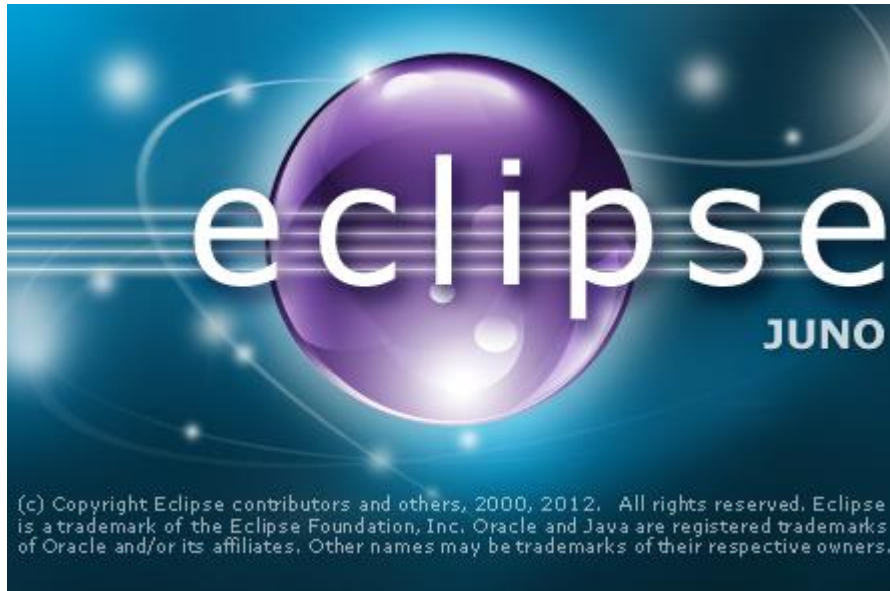


## How To Setup a Selenium WebDriver Project in Eclipse ?

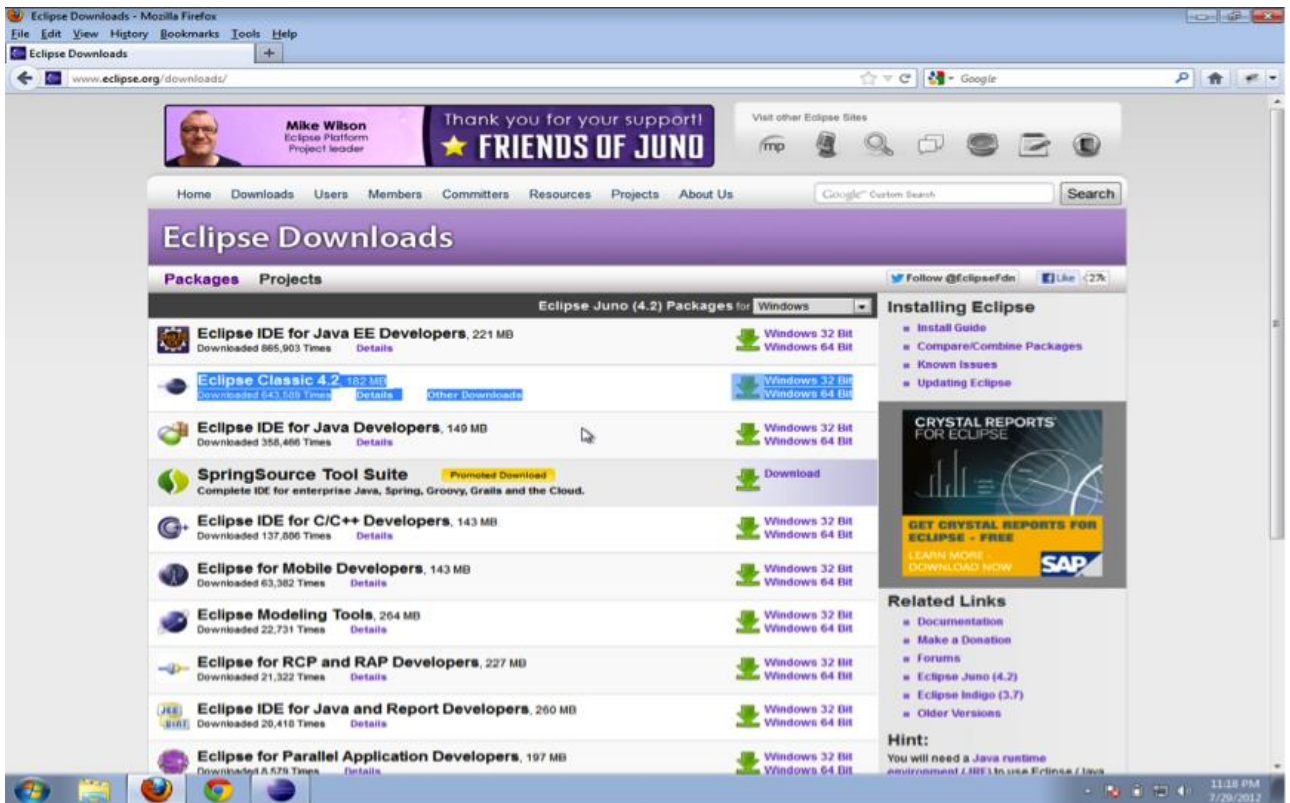
Creating a Simple Selenium – Java project in Eclipse



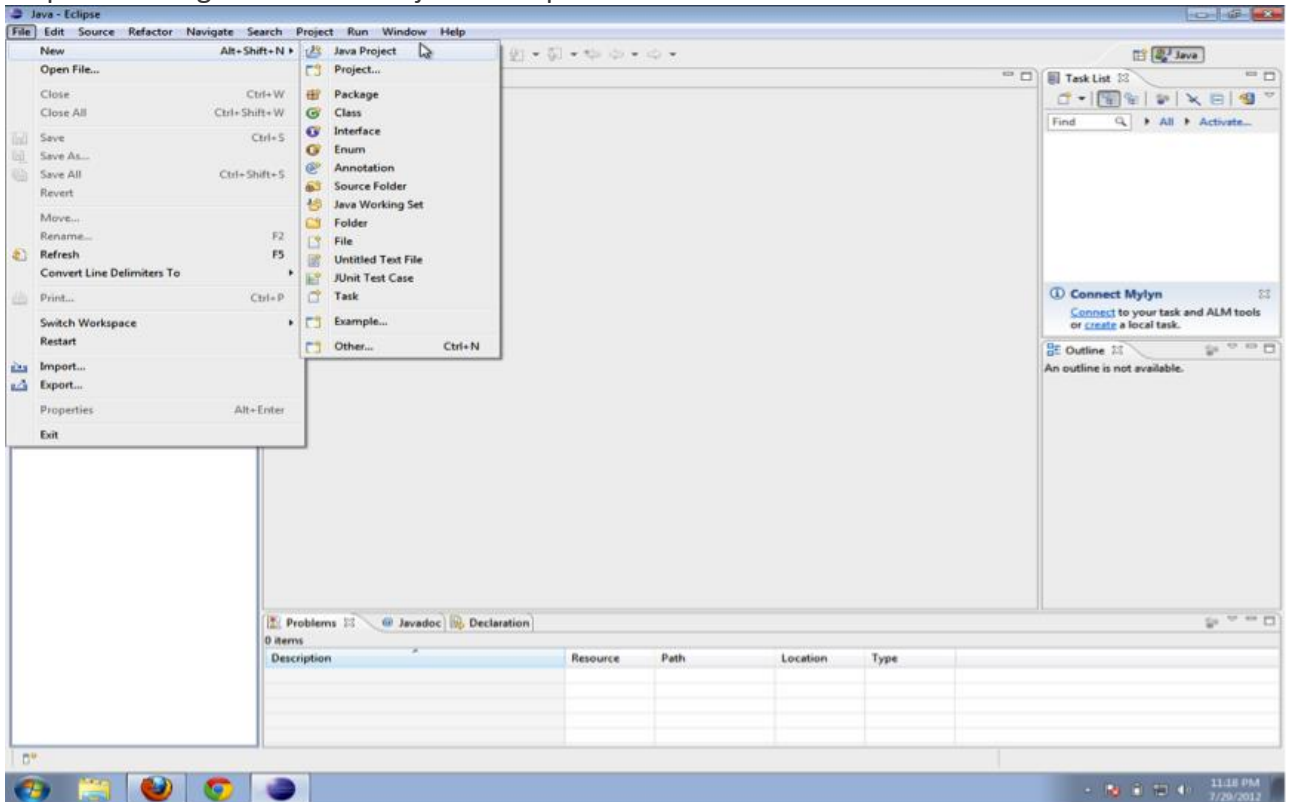
This post will help the selenium users-Beginners to setup the selenium project and execute a simple WebDriver script

