Selenium: An Automated Testing Tool

Dr. D. P. Mohapatra
Associate Professor
Department of Computer Science & Engineering
National Institute of Technology, Rourkela-769008
durga@nitrkl.ac.in

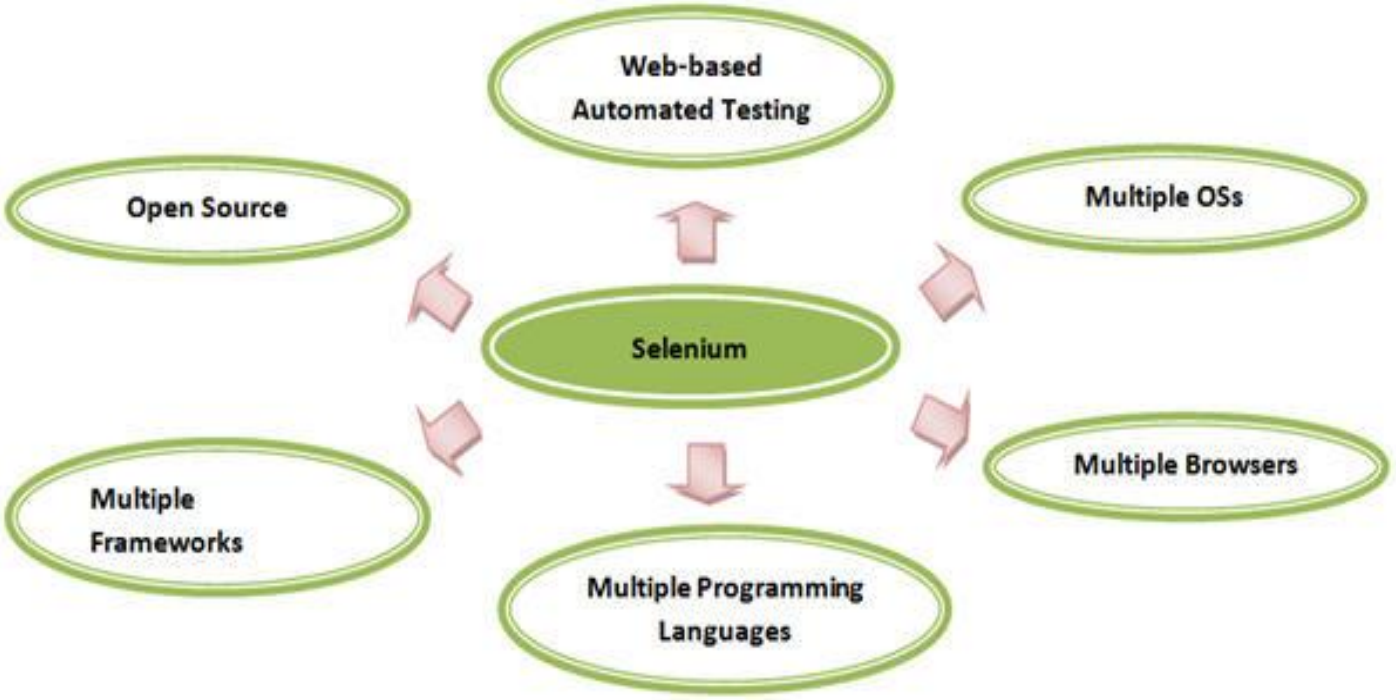
Plan of the Talk

- Introduction to Selenium
- Selenium Integrated Development Environment (IDE)
- Installation of Selenium IDE
- Using Selenium IDE
- Selenium Remote Control (RC)
- Selenium Web Driver

Overview of Selenium

- It is an open source automation testing tool for web based applications.
- It is licensed under Apache License 2.0.
- Different Languages supported by Selenium are Java, C#, Ruby, Python, PHP, Perl.
- Operating systems Supported by Selenium include windows, mac, linux, unix.
- Different Browsers supported by Selenium are Mozilla (till latest version), IE 6,7,8, Google chrome, Opera 8, 9, 10.

Selenium contd….



Selenium supports a broad range of browsers, technologies and platforms

Selenium: History

- Developed in 2004 by Jason Huggins as a JavaScript library and used to automate his manual testing routines
  - That library eventually became Selenium Core, which underlies all the functionality of Selenium Remote Control (RC) and Selenium IDE.
  - Sometimes it is very difficult to perform the testing because of its JavaScript based automation engine and the security limitations browsers apply to JavaScript.

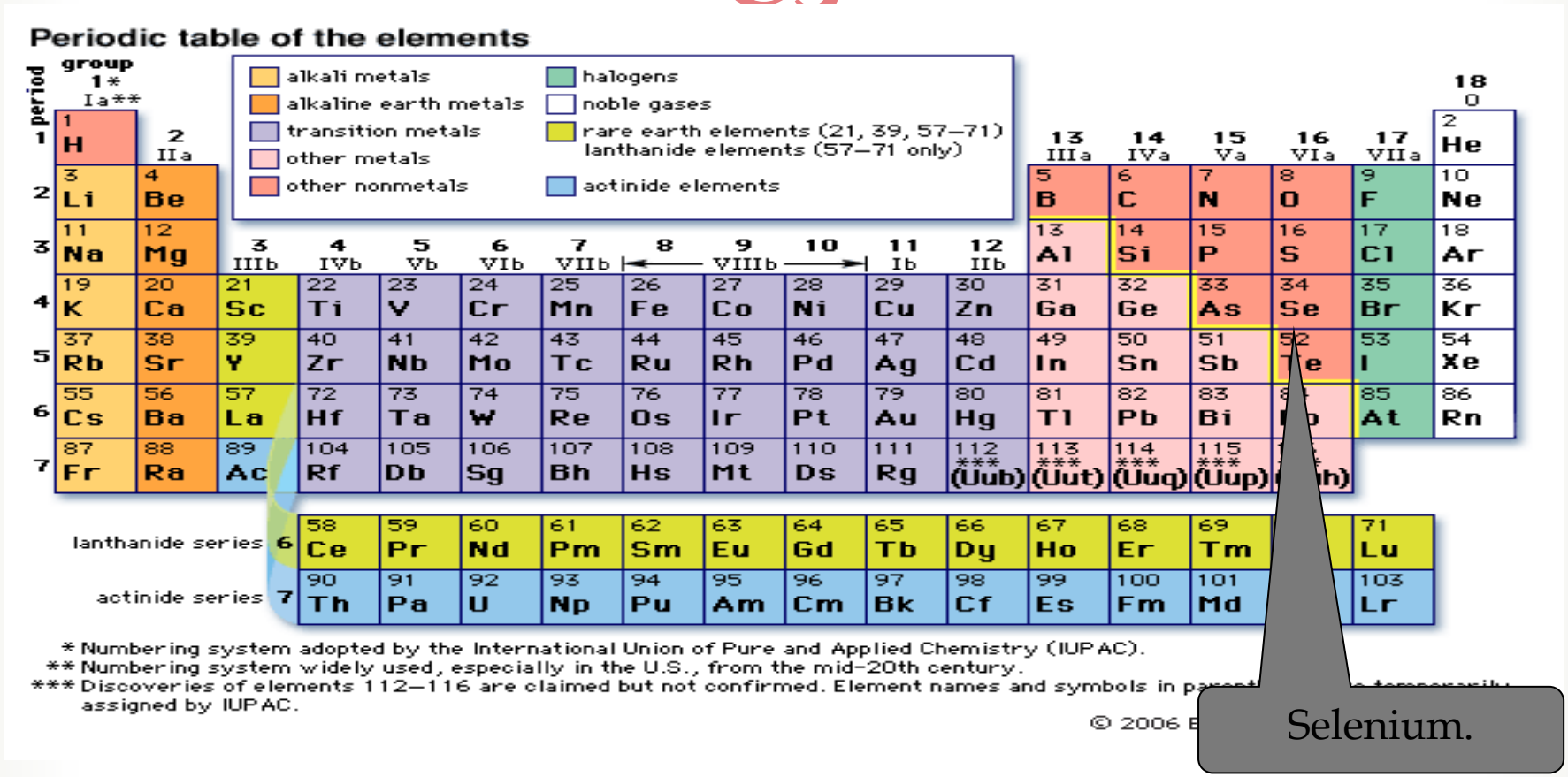
Selenium: History

- Google being a long time user of Selenium, had a developer named Simon Stewart who developed WebDriver.
  - Circumvented Selenium's JavaScript sandbox to allow it to communicate with the Browser and Operating System directly
- In 2008, Selenium and WebDriver merged to provide an excellent test automation framework

Story about Selenium

- Selenium is a chemical element with the atomic number 34, represented by the chemical symbol Se.
- It is a nonmetal, chemically related to sulfur and tellurium, and rarely occurs in its elemental state in nature.

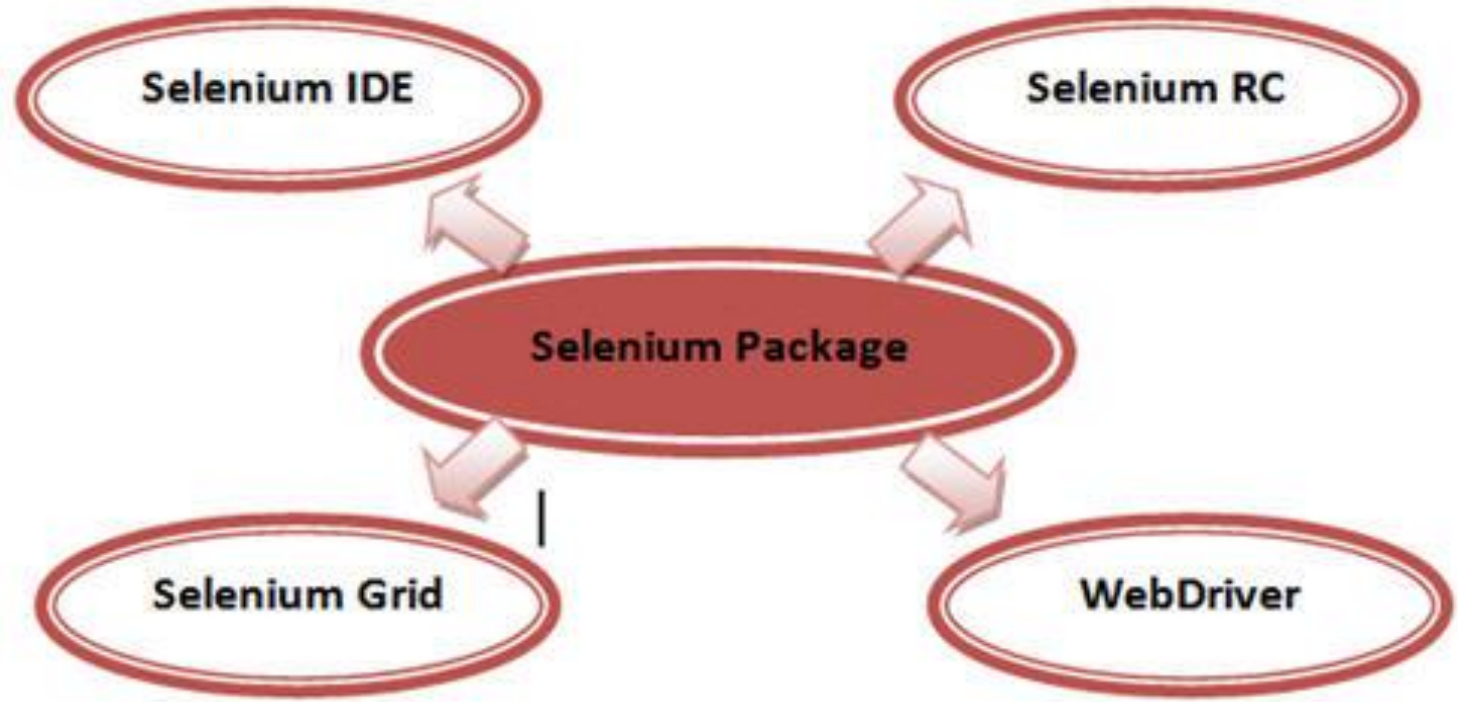
Story about Selenium



28 October 2020

Selenium is used for treating Mercury Poisoning

Components of Selenium



Selenium Packages - © www.SoftwareTestingHelp.com

Components of Selenium Contd…

- **Selenium IDE**
  - Selenium IDE provides facility to export recorded script in many languages like HTML, Java, Ruby, Python, C#, Junit and TestNG.
  - Script created for application testing is called test case in selenium IDE language and set of test cases is called test suite in selenium IDE.
  - It provides record and playback facility to regression test in any web application.

Components of Selenium Contd…

- **Selenium RC**
  - This is the older version of selenium.
  - It works on multiple browsers.

Components of Selenium Contd…

- **Webdriver**
  - Web driver is the new version of selenium.
  - It also supports Android and Iphone Testing.

- **Grid**
  - Grid is used to run test cases parallely on multiple machines and browsers.

Comparison Table

Selenium IDE	Selenium RC	Web driver
Works only on Mozilla	Works on almost all browsers. Does not work on latest version of Firefox/IE	Works on latest versions of almost all browsers - Firefox, IE(6,7,8), Opera, Chrome
Record and run tool	No Record and run	No Record and run
No server required to start	Server is required to start	No server required to start
Core engine is JavaScript based	Core engine is JavaScript based	Interacts natively with browser application
Very simple to use. If using User extensions, you require knowledge on JavaScript which makes the work a little bit tough.	It's a simple and small API	Complex and a bit large API as compared to RC
Not at all object oriented	Less Object oriented API	Purely Object oriented API
Cannot move mouse with it	Cannot move mouse with it	Can move mouse cursor
Full xpaths have to be appended with 'xapth=\\' syntax	Full xpaths have to be appended with 'xapth=\\' syntax	No need to append 'xpath=\\'
No Listeners	No Listeners	Implementation of Listeners is provided
Cannot test iphone/Android applications	Cannot test iphone/Android applications	Can test iphone/Android applications

Purpose of use

- Selenium is an open source testing tool and hence it serves for cost-effective automation testing.
- One highly beneficial feature of Selenium is that the language used for building the program is independent of the language that the web application or website is using.
- This implies that the test script can be developed in any of the languages that Selenium supports.

Purpose of use

- With Selenium, it is convenient to implement frameworks that revolve around Object-oriented programming like Keyword Driven, Data driven and Hybrid.
- With the use of Selenium, it is possible to execute simultaneous tests leveraging various browsers on various machines.
- This turn cuts down the time for test execution when a large project is in progress.

Hardware Requirements

- **Operating system(s)**
  - Windows - XP, 2003, 2003 Server, Windows Vista, Windows 7, Windows 2008
  - Linux
  - MacOS - Leopard, Snow Leopard.
- **Memory**: Minimum 2 GB RAM.

Hardware Requirements

- **Space**: With all included features and libraries installed, standalone Twist takes about 250- 300 MB (depending on OS) space on disk.

- **Display**: Minimum 1024 x 768 resolution. However resolutions of 1280x800 or 1440x900 are preferred if you want to have many views/windows/tabs open on the Twist perspectives view.

Software Requirements

Firefox

- Please be sure you have a recent version of Firefox installed.

Java Script

- The Selenium Remote Control server runs on Java Script.
- It thus runs on any operating system that supports a recent Java implementation.

Software Requirements

You can also use the following programming language to write the code.

- **Ruby**
- **C#**
- **PHP**
- **Perl**
- **Python**

Selenium Integrated Development Environment (IDE)

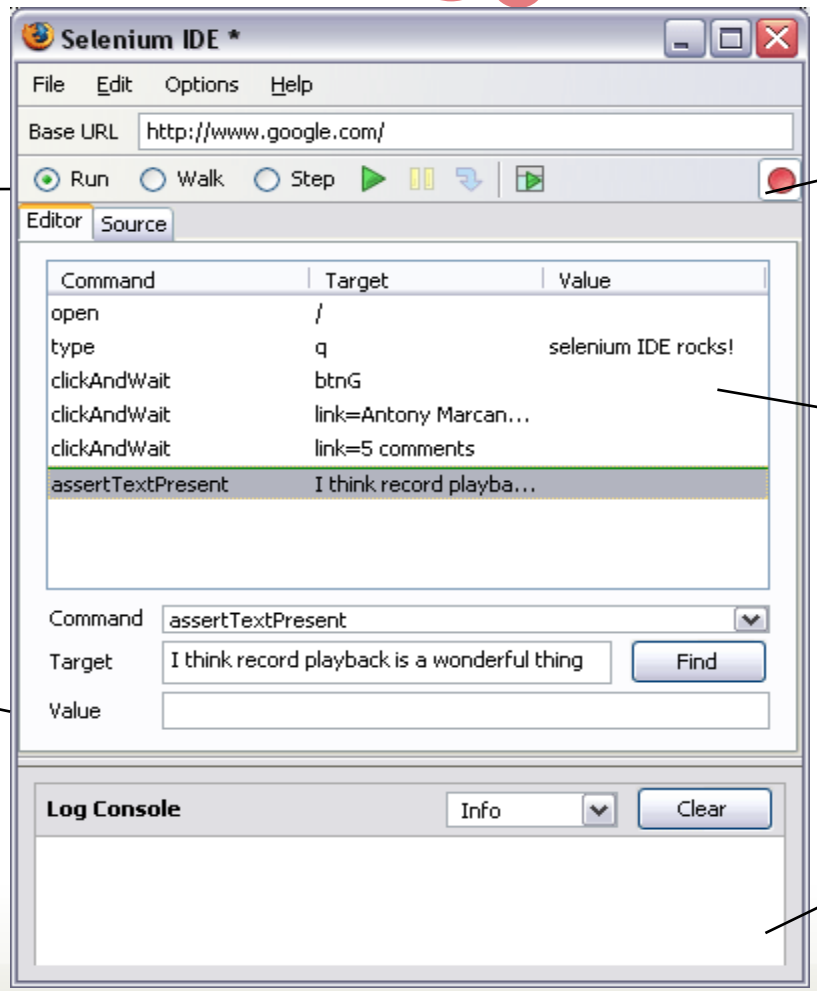
Selenium Integrated Development Environment (IDE)

- The Selenium-IDE (Integrated Development Environment) is the tool used to develop Selenium test cases.
- It's an easy-to-use Firefox plug-in and is generally the most efficient way to develop test cases.

