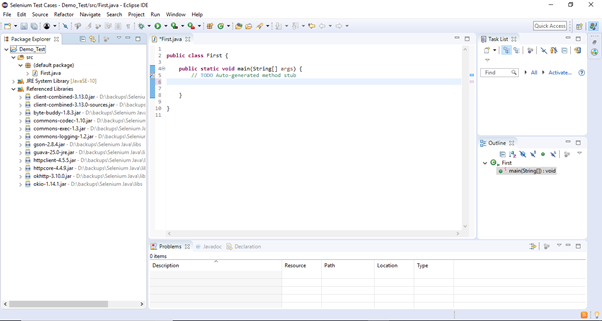
In this section, you will learn how to create your First Selenium Automation Test Script.

Under this test, we will automate the following scenarios:

* Invoke Google Chrome browser.
* Open URL: [www.google.com](https://www.google.com/)
* Click on the Google Search text box.
* Type the value "javatpoint tutorials"
* Click on the Search button.

We will create our test case step by step to give you a complete understanding of each component in detail.

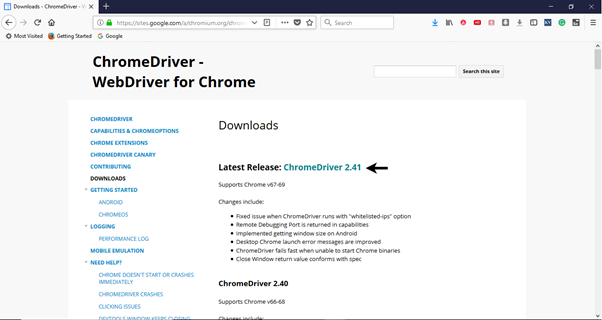
**Step1**. Launch Eclipse IDE and open project "Demo\_Test" which we have created in the previous section (Configure Selenium WebDriver) of this Tutorial. We will write our first Selenium test script in the "First.class" file under the "Demo\_Test" test suite.



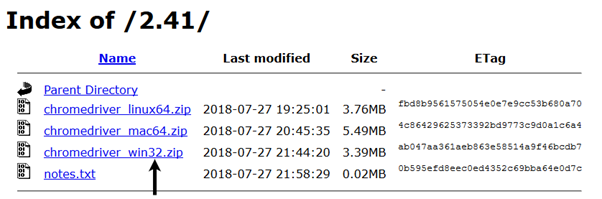
#### Note: To invoke a browser in Selenium, we have to download an executable file specific to that browser. For example, Chrome browser implements the WebDriver protocol using an executable called ChromeDriver.exe. These executable files start a server on your system which in turn is responsible for running your test scripts in Selenium.

Step2. Open URL: <https://sites.google.com/a/chromium.org/chromedriver/downloads> in your browser.

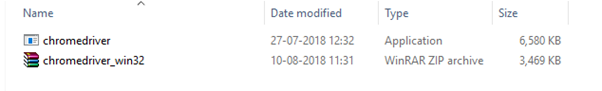
**Step3.** Click on the "ChromeDriver 2.41" link. It will redirect you to the directory of ChromeDriver executables files. Download as per the operating system you are currently on.



For windows, click on the "chromedriver\_win32.zip" download.



The downloaded file would be in zipped format. Unpack the contents in a convenient directory.



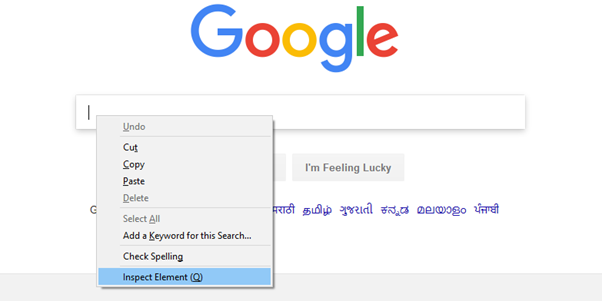
#### Note: Selenium developers have defined properties for each browser that needs the location of the respective executable files to be parsed in order to invoke a browser. For example, the property defined for Chrome browser - webdriver.chrome.driver, needs the path of its executable file - D:\ChromeDriver\chromedriver.exe in order to launch chrome browser.