Steps To Create a Selenium Project in Eclipse

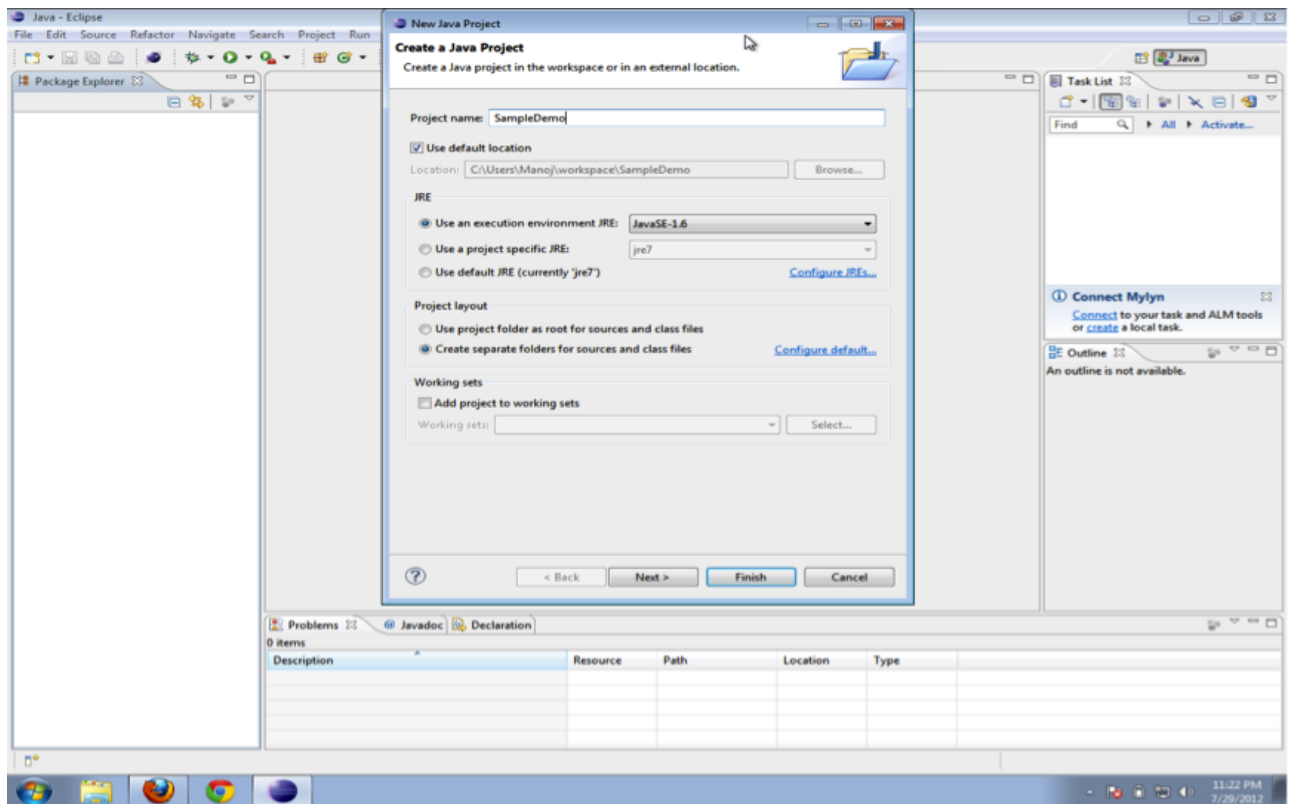
Step 1: Download Eclipse



## Step 2: Creating a New Java Project in Eclipse IDE

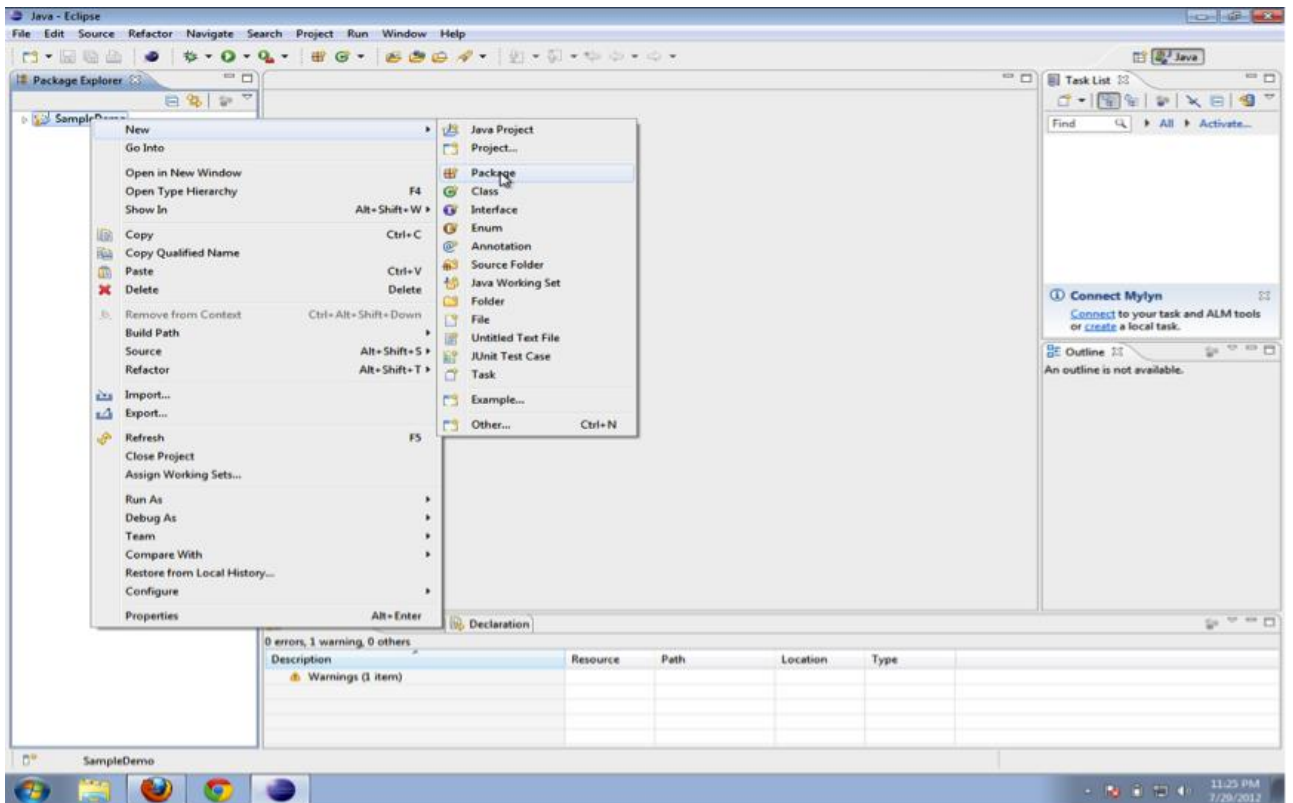