Selenium Integrated Development Environment (IDE)

- It also contains a context menu that allows you
  - to first select a UI element from the browser's currently displayed page and then
  - select from a list of Selenium commands with parameters predefined according to the context of the selected UI element.

Selenium Integrated Development Environment (IDE)



Replay Toolbar

Start and Stop Recording

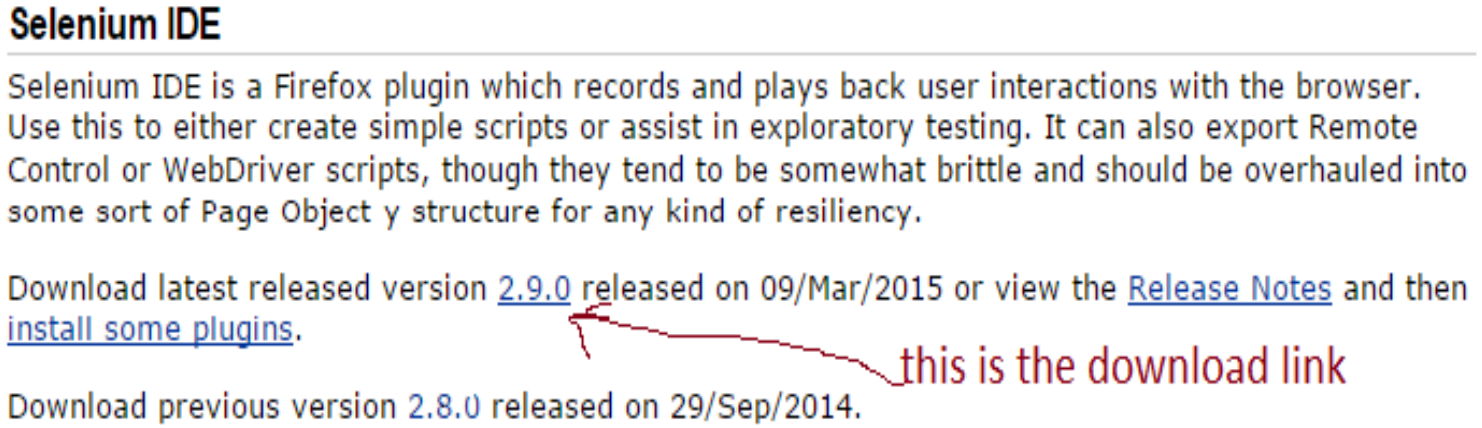
Selenese Script Editor

Accessory Area

Selenium Log

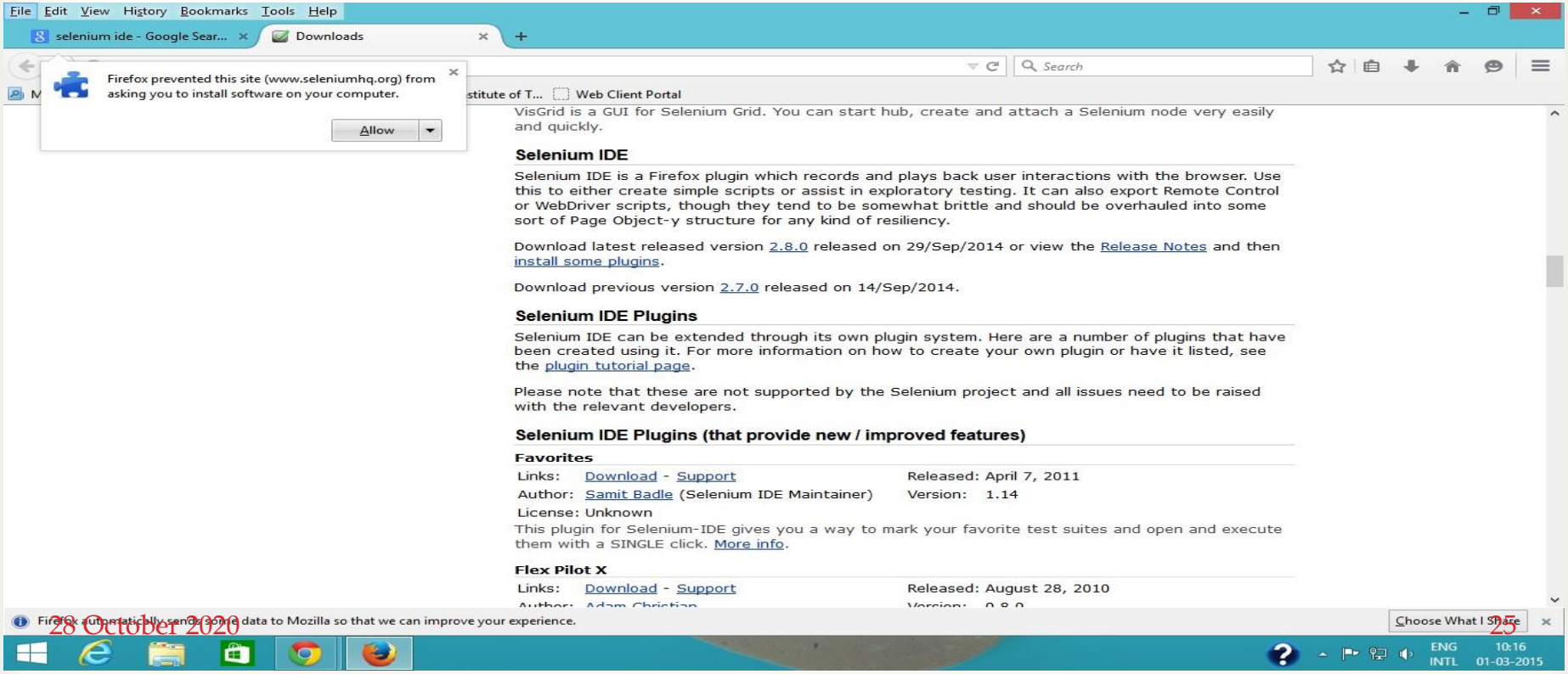
Installation of Selenium IDE

- Launch Firefox and navigate to http://seleniumhq.org/download/.
- Under the **Selenium IDE** section,
  - click on the link that shows the current version number.



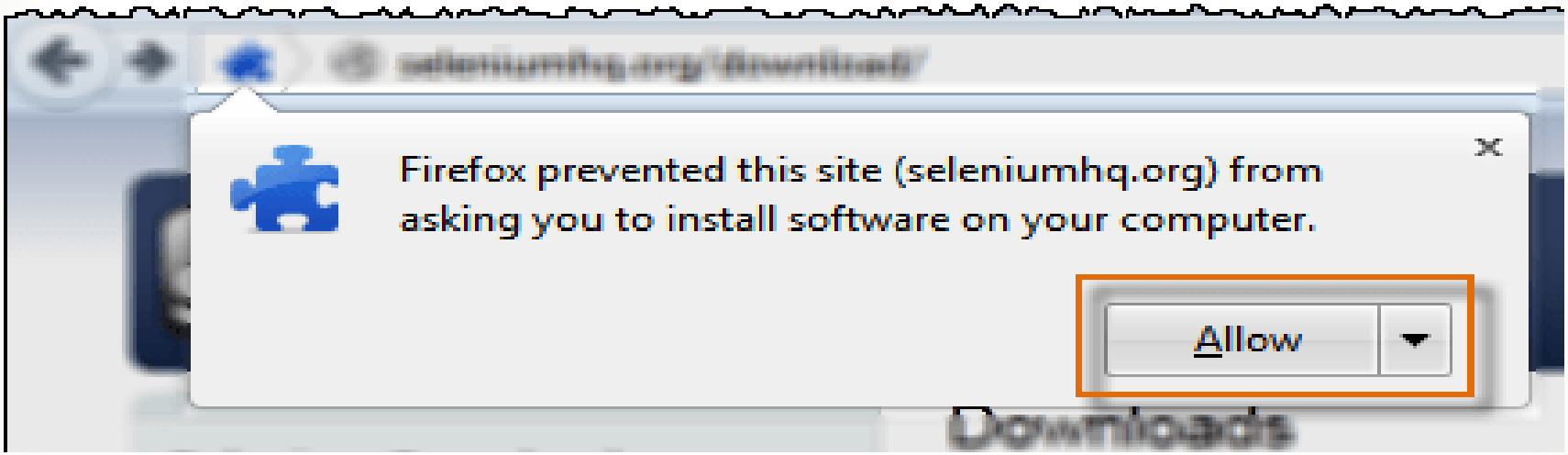
Installation of Selenium IDE contd….

- For security, a Firefox notification will pop up.



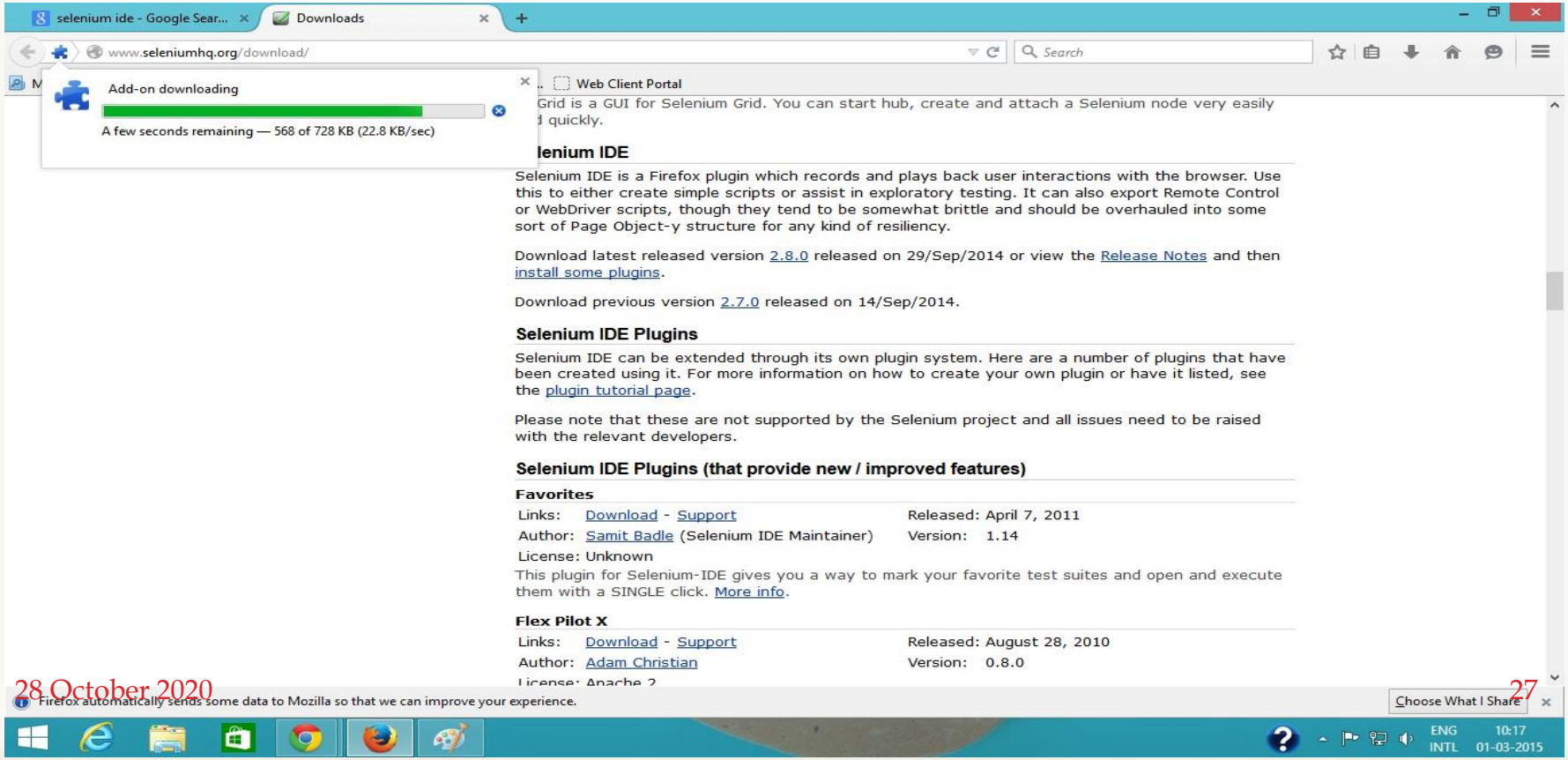
Installation of Selenium IDE contd….

- Click on "**Allow**''.



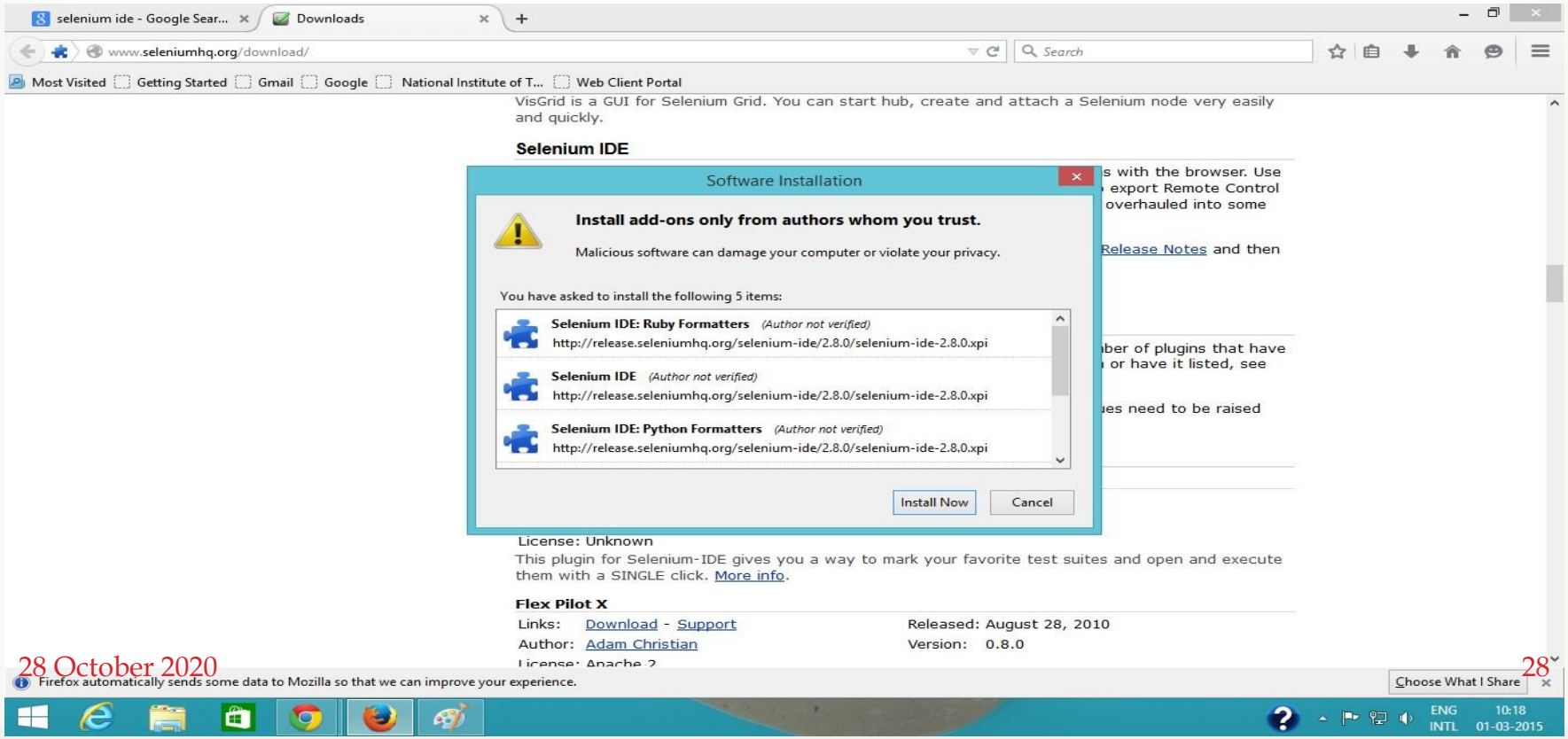
Installation of Selenium IDE contd….

- The addons will be downloaded



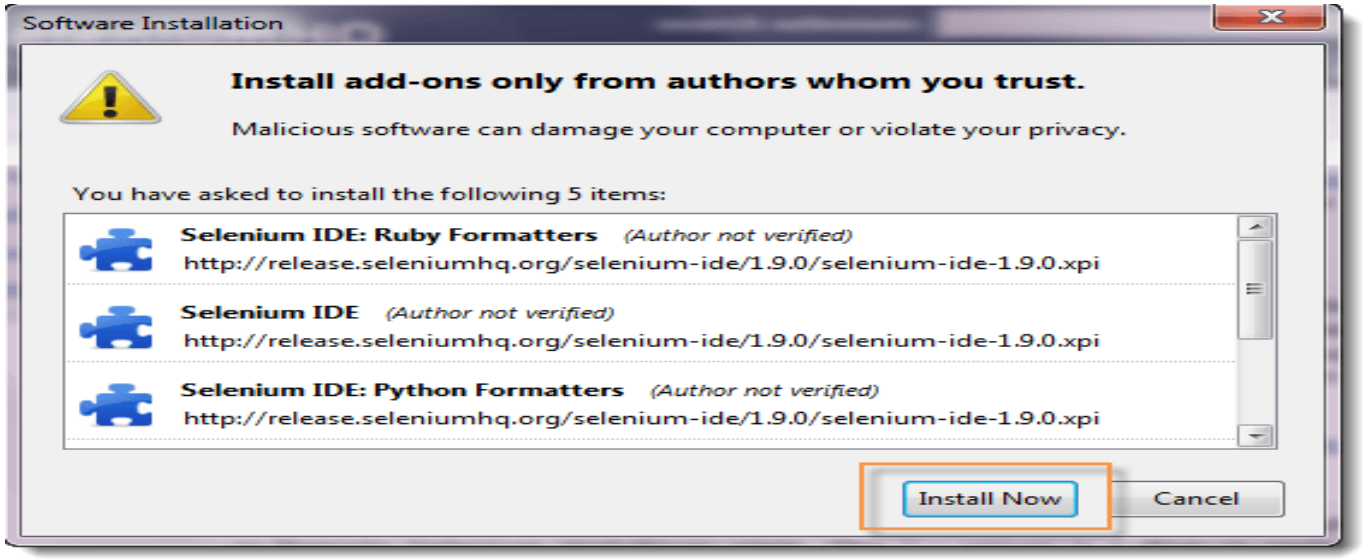
Installation of Selenium IDE contd….