Selenium WebDriver First Test Case

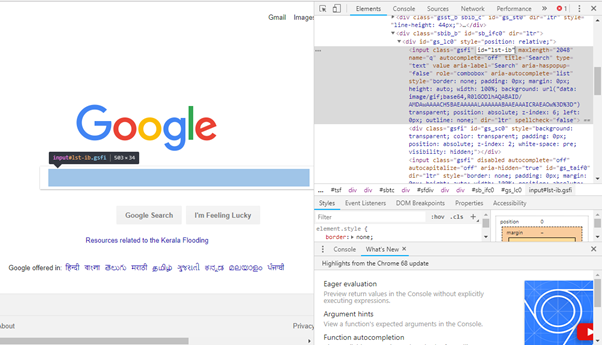
**Step4**. We would need a unique identification for the web elements like Google Search text box and Search button in order to automate them through our test script. These unique identifications are configured along with some Commands/Syntax to form Locators. Locators help us to locate and identify a particular web element in context of a web application.

The method for finding a unique identification element involves inspection of HTML codes.

* Open URL: [https://www.google.com](https://www.google.com/) in your Chrome browser.
* Right click on the Google search text box and select Inspect Element.



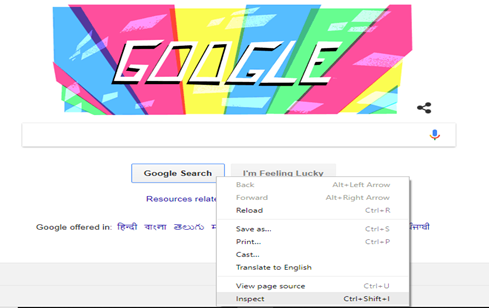
* It will launch a window containing all the specific codes involved in the development of the test box.



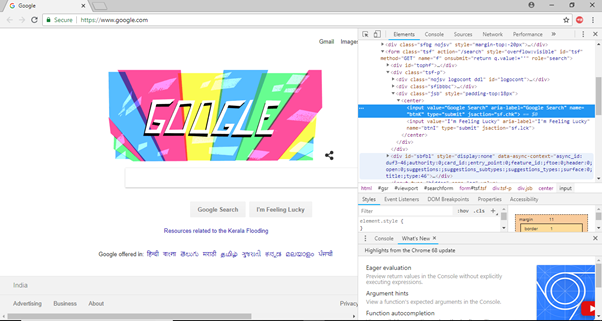
* Pick the value of id element i.e. "lst-ib".

Selenium WebDriver First Test Case

* Given below is the Java syntax for locating elements through "id" in Selenium WebDriver.
  1. driver.findElement(By.id (<element ID>))
* Here is the complete code for locating Google Search text box in our test script.
  1. driver.findElement(By.id ("lst-ib"))
* Now, right click on the Google Search button and select Inspect Element.



* It will launch a window containing all the specific codes involved in the development of the Google Search button.



* Pick the value of **name** element i.e. "btnK".

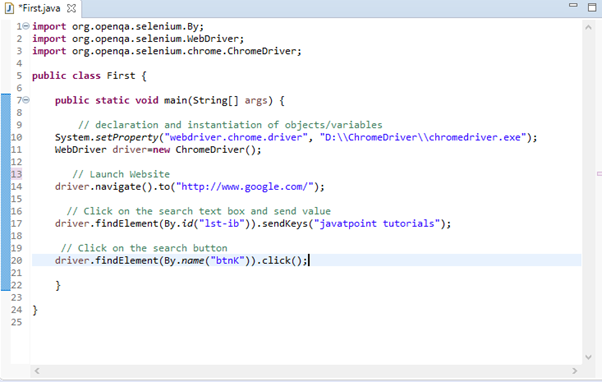
Selenium WebDriver First Test Case

* Given below is the Java syntax for locating elements through "name" in Selenium WebDriver.
  1. driver.findElement(By.name (<element name>))
* Here is the complete code for locating Google Search button in our test script.
  1. driver.findElement(By.name ("btnK"))