Step 3: Enter a Project Name E.g:SampleDemo

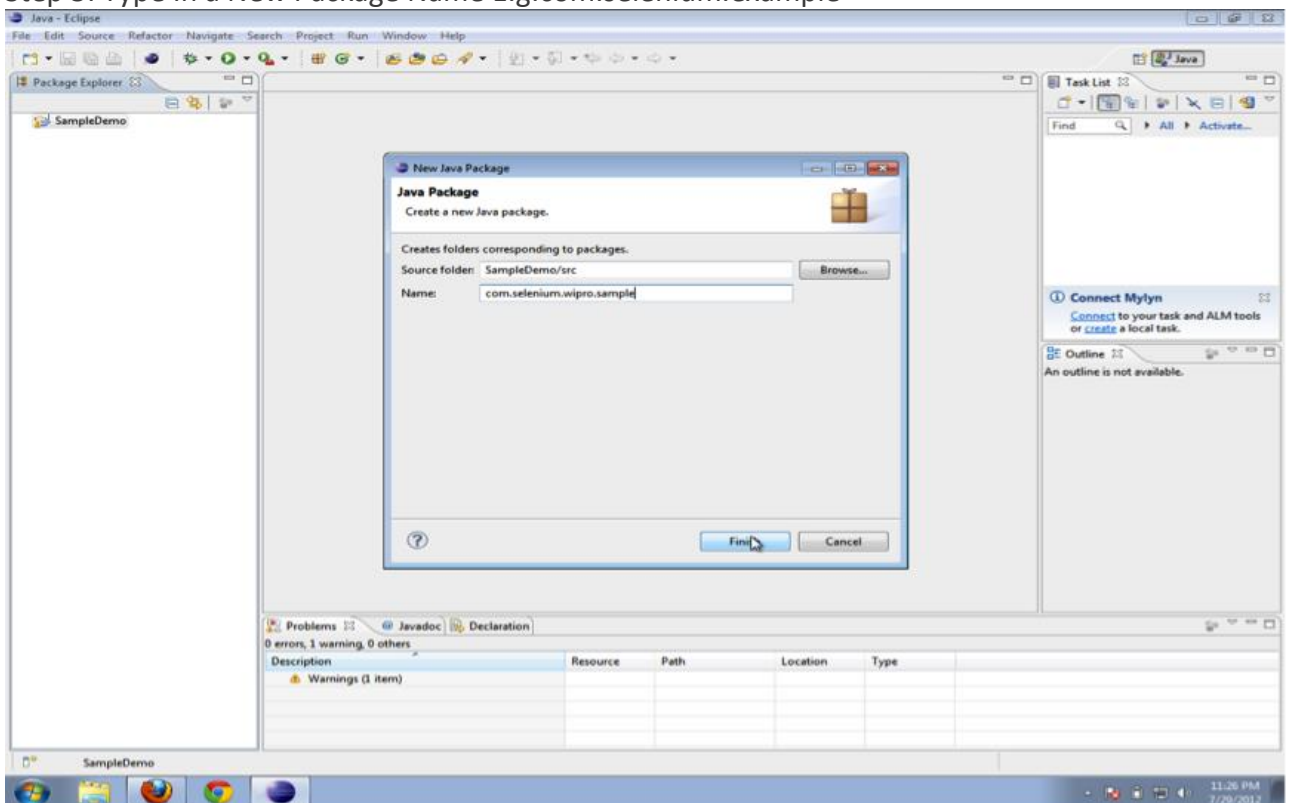


Step 4: Creating a New Package

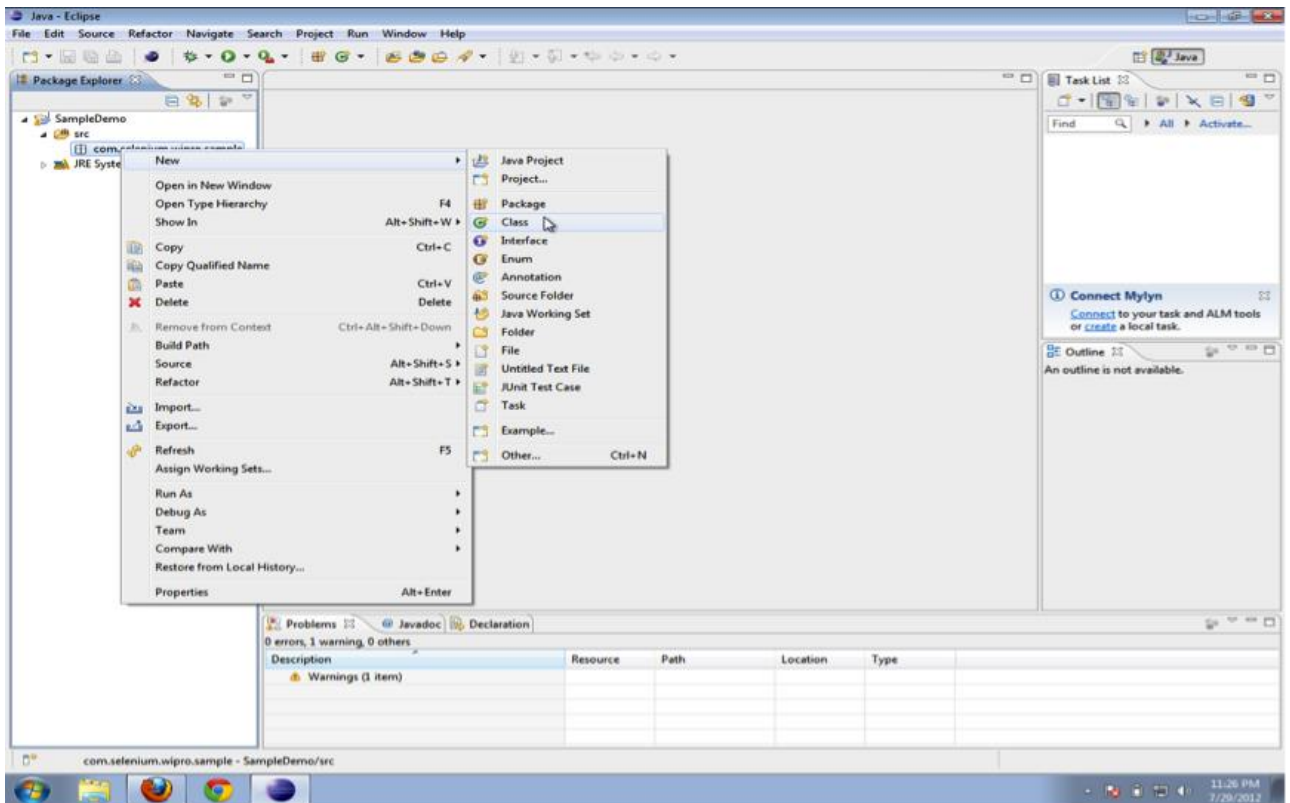
Right Click on the New Java Project created → New → Package