- After downloading from Firefox, you'll be presented with a window shown



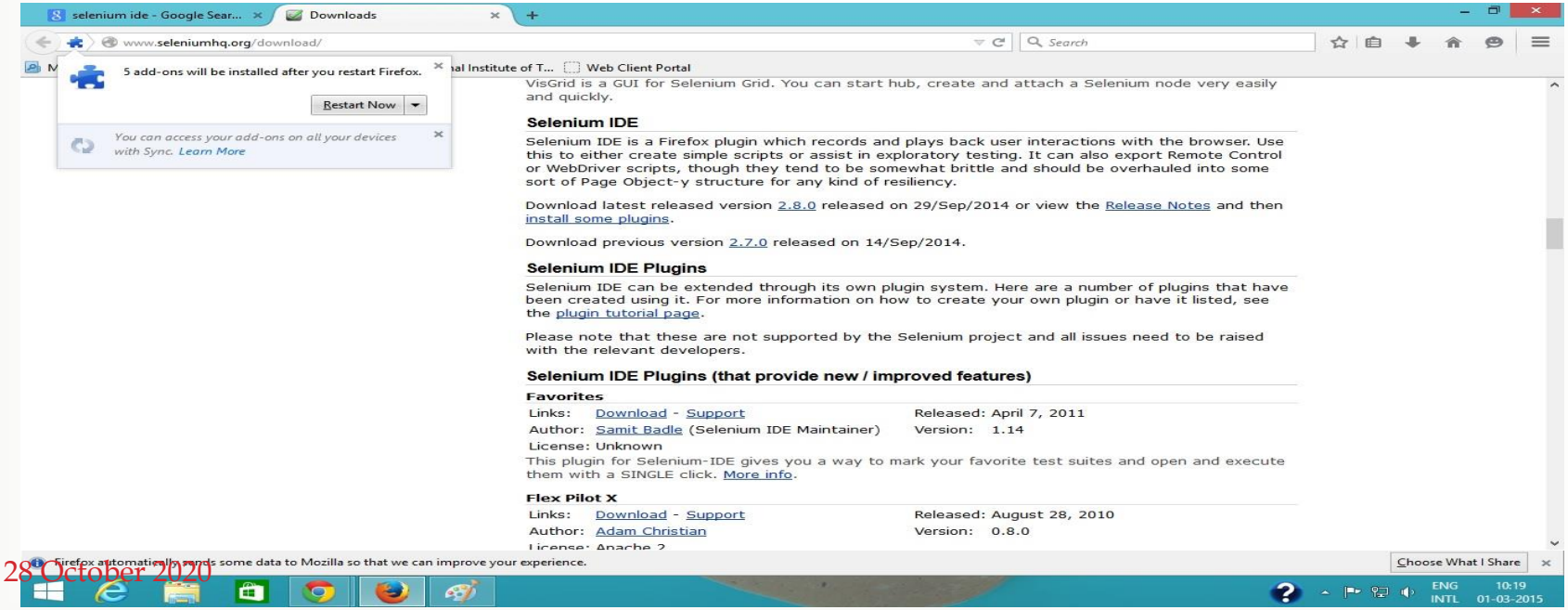
Installation of Selenium IDE contd….

- Select Install Now.



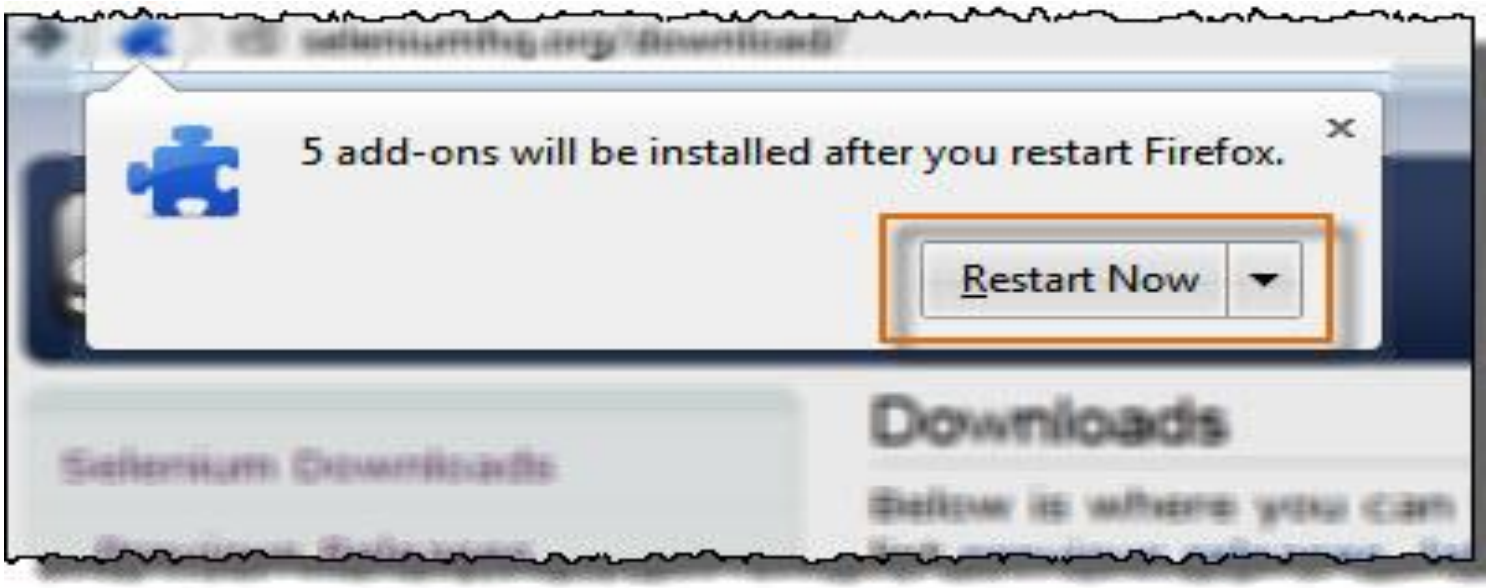
Installation of Selenium IDE contd….

The Firefox Add-ons window pops up, first shows a progress bar, and when the download is complete, displays the window shown



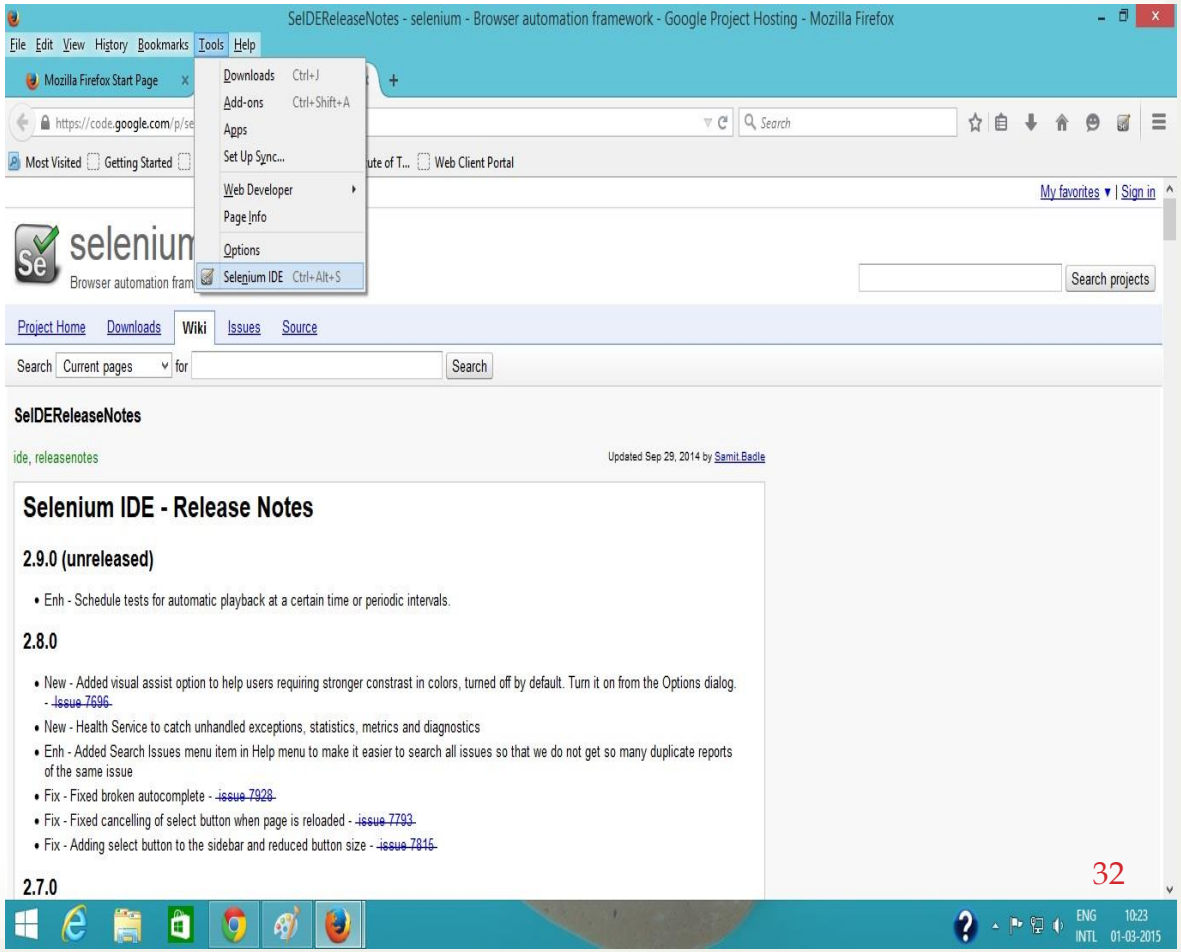
Installation of Selenium IDE contd….

- Click on Restart Now.



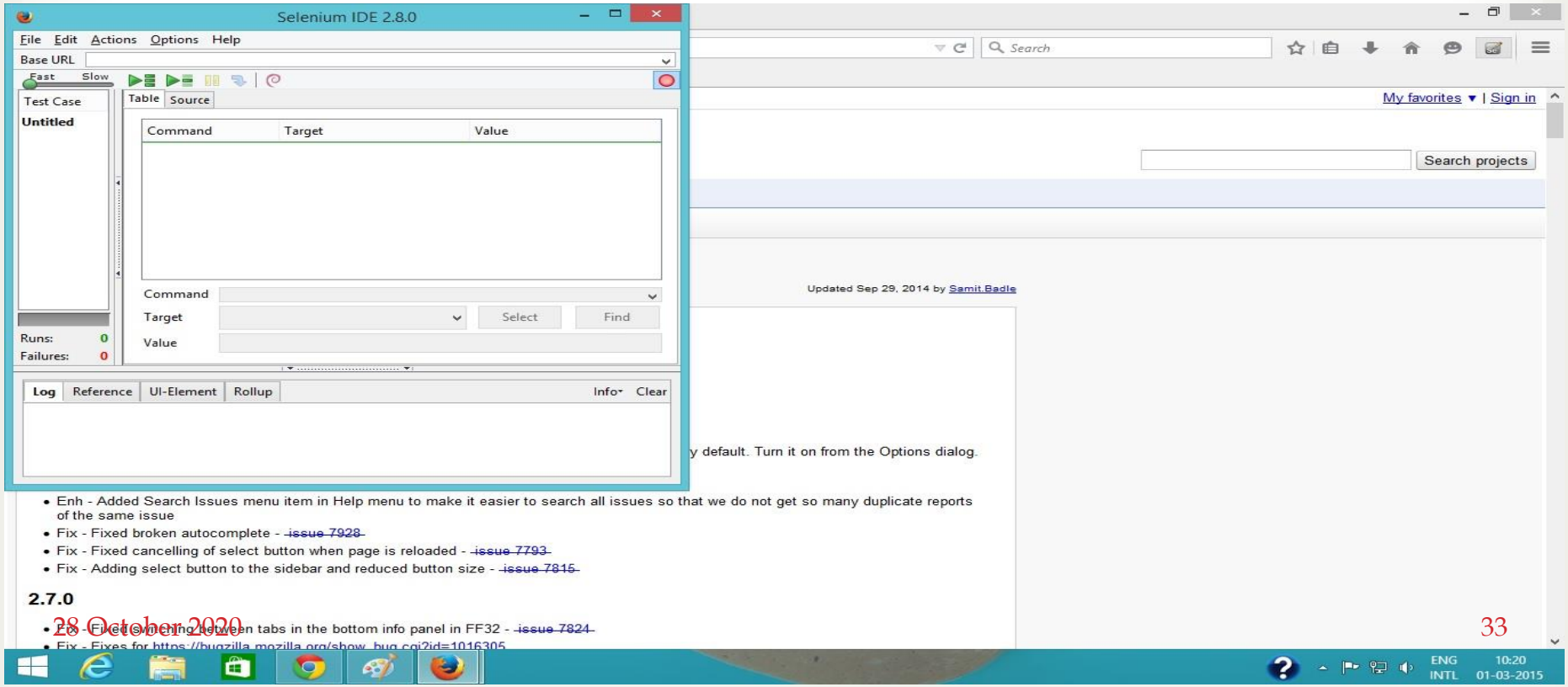
Installation of Selenium IDE contd….

After Firefox reboots you will find the Selenium-IDE listed under the Firefox Tools menu as shown.



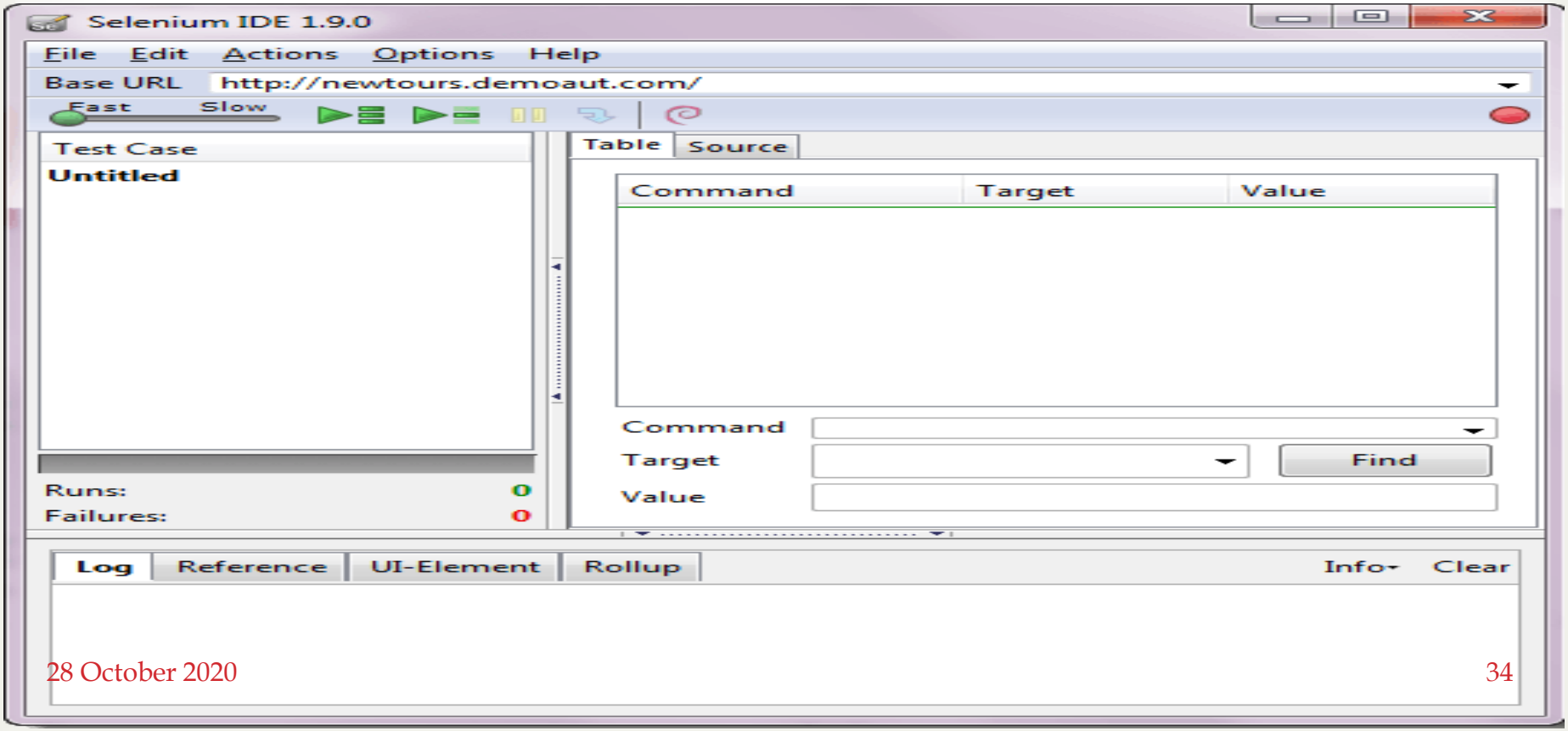
Installation of Selenium IDE contd….

- Also, you will see the Selenium-IDE icon in the toolbar as shown

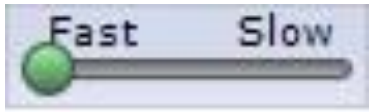


Installation of Selenium IDE contd….