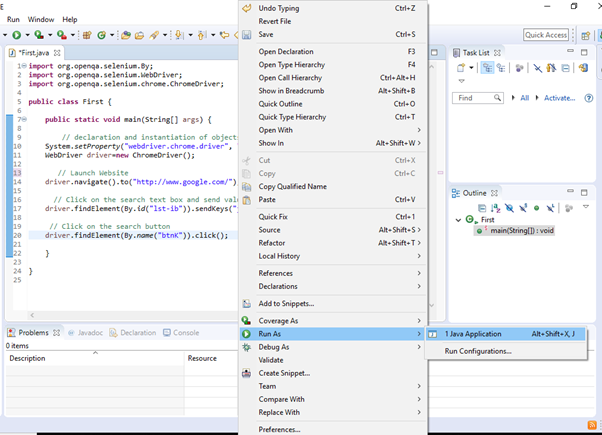
**Step5**. Now it is time to code. We have embedded comments for each block of code to explain the steps clearly.

1. **import** org.openqa.selenium.By;
2. **import** org.openqa.selenium.WebDriver;
3. **import** org.openqa.selenium.chrome.ChromeDriver;
5. **public** **class** First {
7. **public** **static** **void** main(String[] args) {
9. // declaration and instantiation of objects/variables
10. System.setProperty("webdriver.chrome.driver", "D:\\ChromeDriver\\chromedriver.exe");
11. WebDriver driver=**new** ChromeDriver();
13. // Launch website
14. driver.navigate().to("http://www.google.com/");
16. // Click on the search text box and send value
17. driver.findElement(By.id("lst-ib")).sendKeys("javatpoint tutorials");
19. // Click on the search button
20. driver.findElement(By.name("btnK")).click();
22. }
24. }

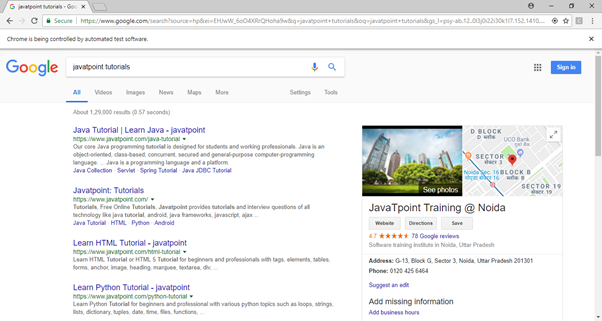
The Eclipse code window will look like this:



**Step6**. Right click on the Eclipse code and select **Run As > Java Application**.



**Step7**. The output of above test script would be displayed in Google Chrome browser.



## Explanation of the Code

### Import Packages/Statements

In java, import statements are used to import the classes present in another packages. In simple words, import keyword is used to import built-in and user-defined packages into your java source file.

1. **org.openqa.selenium.WebDriver** - References the WebDriver interface which is required to instantiate a new web browser.
2. **org.openqa.selenium.chrome.ChromeDriver** - References the ChromeDriver class that is required to instantiate a Chrome-specific driver onto the browser instantiated by the WebDriver class.

### Instantiating objects and variables

A driver object is instantiated through:

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

### Launch Website

To launch a new website, we use navigate().to() method in WebDriver.

1. driver.navigate().to("http://www.google.com/");

### Click on an element

In WebDriver, user interactions are performed through the use of Locators which we would discuss in later sessions of this tutorial. For now, following instance of code is used to locate and parse values in a specific web element.

1. driver.findElement(By.id("lst-ib")).sendKeys("javatpoint tutorials");