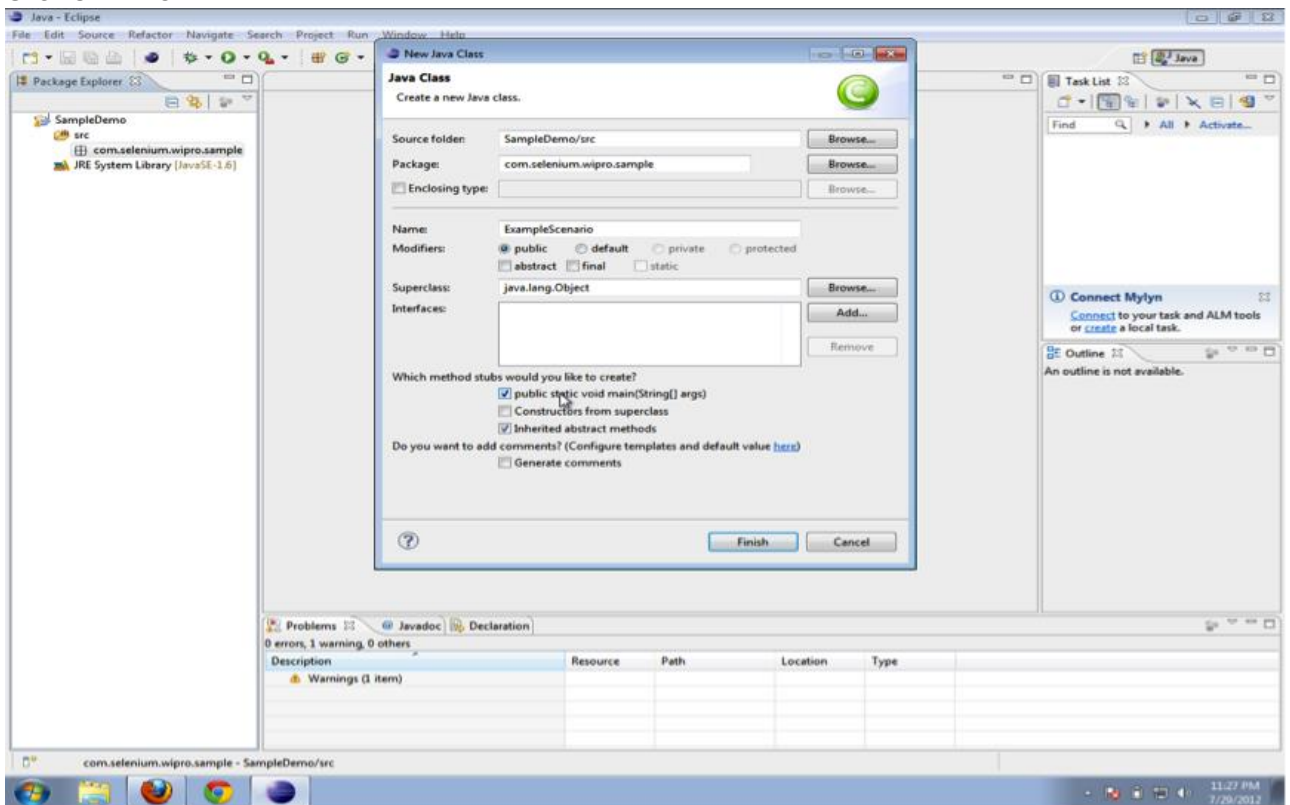
Step 5: Type In a New Package Name E.g:com.selenium.example



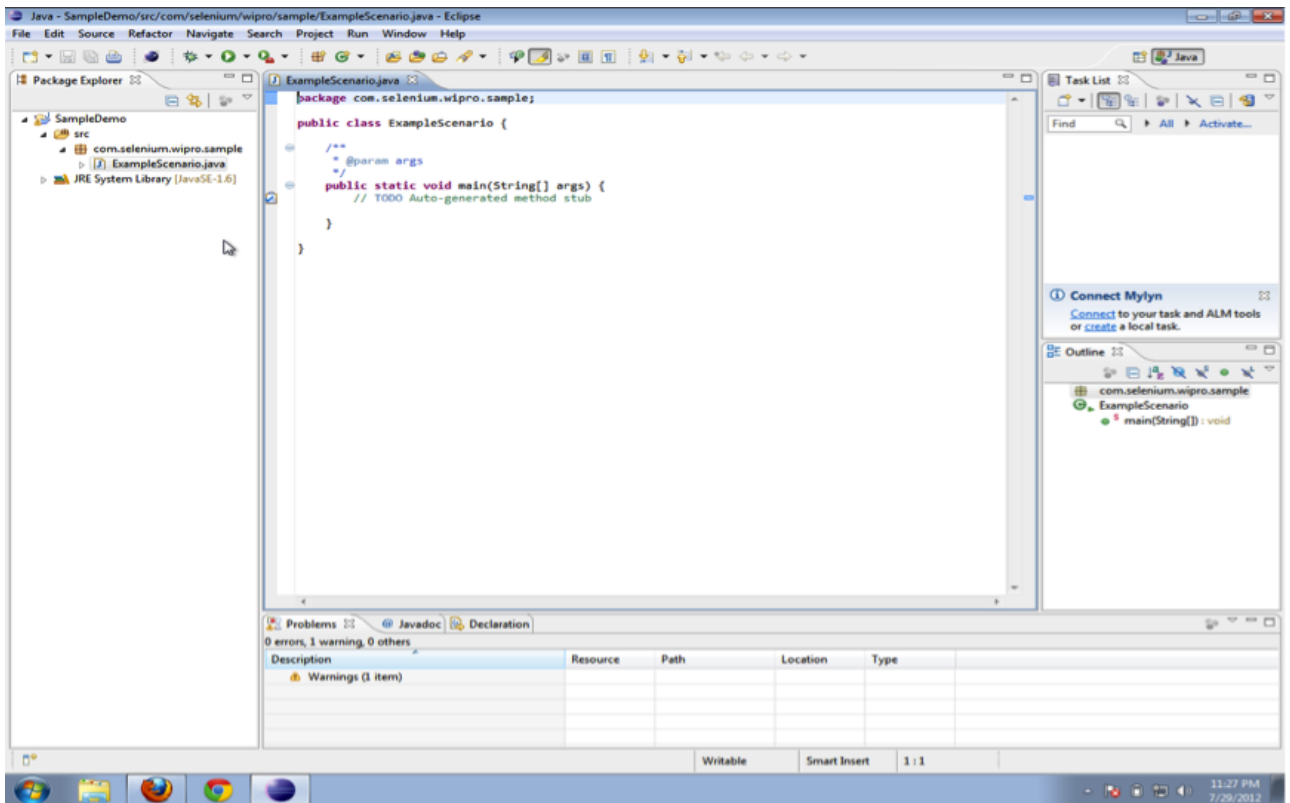
Step 6: Creating a New Java Class file.  
Right Click on the Newly created Package → New → Class