- Selenium IDE should launch as shown below



Elements in Selenium IDE Toolbar

Speed Control 

Run All TCs 

Run single TC 

Pause 

Resume 

Step 

Record 

Recording a Selenium Test Case

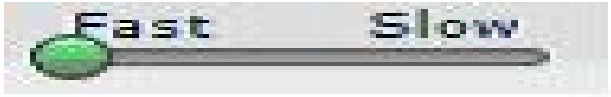
- Open Firefox that has the IDE installed
- Open the base URL of the application to record.
- Keep the application in a common base state.
- Go To Tools → Selenium IDE and the IDE will be opened
- Start recording using the RED Button in the Tool bar.

Recording a Selenium Test Case

- Now perform the operations on the application as you are testing the application.
- Once you are done with the recording
  - Click on the Stop recording button and save the test case through the file menu.
  - By default it will be saved as a selenese script (HTML format)

Running Your First Selenium Script

- Make sure the application is in the common base state.
- Click on the Run button.
  - Here you can also control the speed of the execution using the toolbar



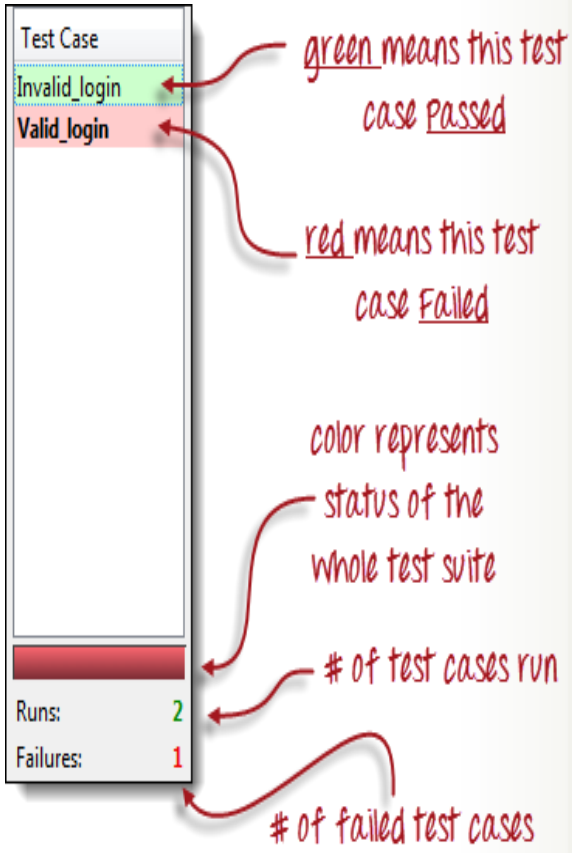
- Once the test is run you can view the test log in the bottom of the IDE window

Selenium IDE Test Case Pane

- In Selenium IDE, you can open **more than one test case at a time**.
- **The test case pane shows you the list of currently opened test cases.**
- When you open a test suite, the test case pane will **automatically list all the test cases** contained in it.

Selenium IDE Test Case Pane

- The test case written in **bold font** is the **currently selected test case**
- At the bottom portion is a summary of the number of test cases that were run and failed.
- After playback, **each test case is color-coded** to represent if it passed or failed.
  1. Green color means "Passed."
  2. Red color means "Failed."
  3. Yellow color means "Currently running".



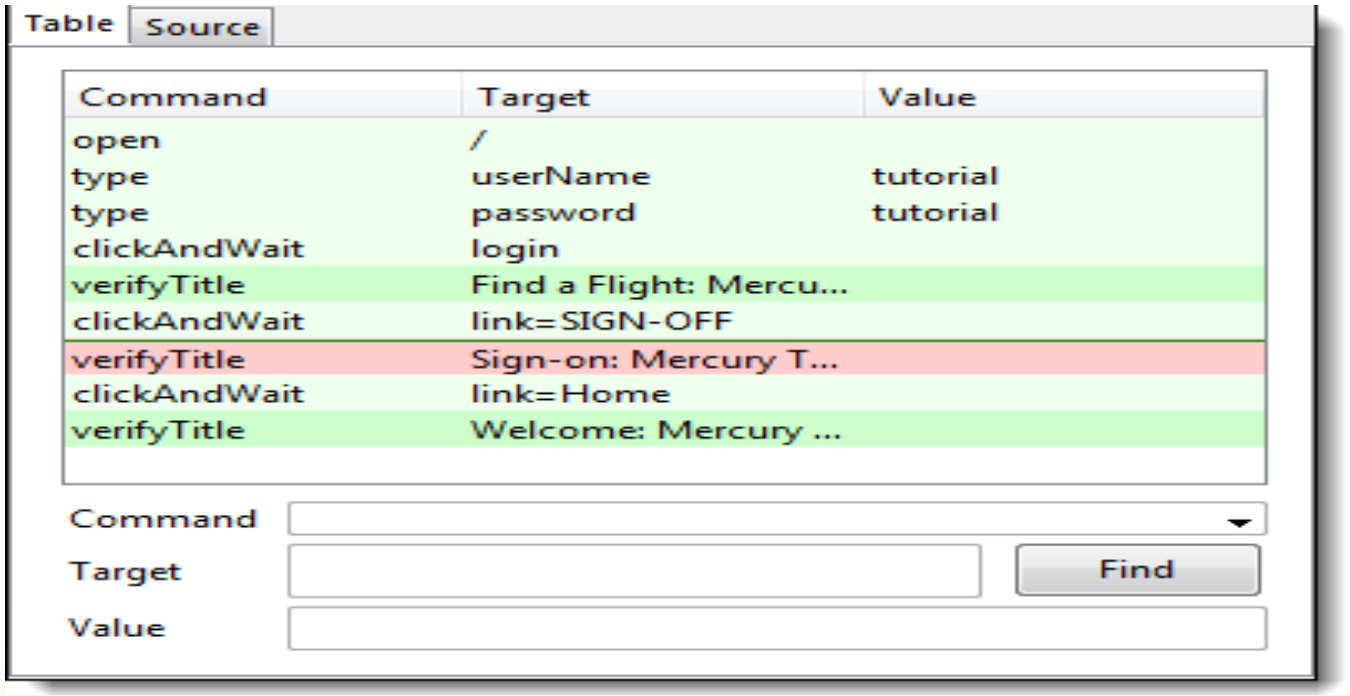
Selenium IDE Editor

- You can think of the editor as **the place where all the actions happen**. It is available in two views:
    1. Table
    2. Source.

Selenium IDE Editor

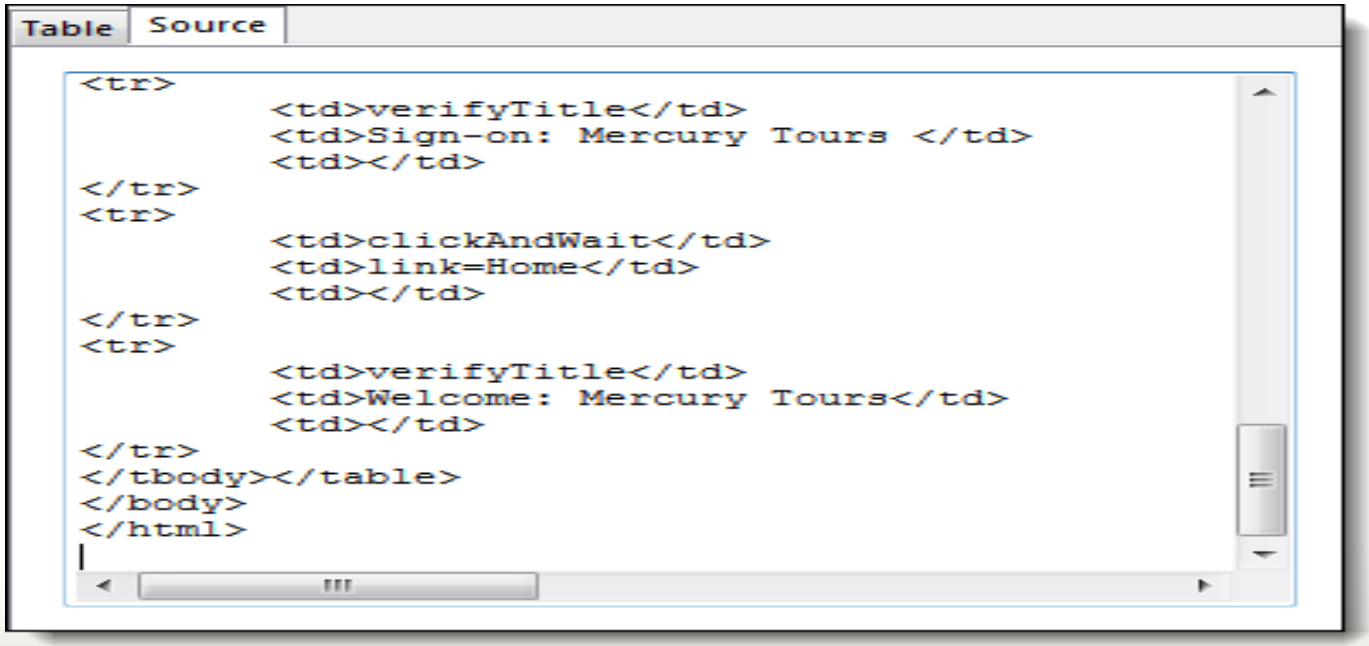
Table View

- This is **where you create and modify Selenese commands.**
- After playback, each step is color-coded.



Selenium IDE Editor

- **Source View**
  - **It displays the steps in HTML (default) format.**
  - **It also allows you to edit your script** just like in the Table View.

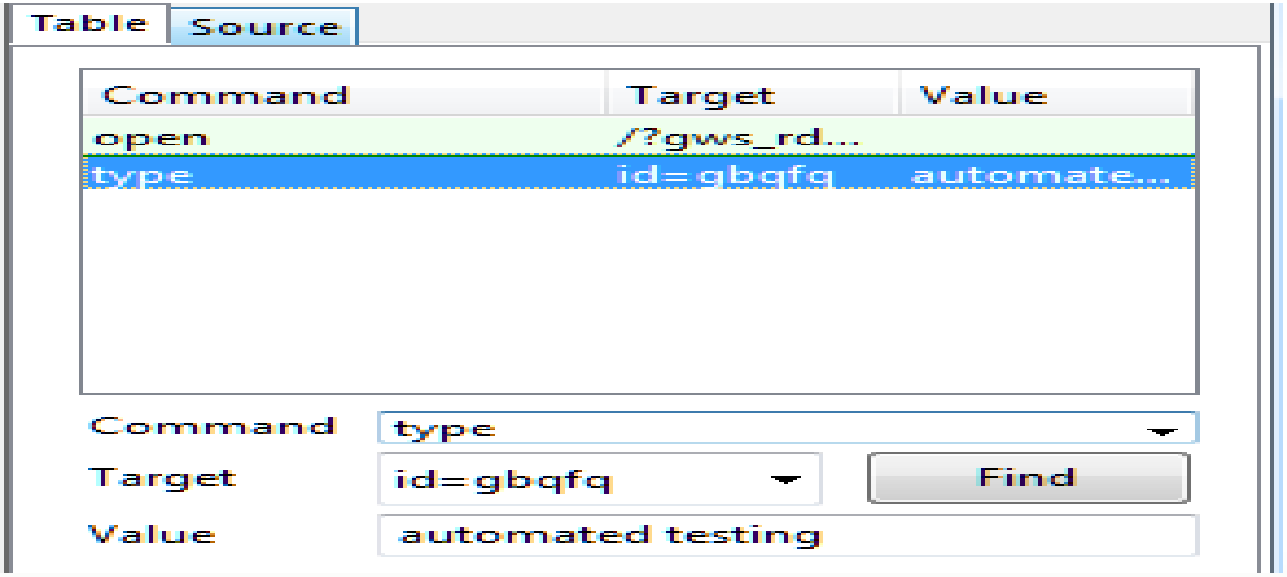


Using Selenium IDE

- Launch Selenium IDE.
- Drag the bar in “Speed Control” to “slow”.
- Click on red button to start recording.
- Open a web page in the firefox browser. (e.g. google.com)
- Type some query in the “google” page.

Using Selenium IDE

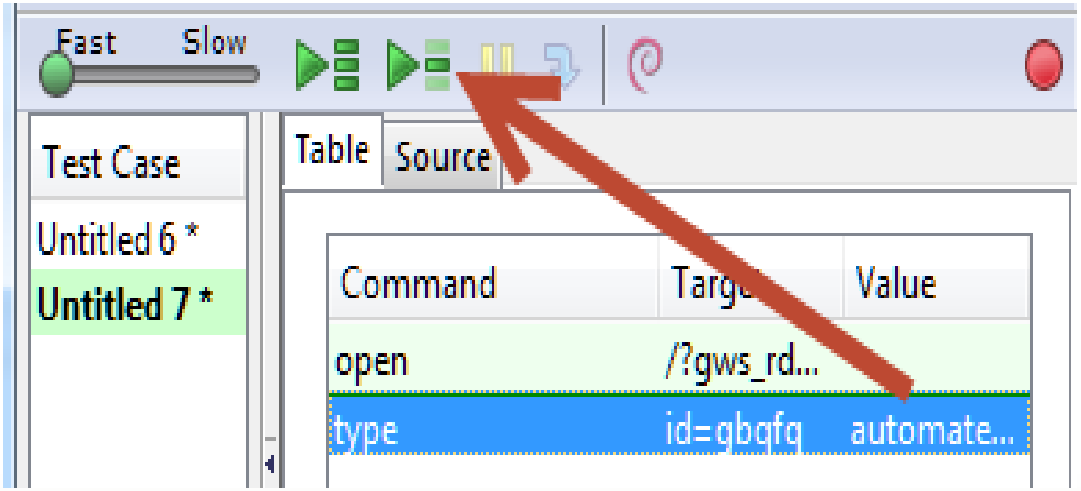
- Click on red button to stop recording.
- Click on File menu->Save Test Case
- The table pane looks like



Using Selenium IDE

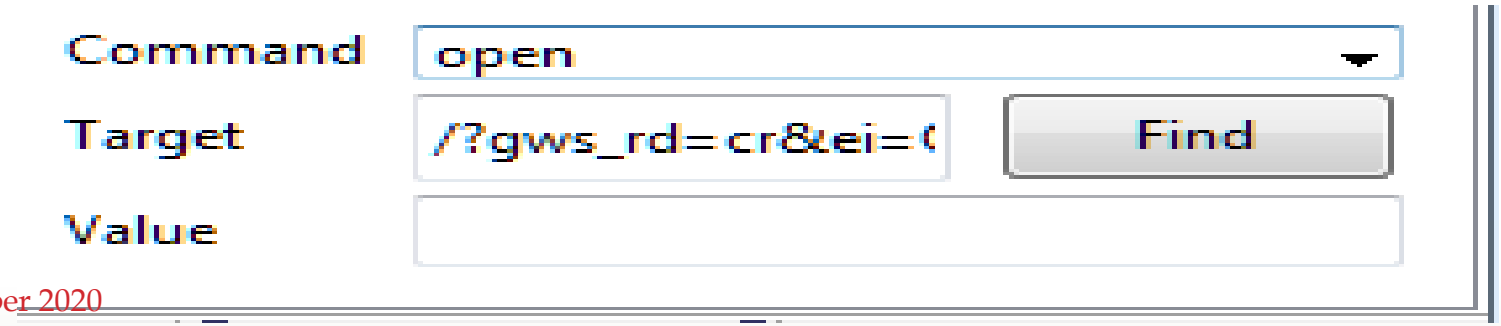
- Click on the 'Play current test case' button.

This will automatically open Google.com, type in 'automated testing' into the query field, and display the search results.



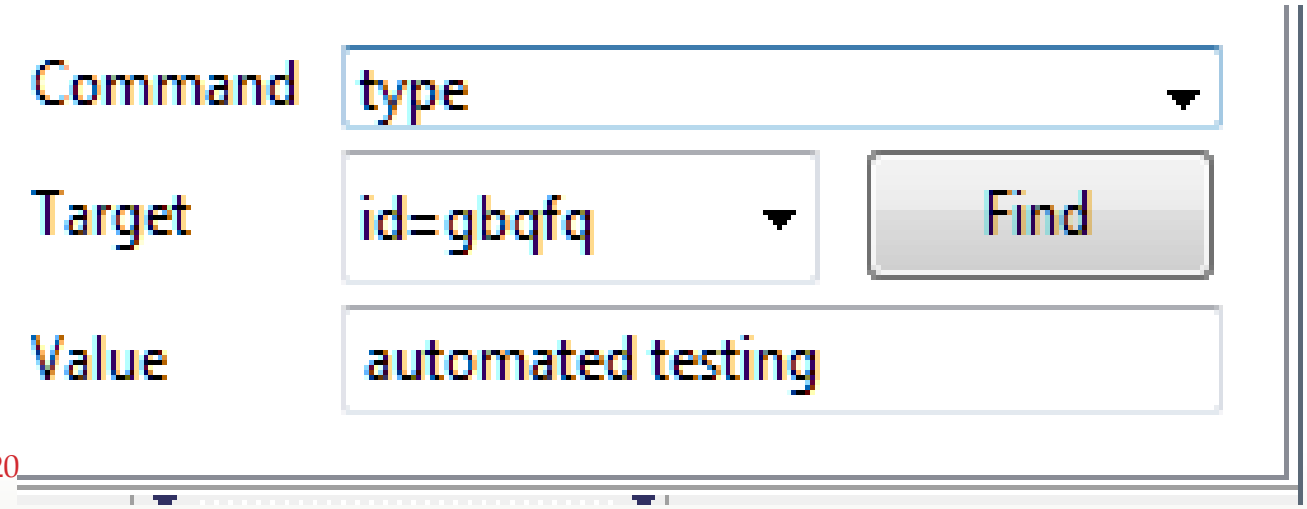
Using Selenium IDE

- If you take a closer look at the two commands in the Table pane, you'll see that you can not only change the command, but also alter the values associated with it.
- For example, the first command is 'open'. You can change this to anything else – Selenium IDE offers a huge list of built-in commands. You can also change the 'Target' URL to anything that you like.