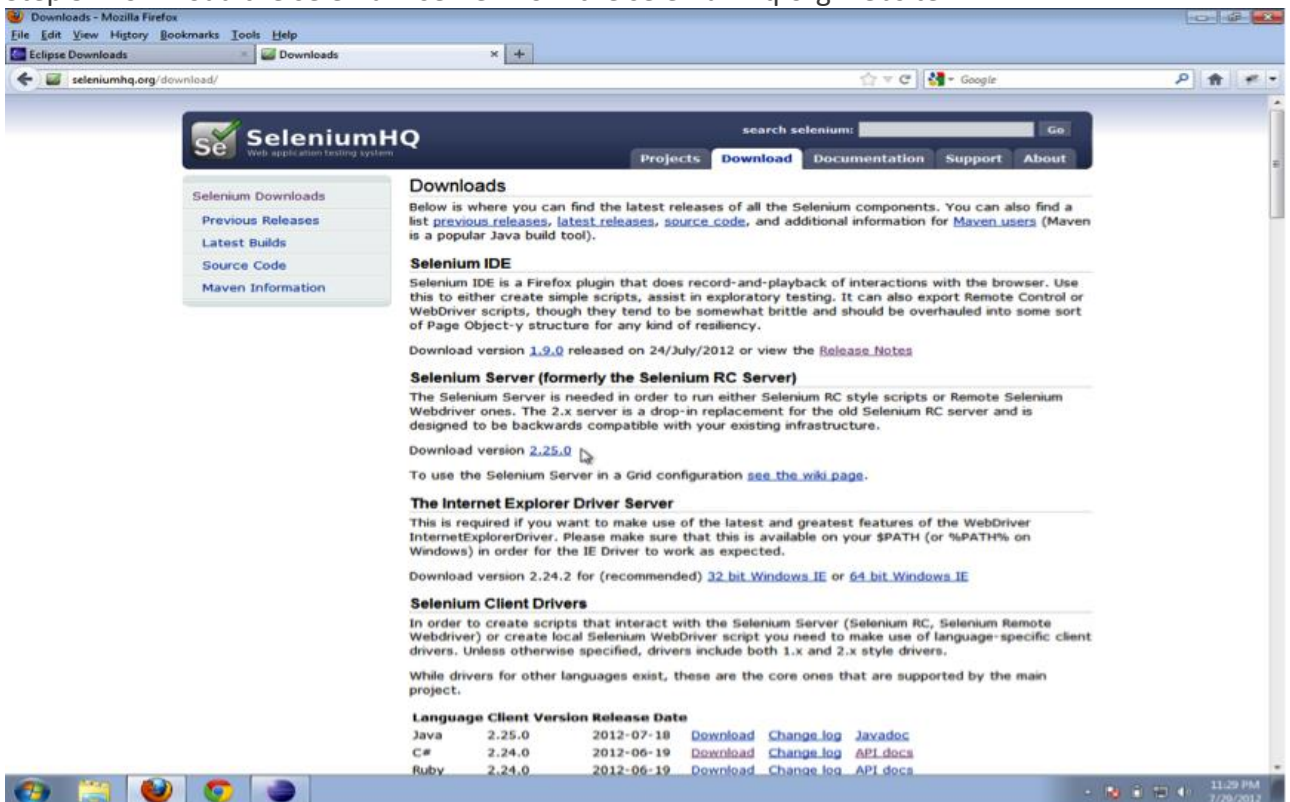
Step 7: Type In a Name for your Java Class file,  
 click on the check box for “public static void main (String args[])”  
 Click on Finish



Step 8: The Java class file is created and ready for Java Scripting.



Now, We need to add the Selenium Library files to our project  
 Step 9: Download the selenium server from the seleniumhq.org website.



Step 10: Click on Downloads Tab and then click on the link of download for Selenium

## Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

While language bindings for [other languages exist](#), these are the core ones that are supported by the main project hosted on google code.

ru

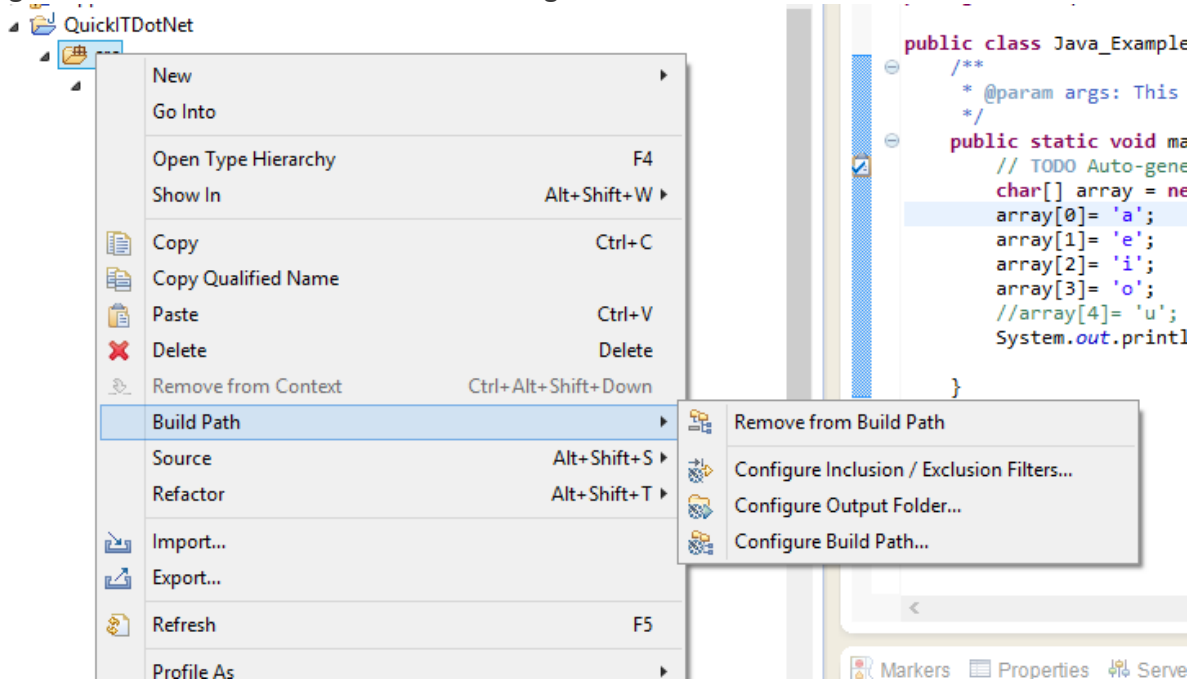
| Language          | Client Version | Release Date | <a href="#">Download</a> | <a href="#">Change log</a> | <a href="#">Javadoc</a>  |
|-------------------|----------------|--------------|--------------------------|----------------------------|--------------------------|
| Java              | 2.48.2         | 2015-10-09   | <a href="#">Download</a> | <a href="#">Change log</a> | <a href="#">Javadoc</a>  |
| C#                | 2.48.0         | 2015-10-07   | <a href="#">Download</a> | <a href="#">Change log</a> | <a href="#">API docs</a> |
| Ruby              | 2.48.0         | 2015-10-07   | <a href="#">Download</a> | <a href="#">Change log</a> | <a href="#">API docs</a> |
| Python            | 2.48.0         | 2015-10-07   | <a href="#">Download</a> | <a href="#">Change log</a> | <a href="#">API docs</a> |
| Javascript (Node) | 2.47.0         | 2015-09-15   | <a href="#">Download</a> | <a href="#">Change log</a> | <a href="#">API docs</a> |