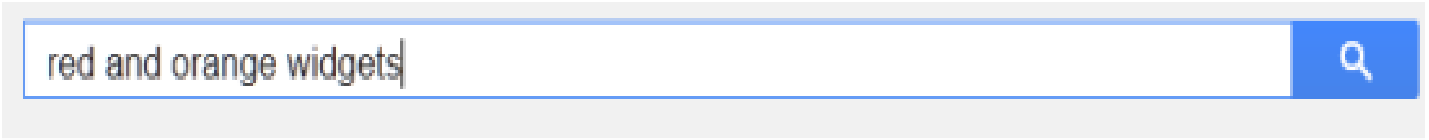
Using Selenium IDE

- The next command is ‘type’. This basically instructs Selenium to type-in the desired query into the search bar.
- You can change the query to anything you want by changing the ‘value’.



Using Selenium IDE

- For example, if you change it to ‘red and orange widgets’, Selenium IDE will automatically query ‘red and orange widgets’ into Google.



This is one of the most basic examples of what Selenium IDE can do. By combining different commands, you can make the IDE run complex tasks and test for bugs automatically.

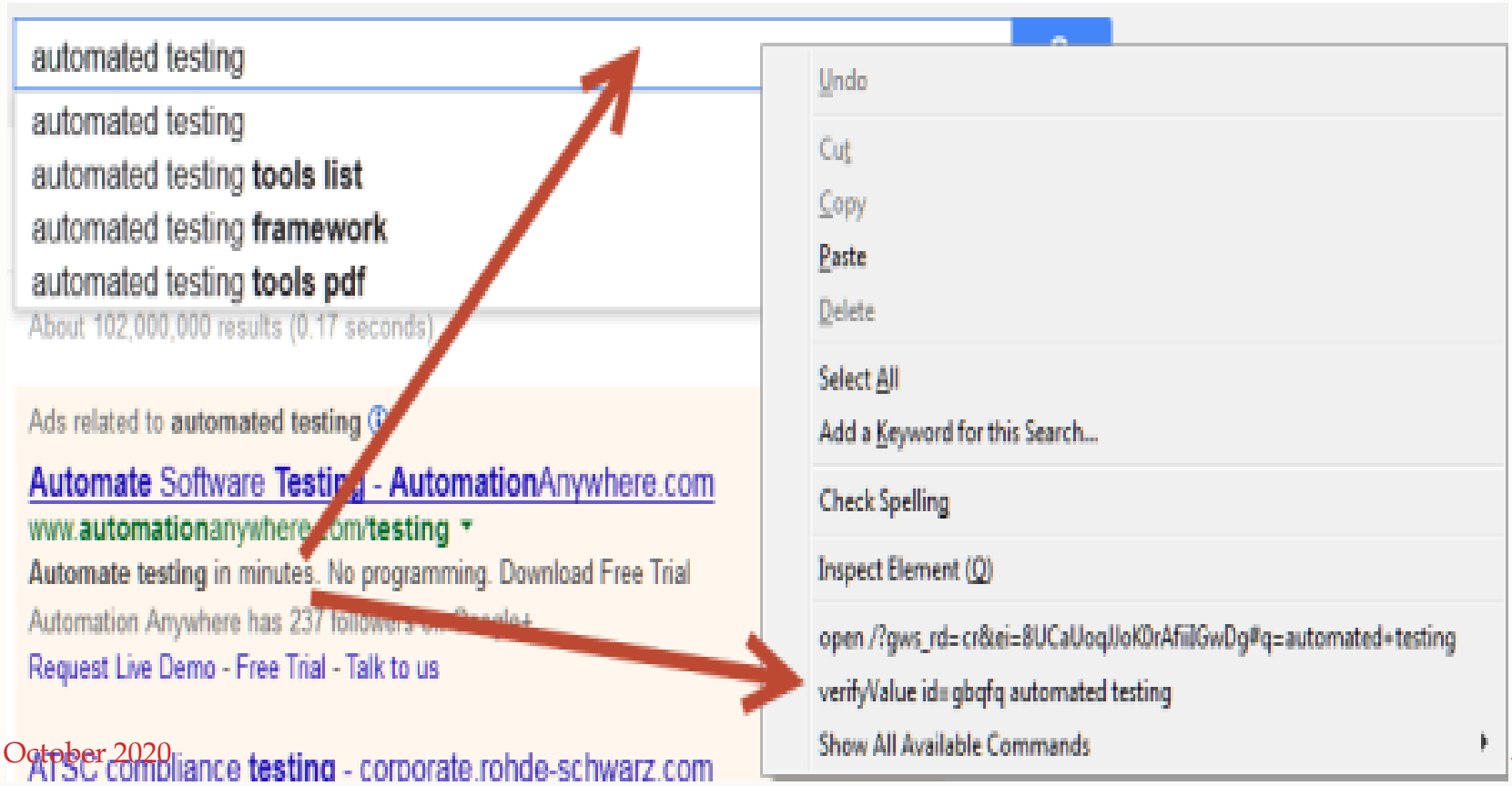
Using Selenium IDE Contextual Menu

- Besides the Selenium IDE window, the plugin also installs a contextual menu into Firefox that can be accessed by pressing the right mouse button.
- Let's see an example of how it works:

Using Selenium IDE Contextual Menu

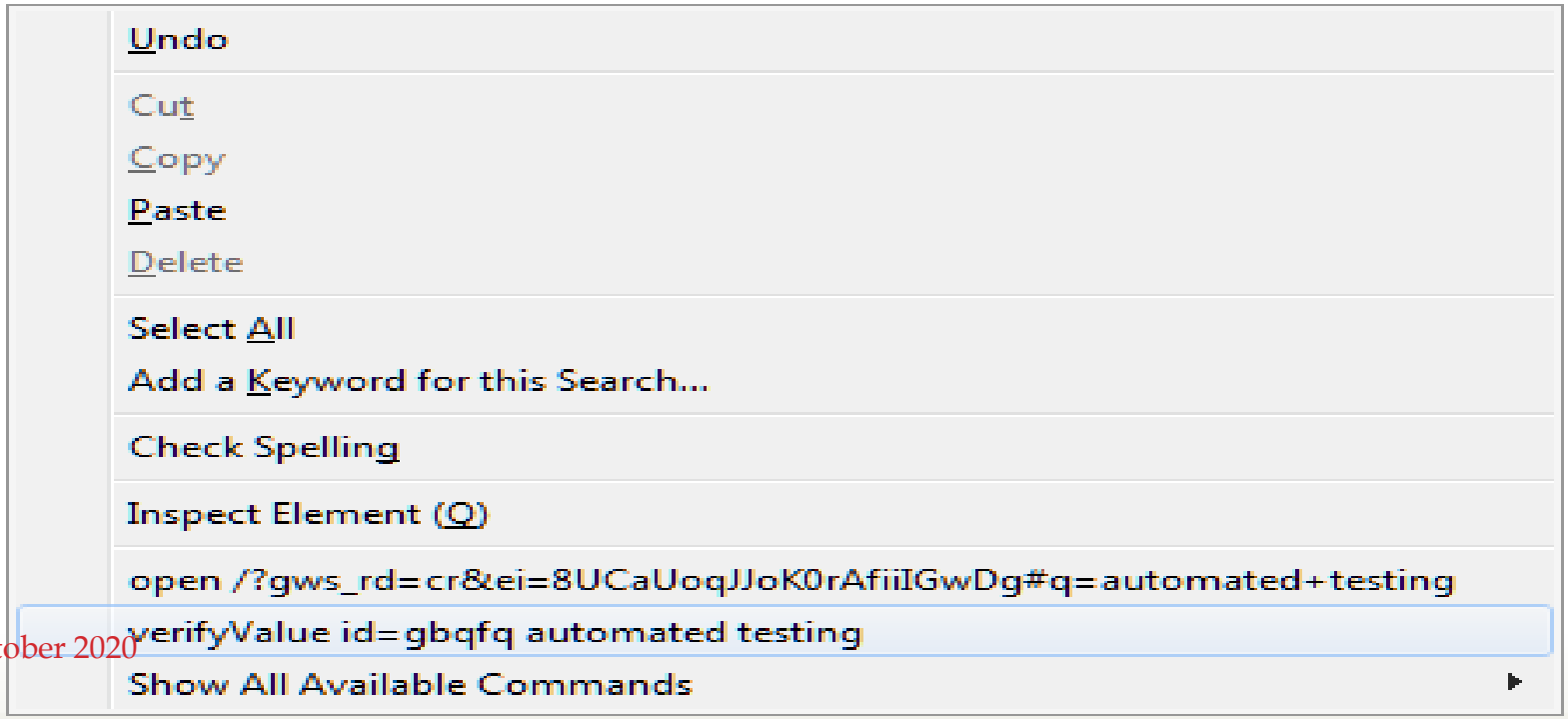
- Like the previous example, create a new test case and hit ‘record’. Then go to Google.com, type in a query (‘automated testing’) and wait for the search results page to load.
- After the results page opens, right click on the main search field on Google. A contextual menu will pop up with a list of actions you can perform.

Using Selenium IDE Contextual Menu Contd….



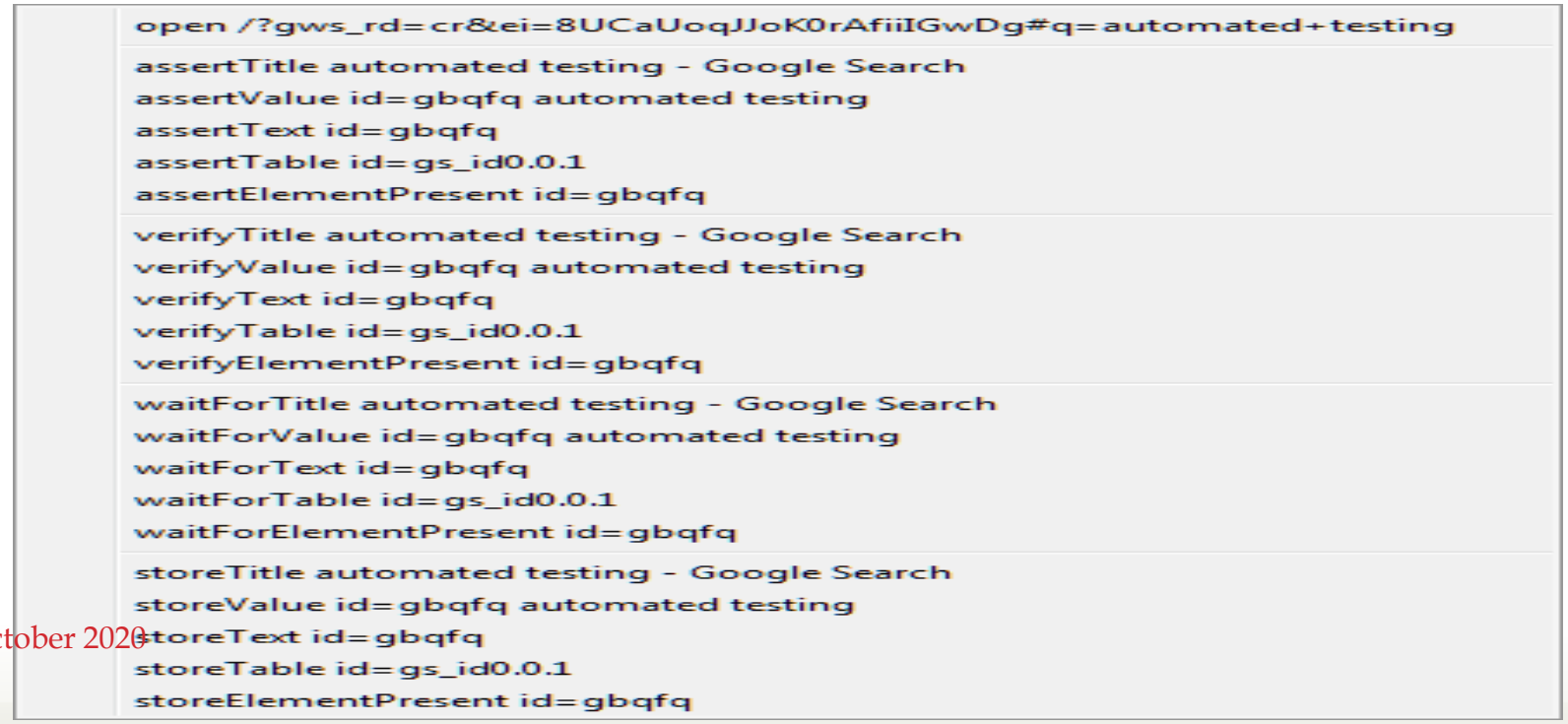
Using Selenium IDE Contextual Menu contd….

- At the bottom of the menu, you'll see a 'verifyValue' option. If you select this option, Selenium will make sure that there is always a value present in the search bar – a simple test.



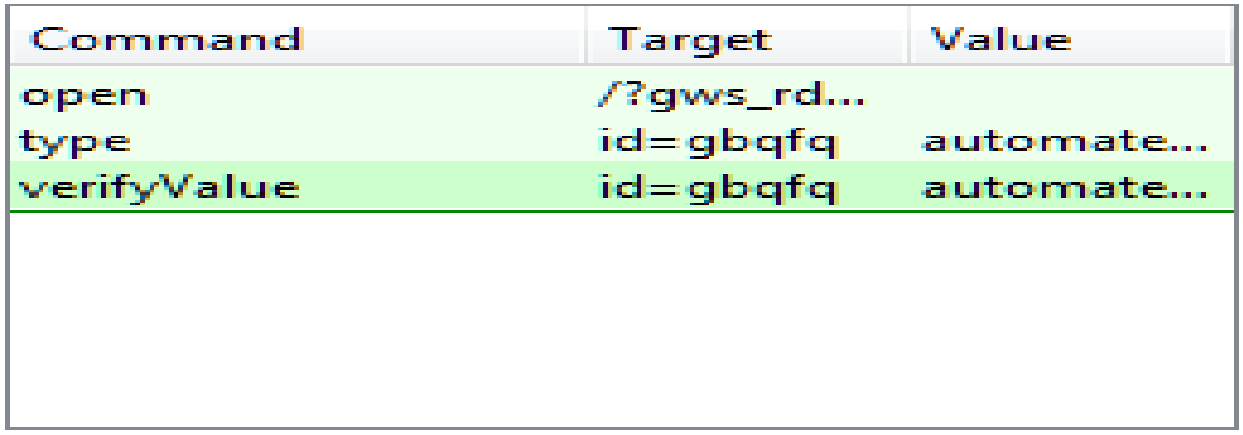
Using Selenium IDE Contextual Menu contd….

- You can also see a list of all available commands on the same menu. This will give you a huge list of commands to choose from like ‘verifyTitle’, ‘verifyValue’, ‘open’, ‘storeTitle’, etc.



Using Selenium IDE Contextual Menu contd….

- Using these different options, you can create automated tests to check different elements in a web app.
- For this example, if you stop the recording and play back the test, Selenium IDE will first open Google.com, type in the desired query, and finally, verify that the query has been entered into the search bar.



Selenium Remote Control (RC)

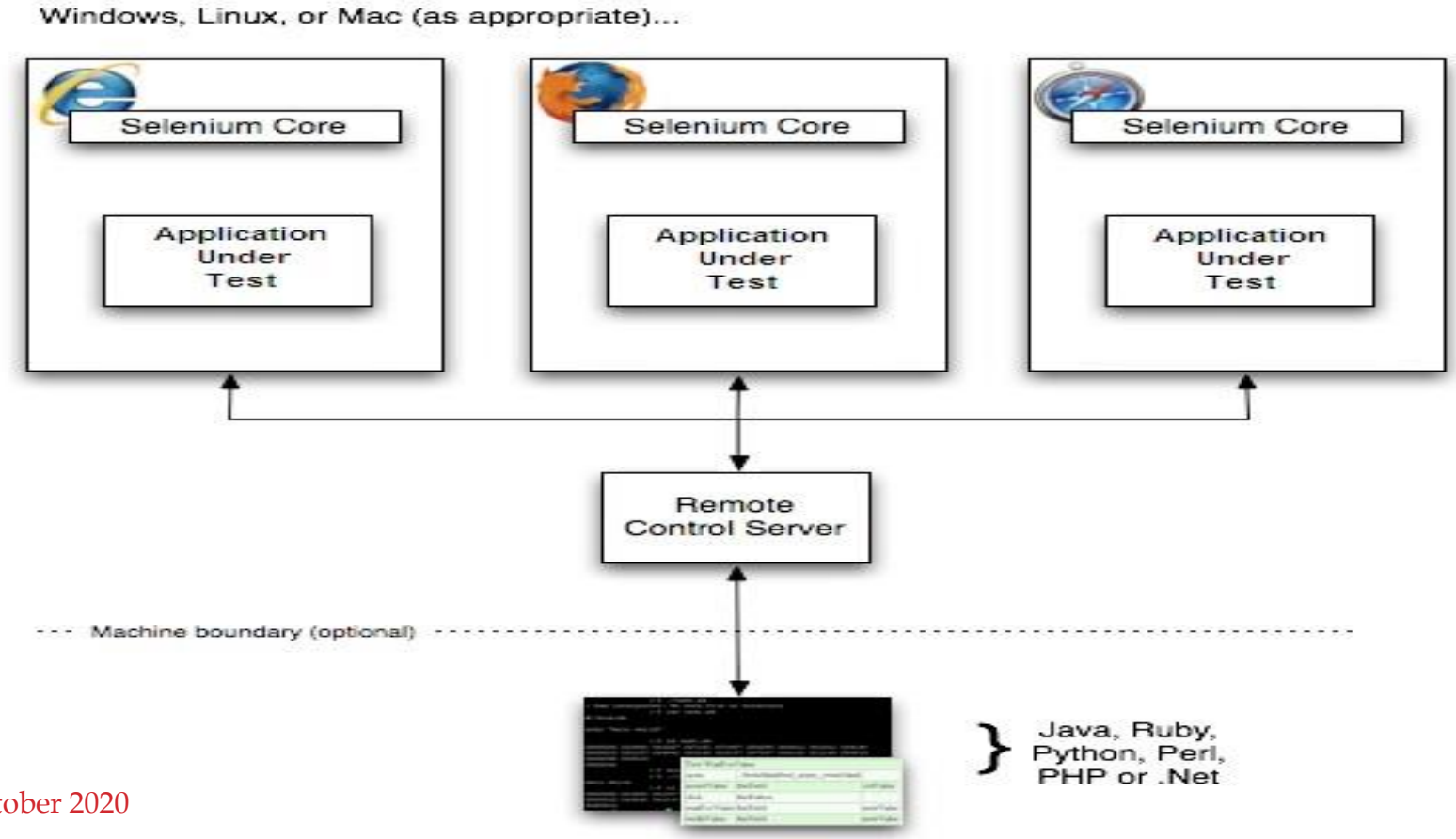
Selenium RC Components

- The Selenium Server
  - which launches and kills browsers, interprets and
  - runs the Selenese commands passed from the test program, and
    - acts as an HTTP proxy, intercepting and verifying HTTP messages passed between the browser and the AUT.