C# NuGet

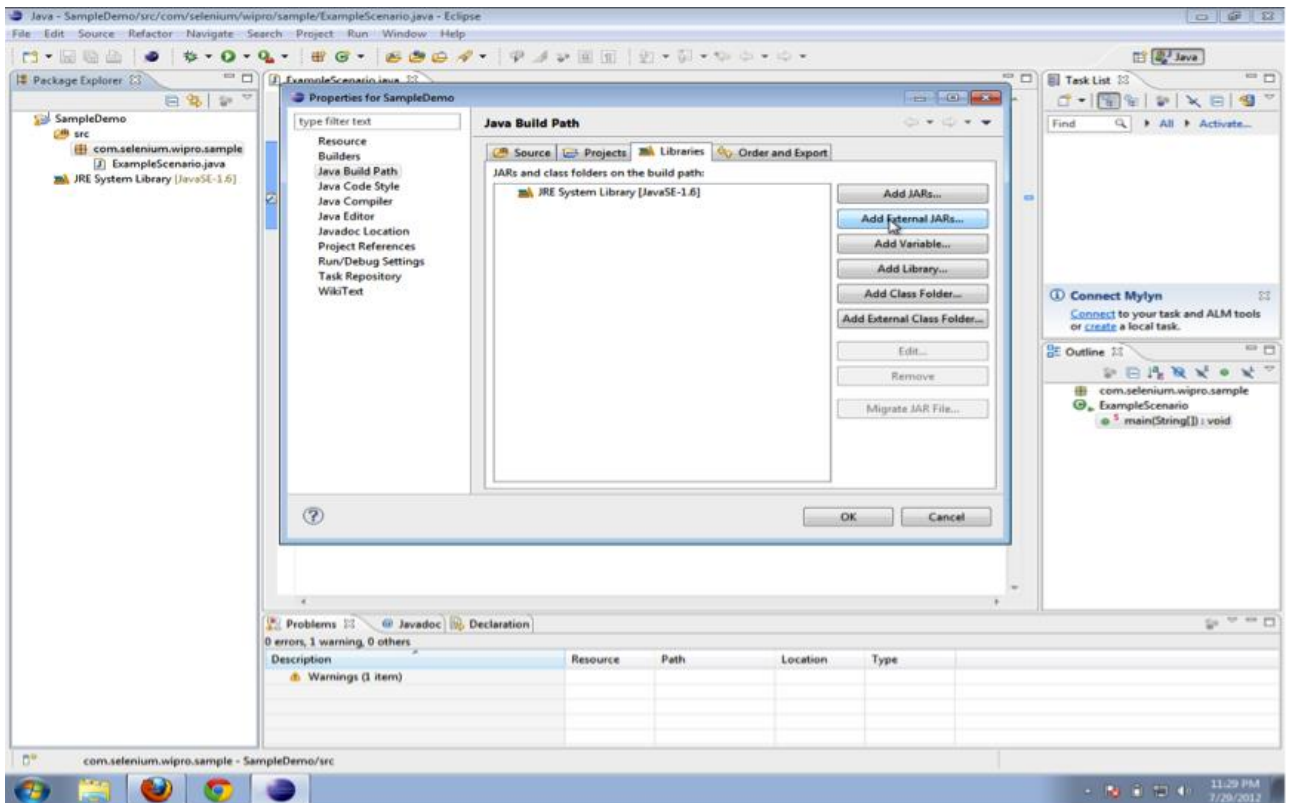
Step 11: Save the jar file in a specific Location E.g C:/Selenium

Step 12: Configuring Build Path.

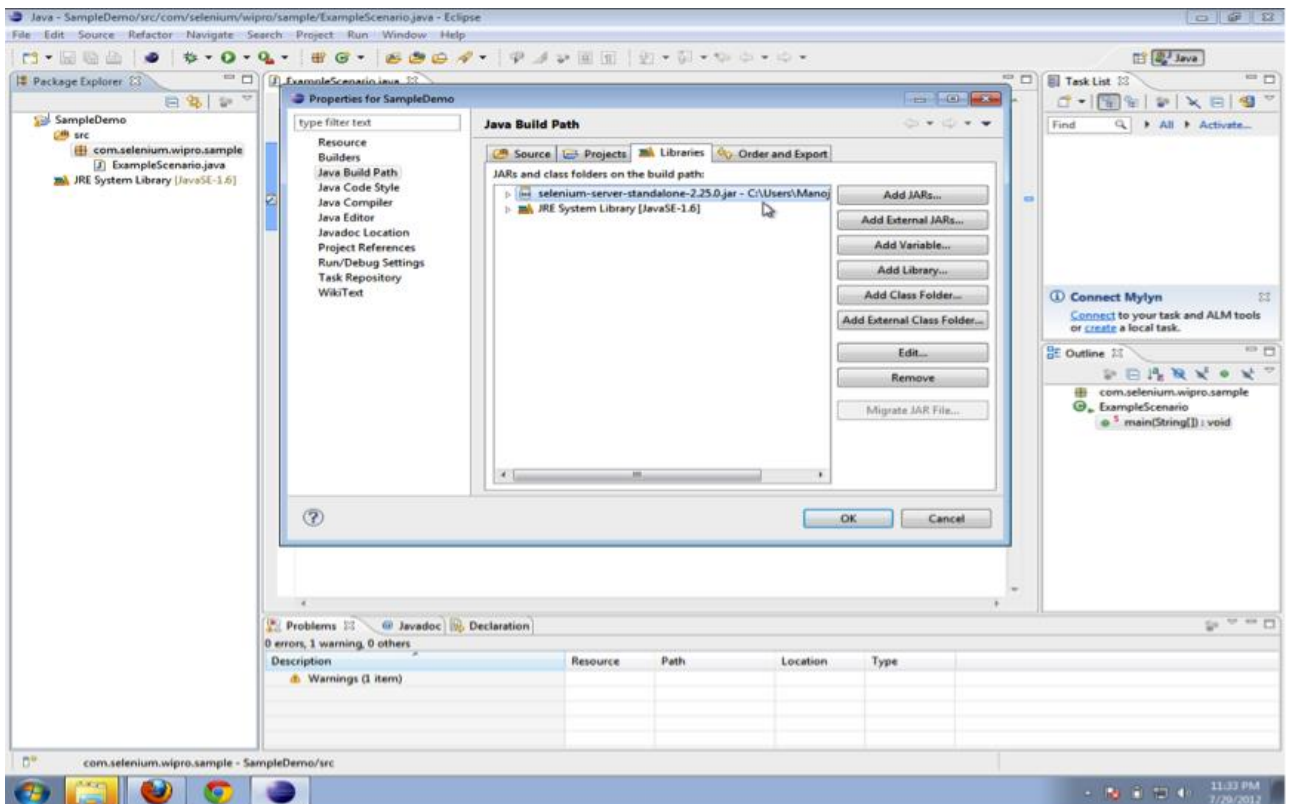
Right Click on the Src → Build Path → Configure Build Path



Step 13: Click on Libraries tab → Add External Jar's

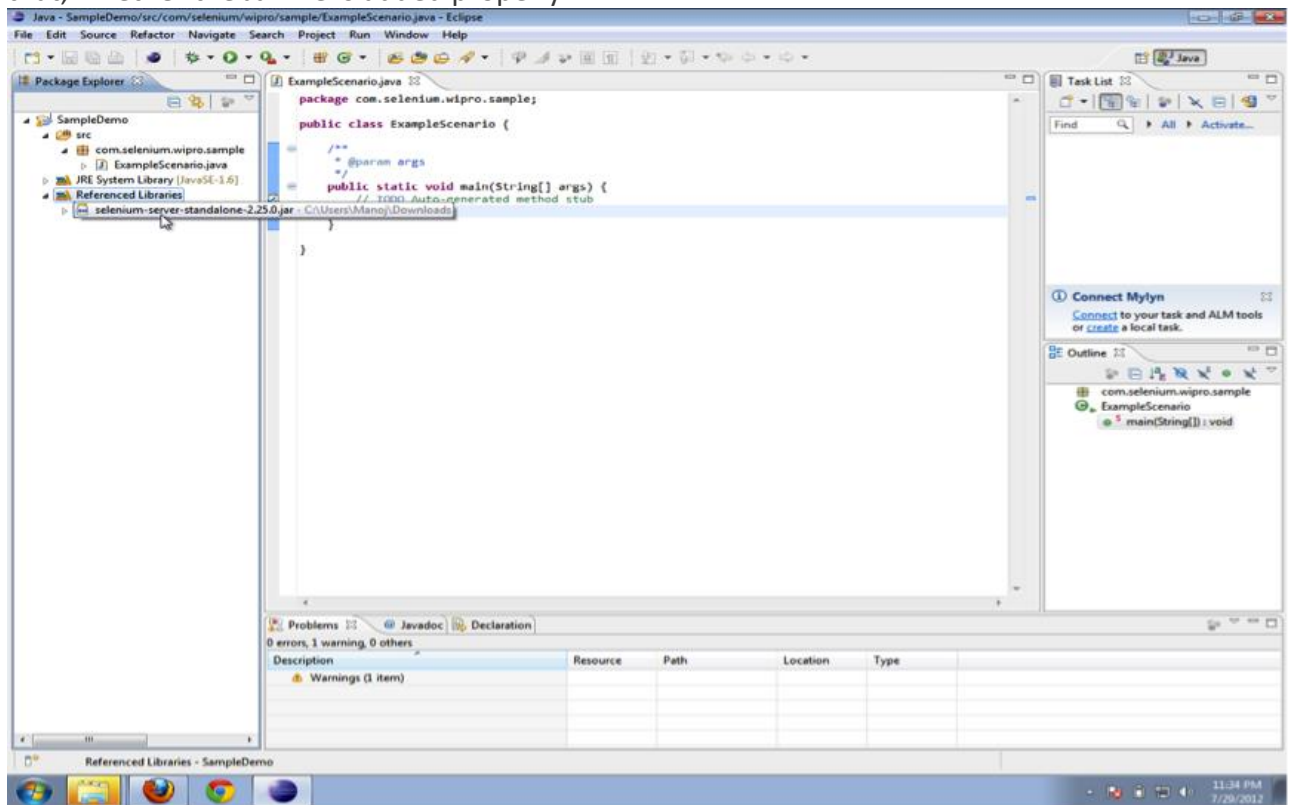


Step 14: The jar files should be loaded from the C:/Selenium folder (or where you have saved the downloaded Jar file from Seleniumhq website) to our Eclipse Workspace and should look as in the screenshot





Step 15: We can also verify by expanding the Referenced Libraries in the Project Explorer that, whether the Jar file is added properly