Selenium RC Components

- Client libraries
  - which provide the interface between
    - each programming language and
    - the Selenium RC Server.

Selenium RC Architecture



Selenium RC Architecture

- The client libraries communicate with the Server passing each Selenium command for execution.
- Then the server passes the Selenium command to the browser using Selenium-Core JavaScript commands.
- The browser, using its JavaScript interpreter, executes the Selenium command.
- This runs the Selenese action or verification you specified in your test script.

Selenium Server

- Selenium Server receives Selenium commands from your test program, interprets them, and reports back to your program the results of running those tests.
- The RC server bundles Selenium Core and automatically injects it into the browser.
- This occurs when your test program opens the browser (using a client library API function).

Selenium Server

- Selenium-Core is a JavaScript program, actually a set of JavaScript functions
  - which interprets and executes Selenese commands
    - using the browser's built-in JavaScript interpreter.
- The Server receives the Selenese commands from your test program
  - using simple HTTP GET/POST requests.
- This means you can use any programming language that can send HTTP requests to automate Selenium tests on the browser.

Client Libraries

- The client libraries provide the programming support that
  - allows you to run Selenium commands from a program of your own design.
- There is a different client library for each supported language.
- A Selenium client library provides
  - a programming interface (API), i.e., a set of functions,
    - which run Selenium commands from your own program.

Client Libraries

- Within each interface, there is a programming function that supports each Selenese command.
- The client library
  - takes a Selenese command and
  - passes it to the Selenium Server for processing a
    - specific action or test against the application under test (AUT).

Client Libraries

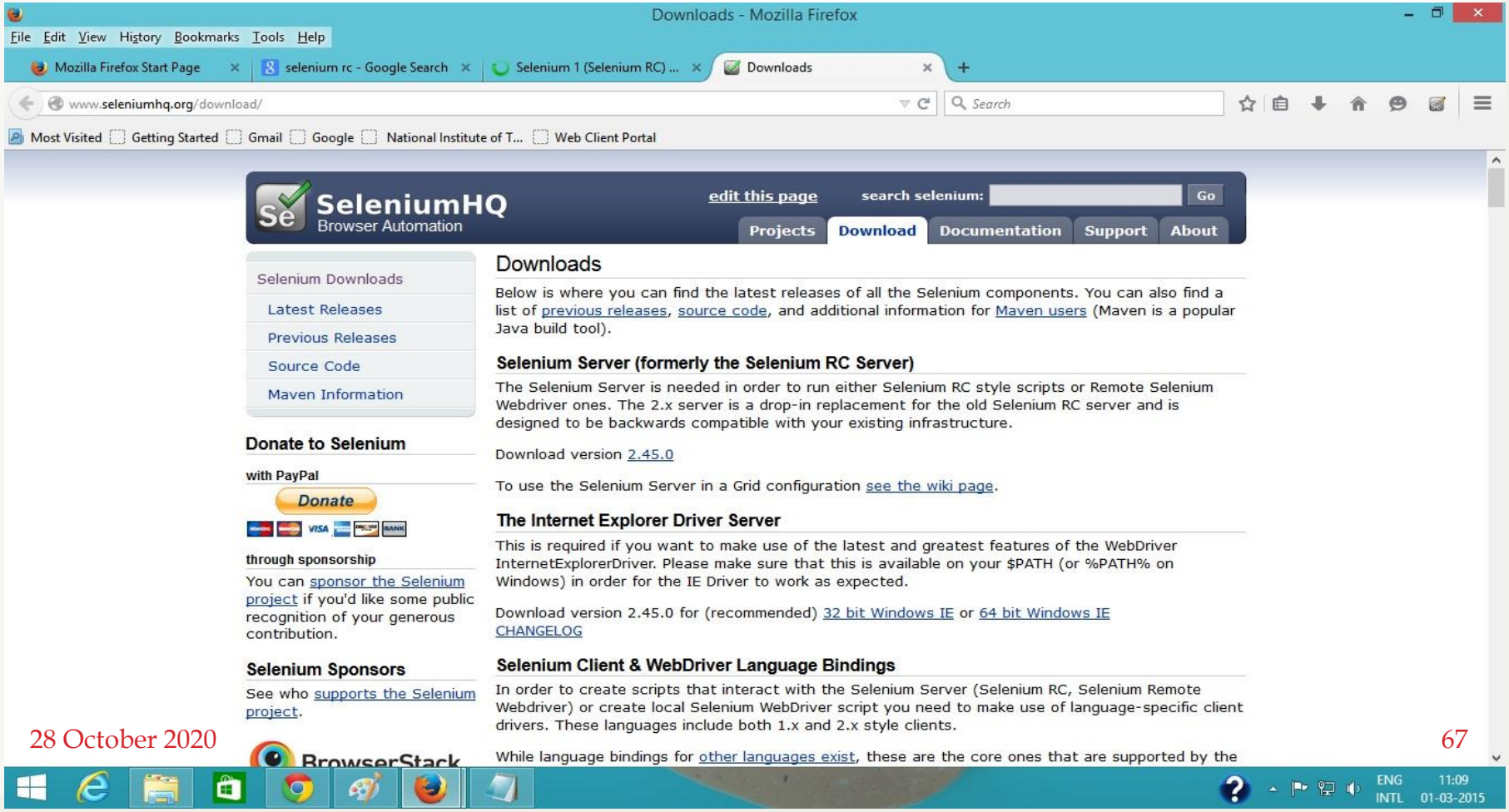
- The client library also
  - receives the result of that command and
  - passes it back to your program.
- Your program can
  - receive the result and
  - store it into a program variable and
  - report it as a success or failure, or possibly take corrective action if it was an unexpected error.

Installation steps for Selenium RC

The pre-requisites to install RC are to have Java 1.5 or above version and Eclipse IDE installed in your machine.

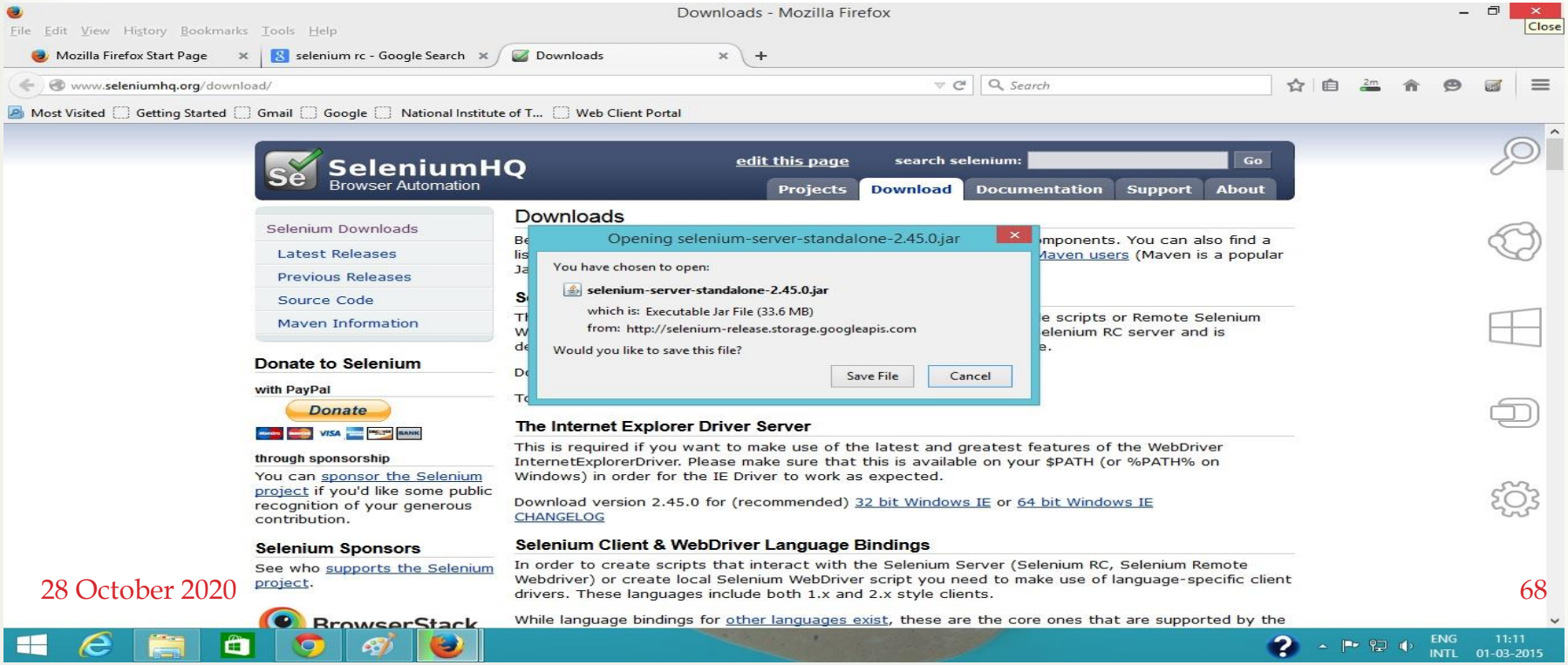
1. You can download and install Java URL from the following http://www.oracle.com/technetwork/java/javase/downloads/index.html
2. You can download and install Eclipse from the following URL:http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/R/eclipse-jee-helios-win32.zip
3. Extract the Zip Folder and run the eclipse.exe File. There is no separate installation required rather than running the .Exe file.
4. Now go to http://seleniumhq.org/download/ and click on Download under **Selenium Server** section

Installation steps for Selenium RC

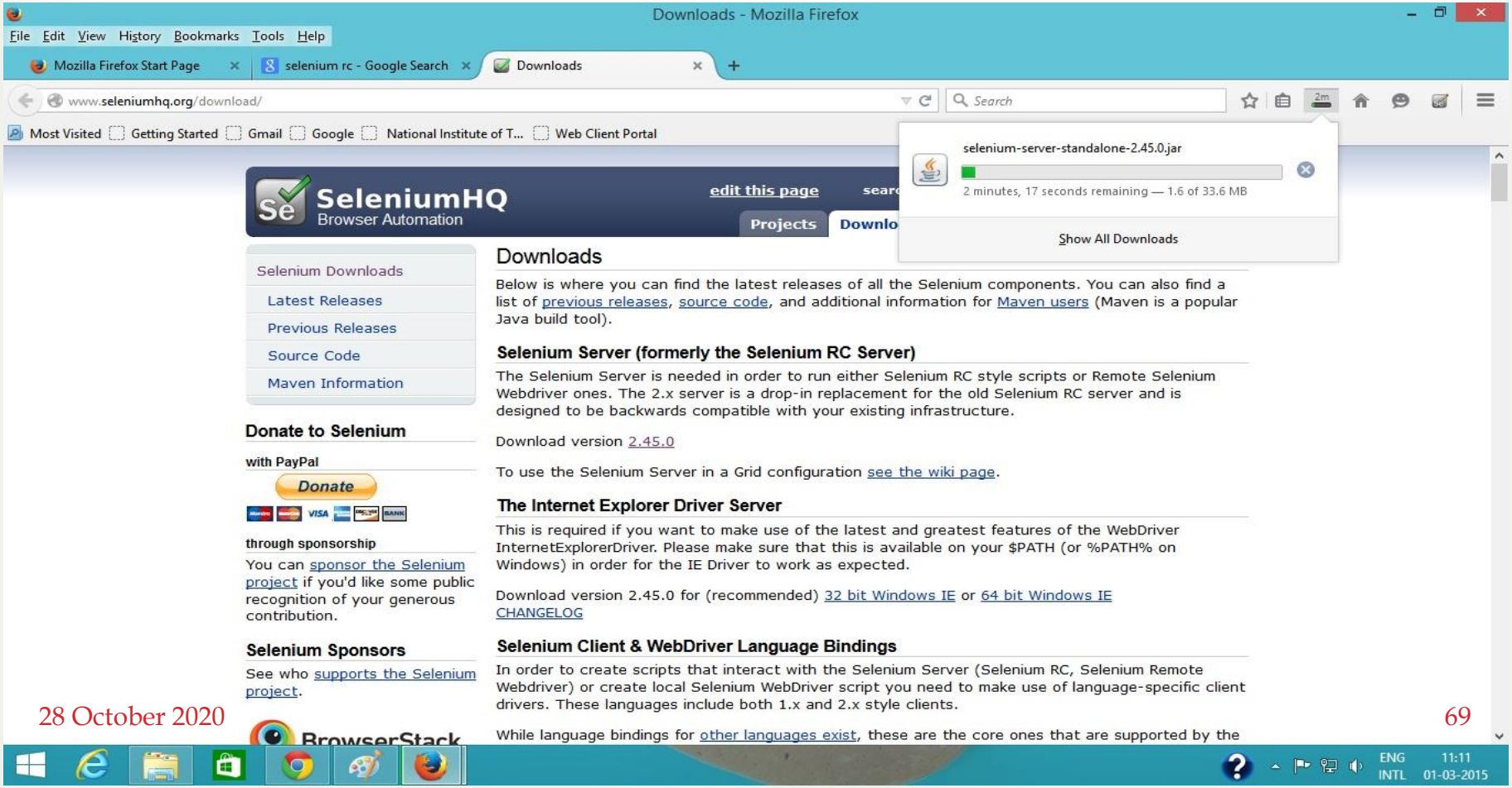


Installation steps for Selenium RC

5. Click download link under **selenium Client drivers** for your desired language. (e.g. To download Selenium Java client drivers, click on Download link for Java)



Installation steps for Selenium RC



Installation steps for Selenium RC

6. Extract the Client drivers into to your Local disk.
7. Open Eclipse IDE.
8. Create a New Project (Right Click on Project Explore-->New-->select Java Project-->Give the name and click on Finish.
9. Right click on the Project-->Build Path-->Configure Build Path--> Libraries tab-->Add External Jars.
10. Add the following list of jars to the libraries and Click on Ok to complete the setup.

    Selenium-server-standalone-2.14.0

    Selenium-java-2.14.0

    Selenium-java-2.14.0-srcs

Configuring TestNG Framework to Selenium RC

1. Go to http://testng.org/doc/download.html and download the latest version of TestNG.
2. Extract the zip file to your local disk
3. Go to Eclipse and add the testng-6.3.1.jar file to the external jar files as above.
4. Now go to Help Menu and click on **Install New Software**
5. Click on Add button.
6. Give the Name as TestNG (you can give any name, but we prefer TestNG)
7. In the location provide the following URL and click on OK http://beust.com/eclipse (For Eclipse 3.4 and above) http://beust.com/eclipse1 (For Eclipse 3.3 and below)
8. Now Check the checkbox TestNG and click on Next to install the TestNG

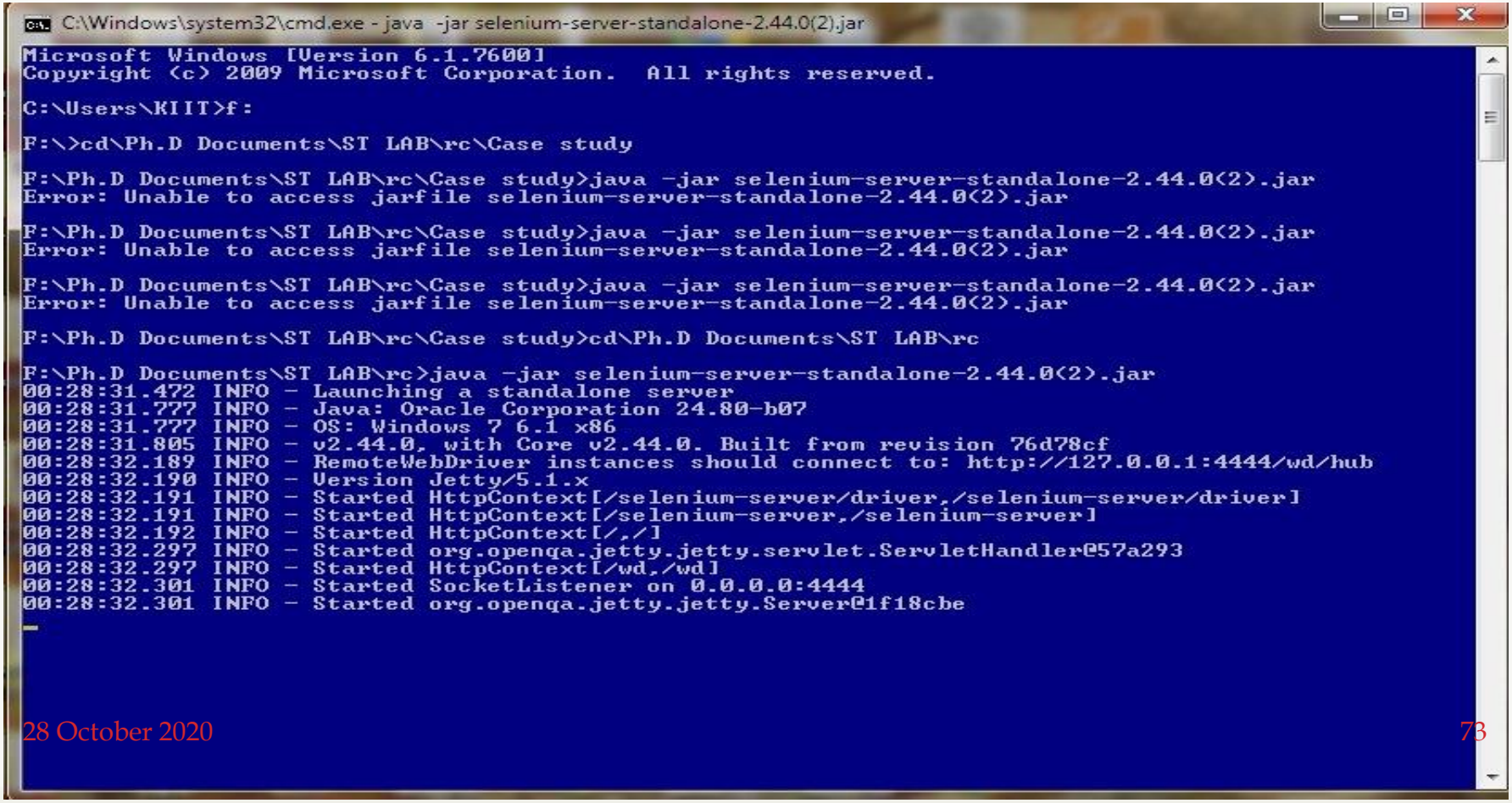
Steps to Run Selenium Server

- Before starting the execution of your scripts on selenium RC we have to start the selenium server. Following are the steps to start your server.