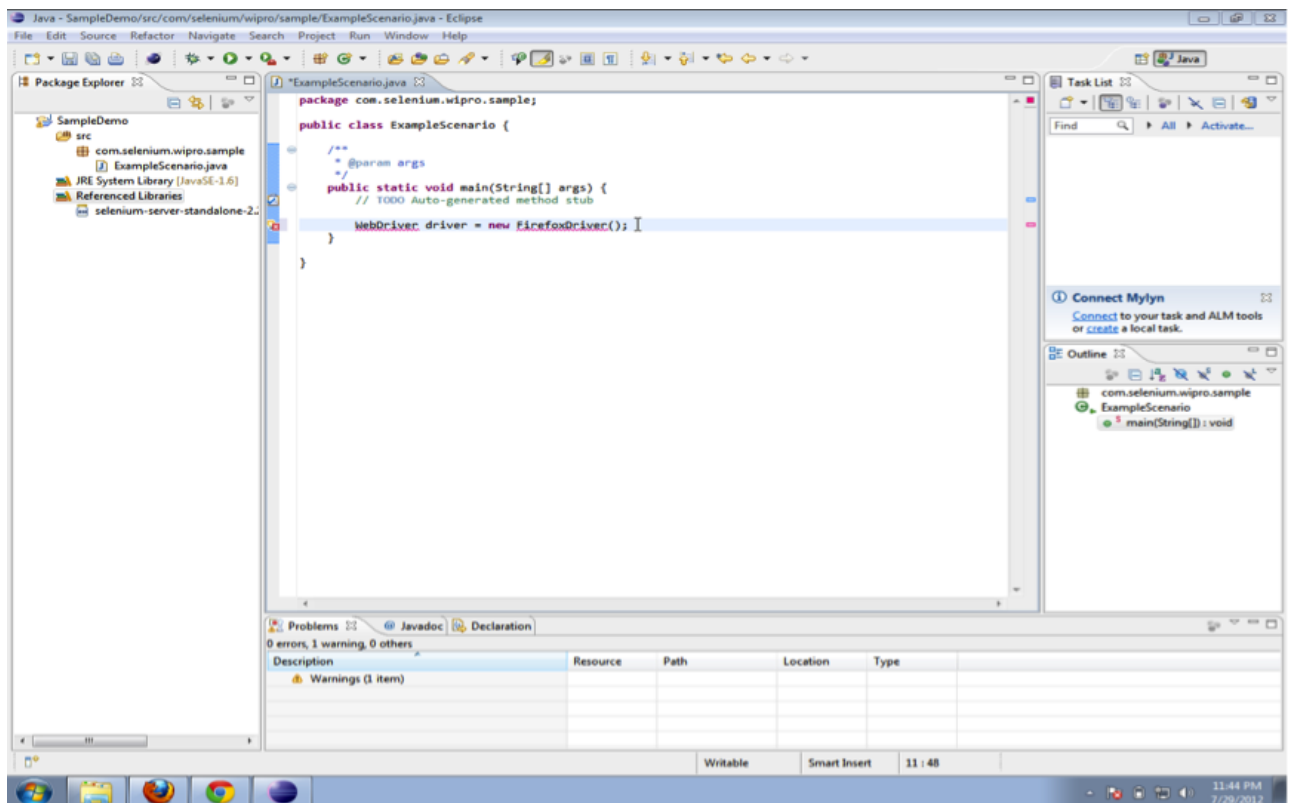


Step 16: Now, our Eclipse is ready for Selenium Scripting

Web Driver object initialization as

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

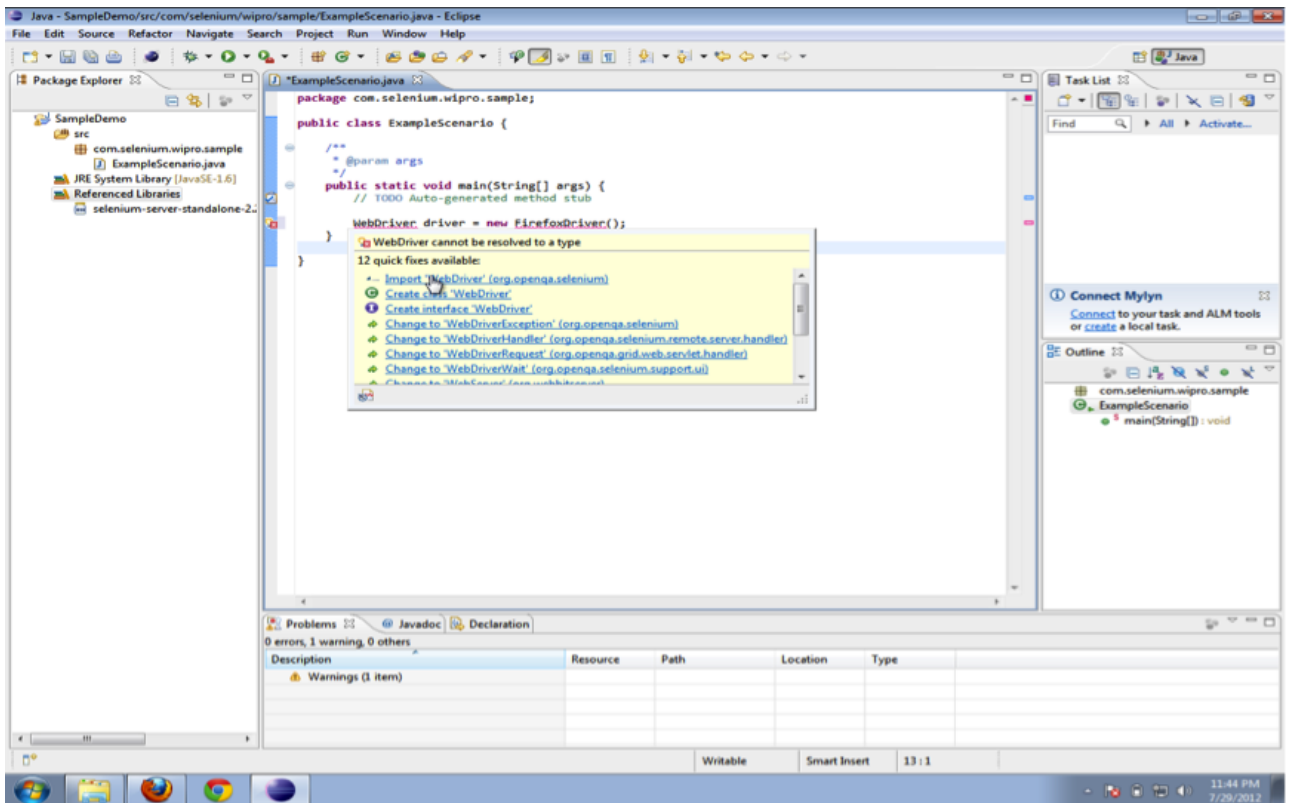
As in the below Screenshot



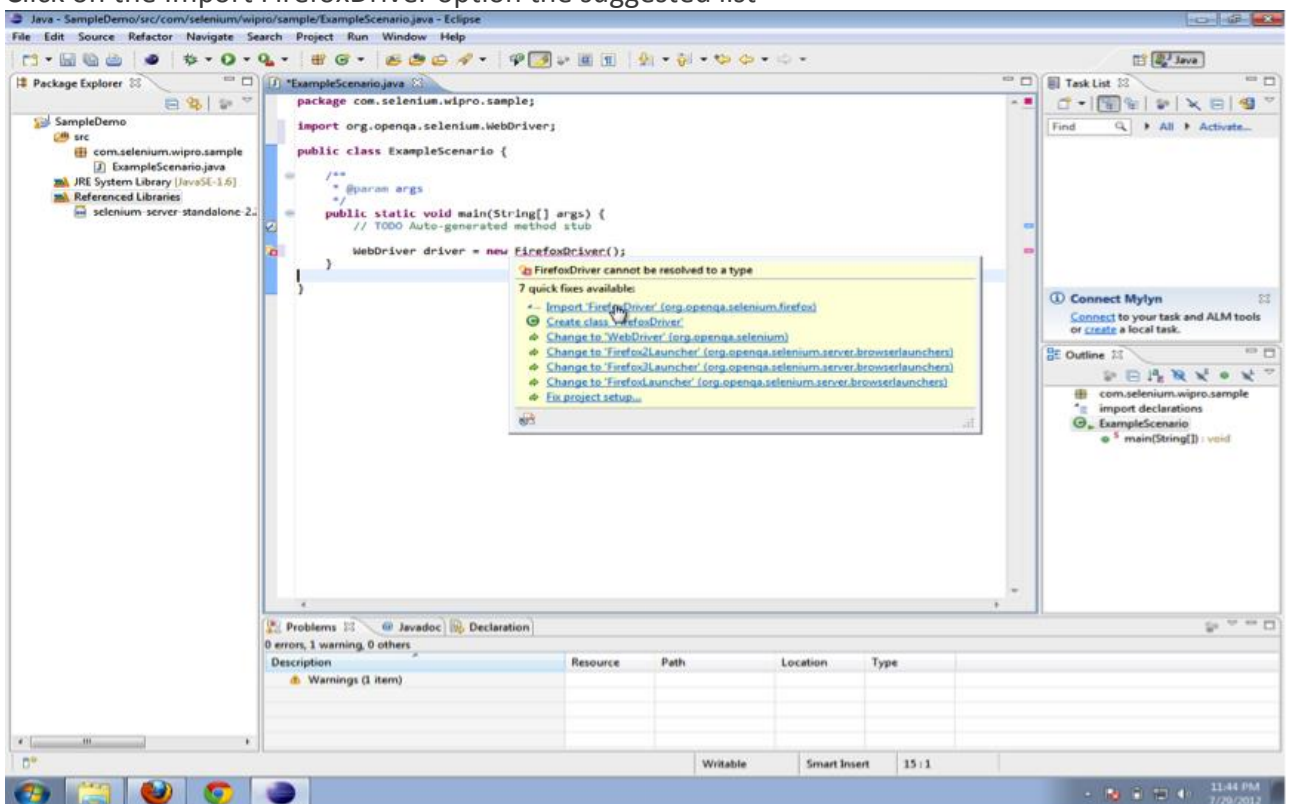
You may see errors, but that's not a problem here is how we resolve it.

Step 17: Mouse Hover near the WebDriver error line you will see an Auto suggest Option, you will find Import WebDriver.

Click on Import WebDriver –Now the error will be vanished.



Step 18: Now Mouse Hover near the Firefox Driver, and you will get an autosuggest option. Click on the Import FirefoxDriver option the suggested list



You can also eliminate the steps 15,16,17 by manually adding the below Import statements

```
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;
```

And, that's it we are done with Selenium Project setup in Eclipse.

Hope this helps!!!

Please comment if you have any questions/comments, I will try to reply at the earliest.