1. Open command prompt.
2. Type cmd.
3. Go to the location where your selenium-standalone-server2.14.0.jar resided.
4. Type the following command at command prompt to start the server.

java –jar selenium-standalone-server-2.14.0.jar

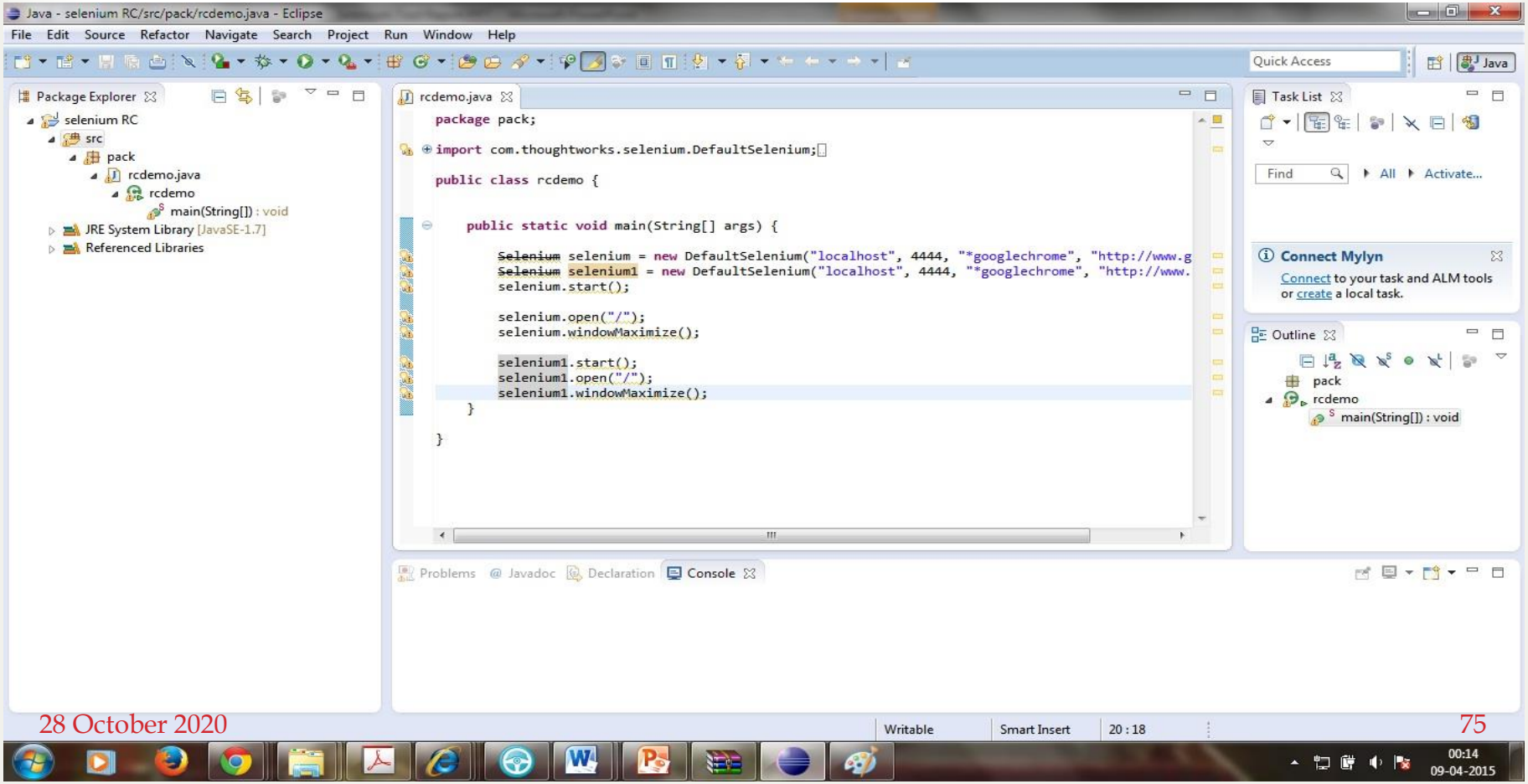
Creating a Batch File for Selenium server



Implementation of Selenium RC

1. Create a new project as Selenium RC in eclipse.
2. Right click on src → New → class→ give name as rcdemo and package name as pack.
3. Write the code.

Implementation of Selenium RC contd….

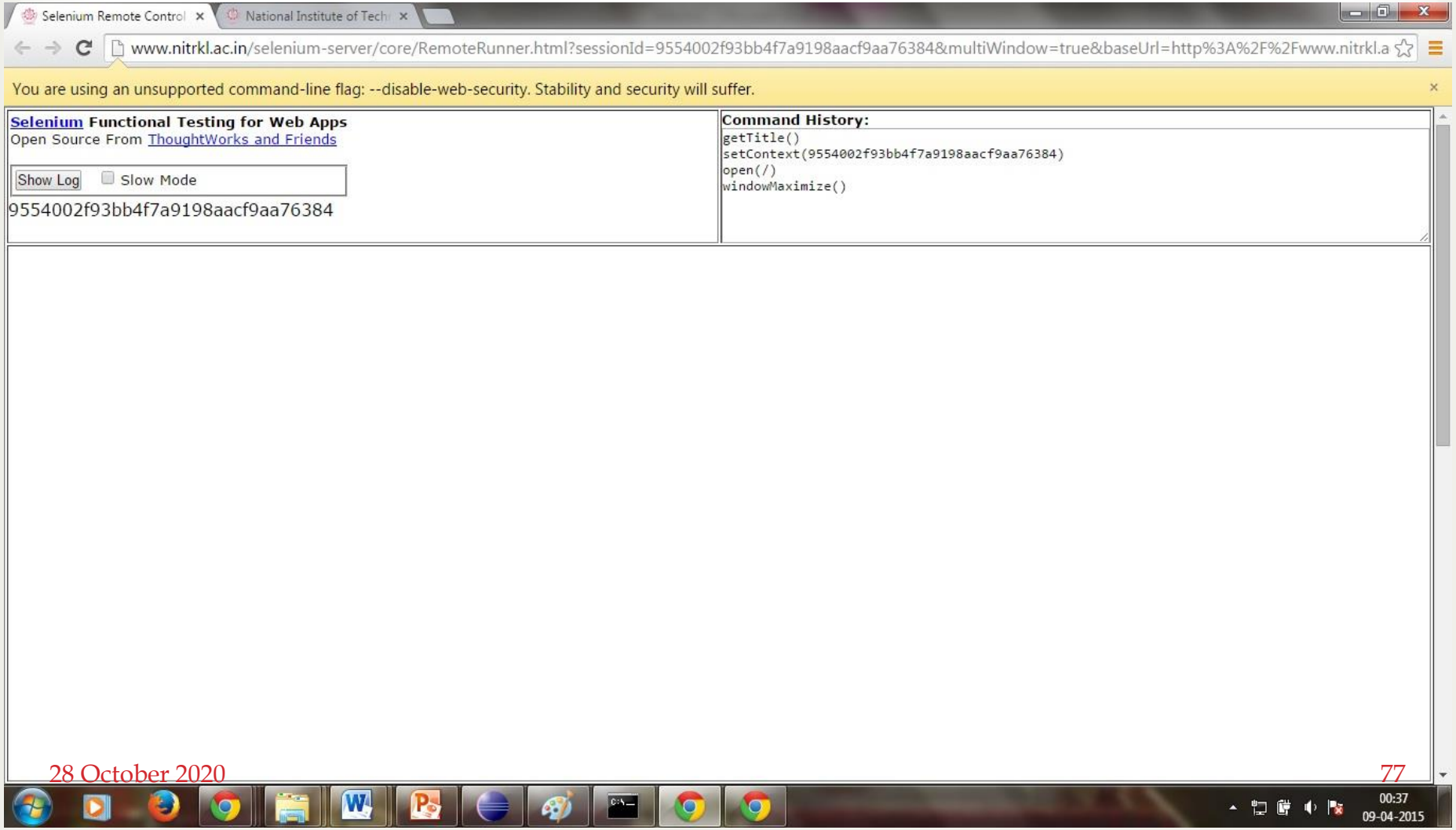


Implementation of Selenium RC contd….

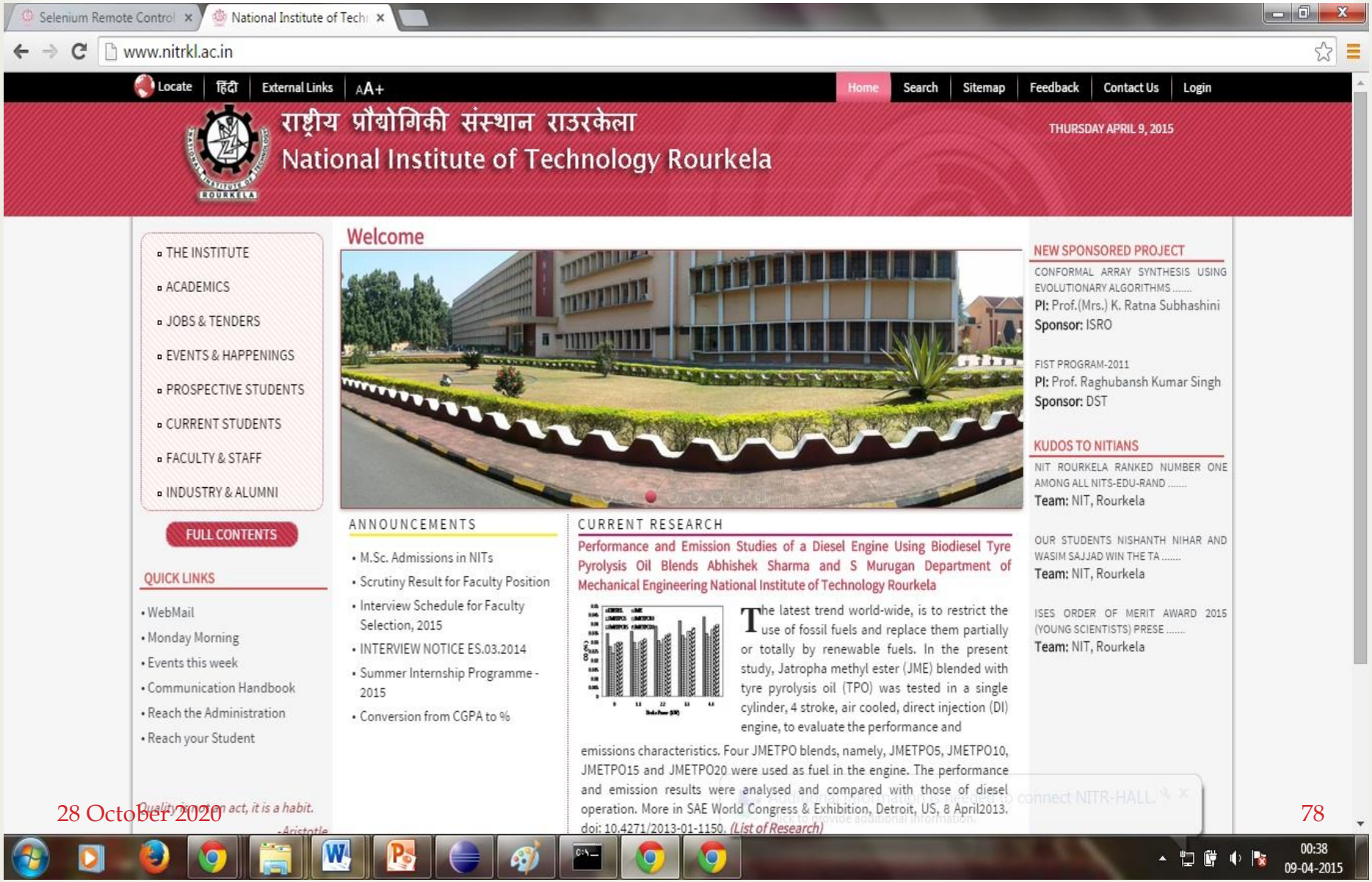
```
package pack;
import com.thoughtworks.selenium.DefaultSelenium;
import com.thoughtworks.selenium.Selenium;
public class rcdemo {
        public static void main(String[] args) {
                Selenium selenium = new DefaultSelenium("localhost", 4444,
"*googlechrome", "http://www.nitrkl.ac.in");
                Selenium selenium1 = new DefaultSelenium("localhost", 4444,
"*googlechrome", "http://www.gmail.com");
                selenium.start();
                selenium.open("/");
                selenium.windowMaximize();
                selenium1.start();
                selenium1.open("/");
                selenium1.windowMaximize();
        }
}
```

Implementation of Selenium RC contd….

Then both the given websites will open.



Implementation of Selenium RC contd….



Selenium Webdriver

Introduction

- Selenium WebDriver fits in the same role as RC did, and has incorporated the original 1.x bindings.
- It refers to both the language bindings and the implementations of the individual browser controlling code.
- This is commonly referred to as just "WebDriver" or sometimes as Selenium 2.

  Selenium 1.0 + WebDriver = Selenium 2.0

Introduction

- Selenium-WebDriver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded.
- WebDriver's goal is to supply a well-designed object-oriented API that provides improved support for modern advanced web-app testing problems.
- Selenium-WebDriver makes direct calls to the browser using each browser's native support for automation.

Introduction

- WebDriver is a tool for automating web application testing, and in particular to verify that they work as expected.
- It aims to provide a friendly API that's easy to explore and understand, easier to use than the Selenium-RC (1.0) API, which will help to make your tests easier to read and maintain.
- It's not tied to any particular test framework, so it can be used equally well in a unit testing or from a plain old "main" method.

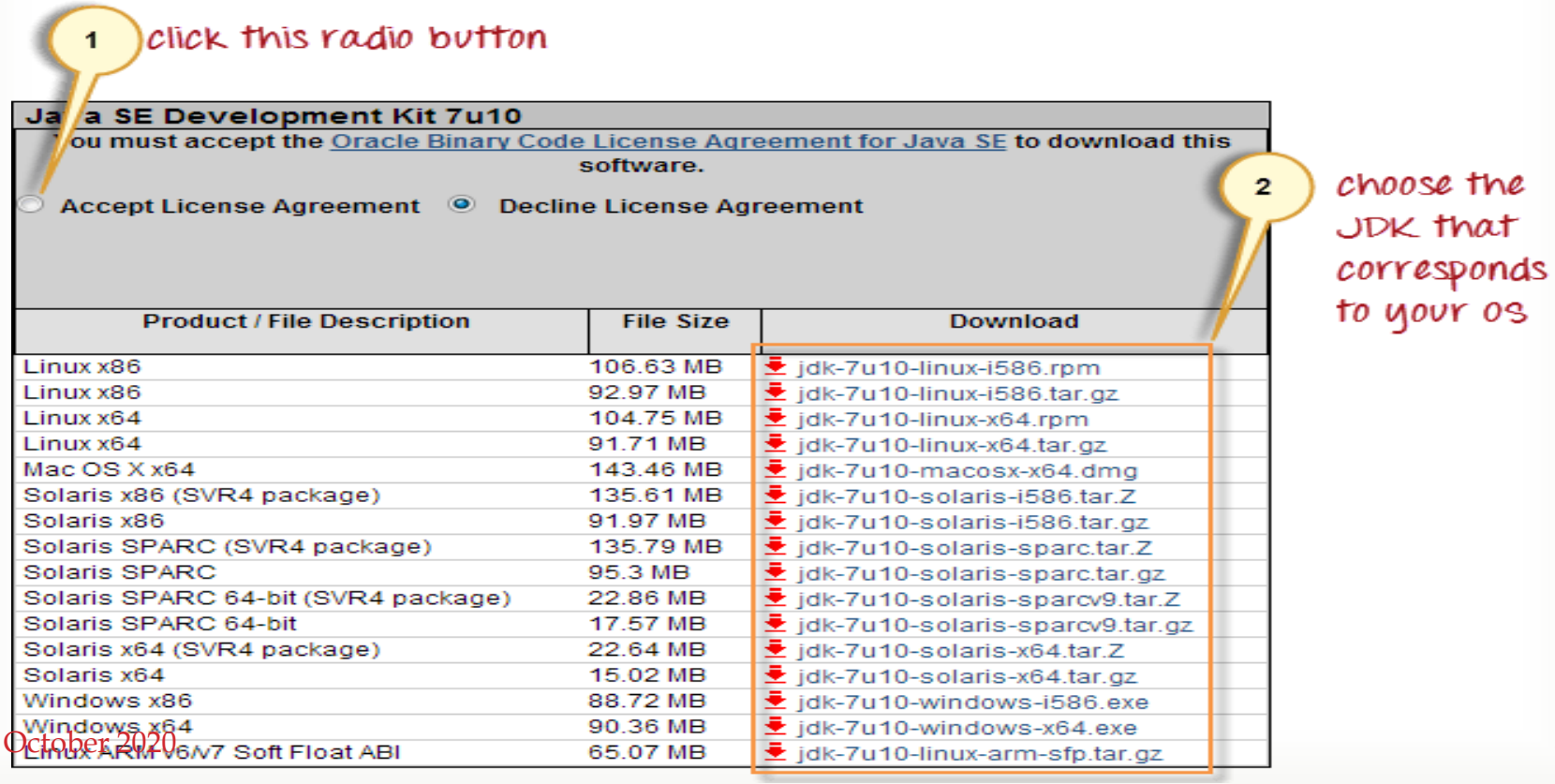
Installation of Selenium WebDriver

- **Step 1** - Install Java on your computer

  Download and install the **Java Software Development Kit (JDK)**.

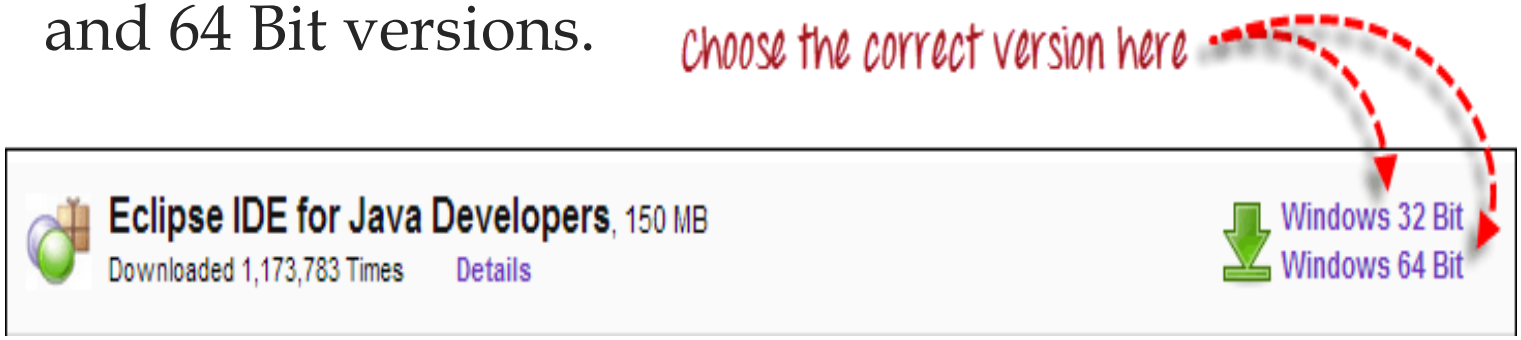
Installation of Selenium WebDriver contd….

- Next –



Installation of Selenium WebDriver contd….

- This JDK version comes bundled with Java Runtime Environment (JRE) so you do not need to download and install the JRE separately.
- **Step 2** - Install Eclipse IDE
- Download "**Eclipse IDE for Java Developers**" . Be sure to choose correctly between Windows 32 Bit and 64 Bit versions.



Installation of Selenium WebDriver contd….

- You should be able to download a ZIP file named "eclipse-java-juno-SR1-win32-x86_64.zip" (the version number “SR1” may change over time).

Installation of Selenium WebDriver contd….

- Inside that ZIP file, there is an "eclipse" folder which contains all the application files. You can extract the "eclipse" folder anywhere you want in your PC; but for this tutorial, extract it to your C drive.

- Unlike other popular software, no installation is required to use eclipse.

Installation of Selenium WebDriver contd….

- **Step 3** - Download the Selenium Java Client Driver from **http://www.seleniumhq.org/download/**

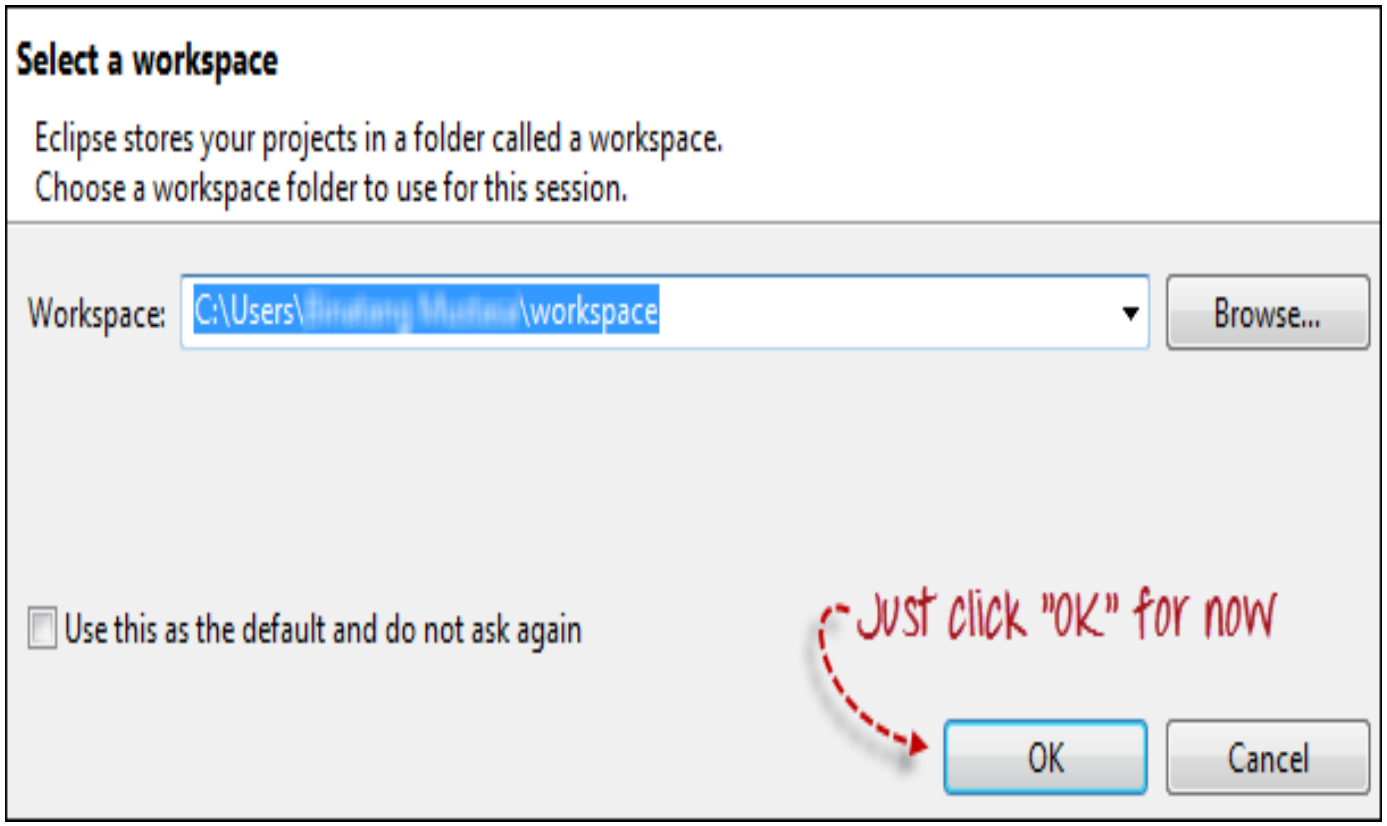
You will find client drivers for other languages there, but only choose the one for Java.

Installation of Selenium WebDriver contd….

- This download comes as a ZIP file named "selenium-2.25.0.zip".
- For simplicity, extract the contents of this ZIP file on your C drive so that you would have the directory “C:\selenium-2.25.0\”.
- This directory contains all the JAR files that we would later import on Eclipse.
- **Step 4** - Configure Eclipse IDE with WebDriver

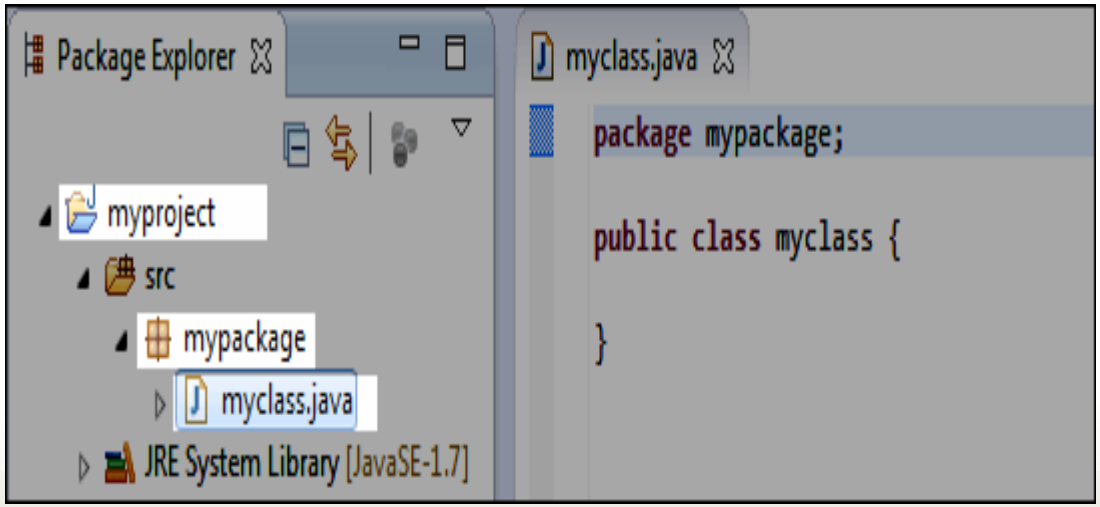
1. Launch the “eclipse.exe” file inside the “eclipse” folder that we extracted in Step 2.
2. If you followed Step 2 correctly, the executable should be located on C:\eclipse\eclipse.exe.
3. When asked to select for a workspace, just accept the default location.

Installation of Selenium WebDriver contd….



Installation of Selenium WebDriver contd….

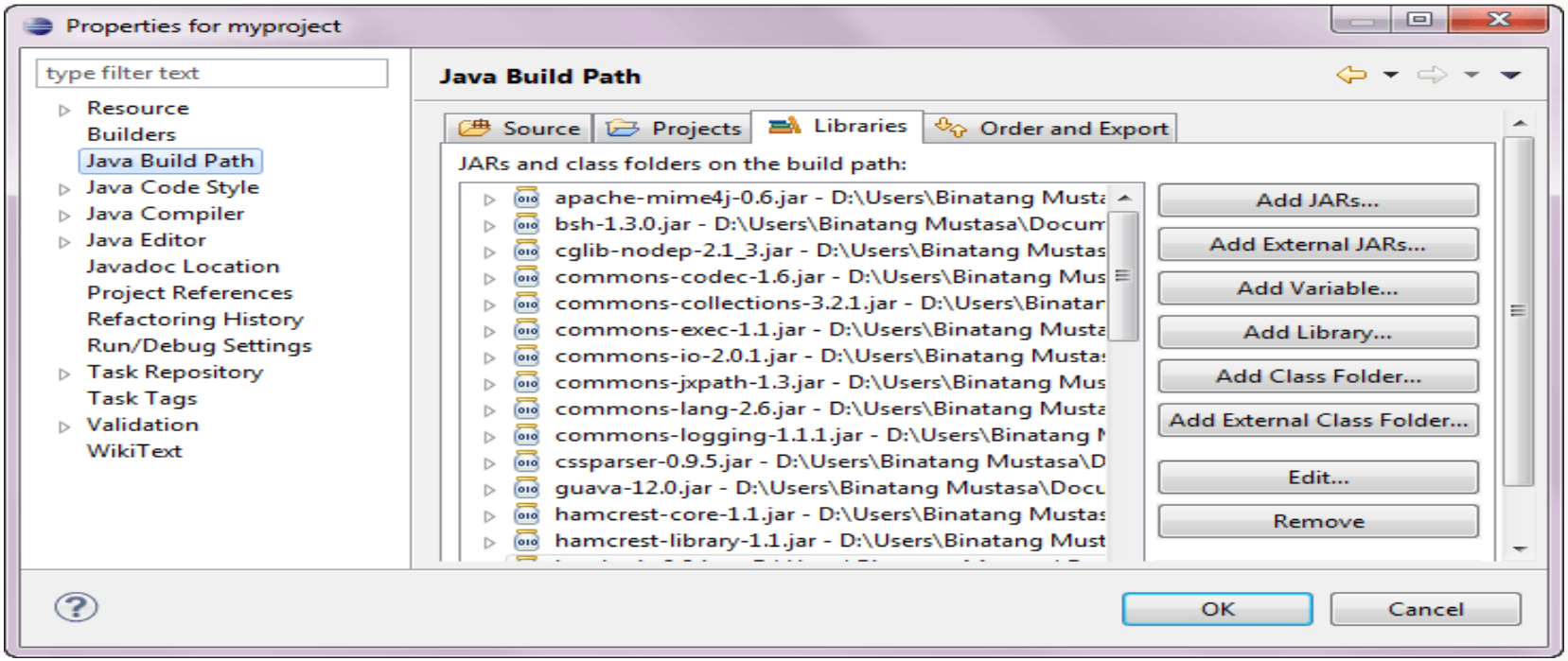
4. Create a new project through File > New > Java Project.
5. Name the project as “myproject”.
6. Right-click on the newly created project and select New > Package.
7. Name that package as "mypackage".
8. Create a new Java class under *mypackage* by right-clicking on it and then selecting New > Class, and then name it as "myclass".
9. Your Eclipse IDE should look like the image below.



Installation of Selenium WebDriver contd….

10. Right-click on *myproject* and select **Properties**.
11. On the Properties dialog, click on "Java Build Path".
12. Click on the **Libraries** tab, and then click "Add External JARs.."
13. Navigate to C:\selenium-2.25.0\ (or any other location where you saved the extracted contents of "selenium-2.25.0.zip" in Step 3).
14. Add all the JAR files inside and outside the "libs" folder.
15. Your Properties dialog should now look similar to the image below.

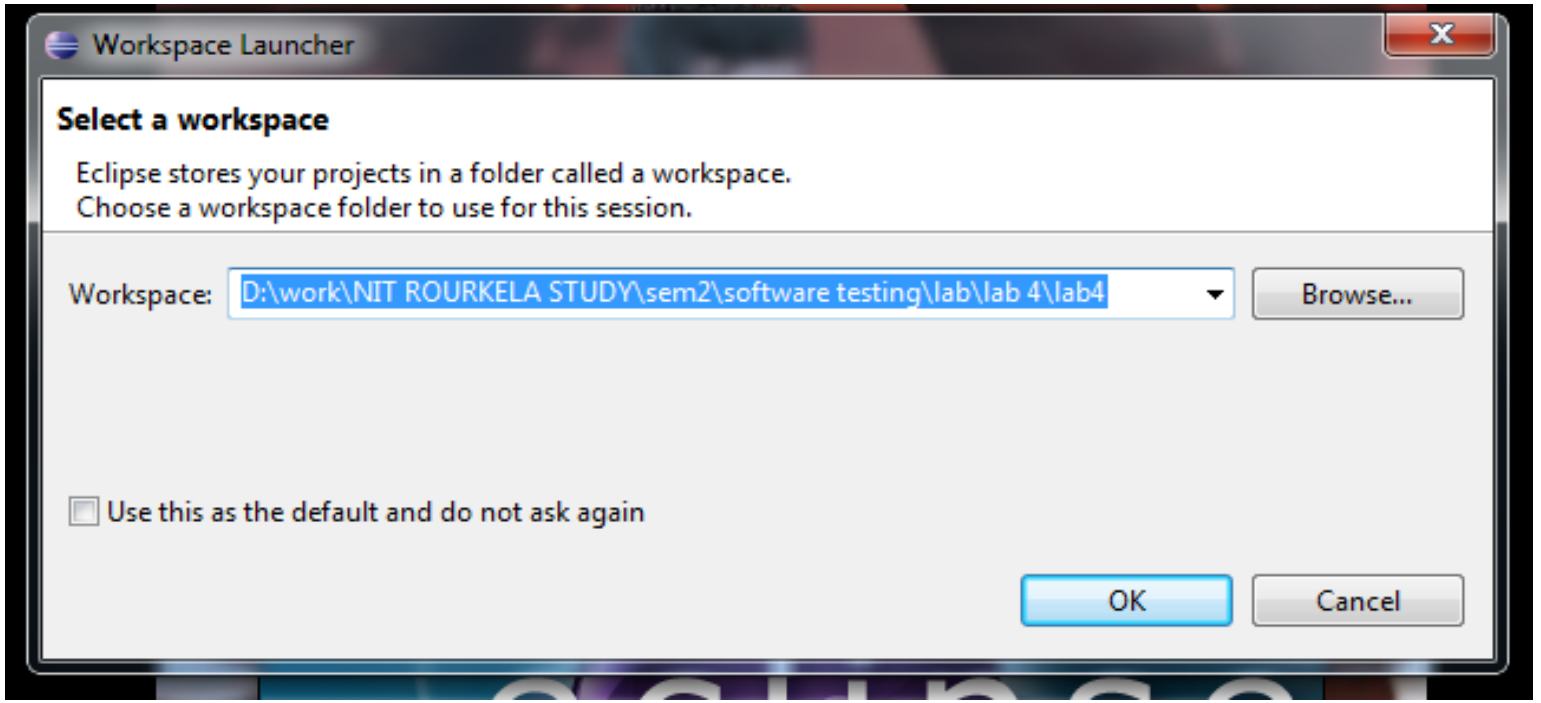
Installation of Selenium WebDriver contd….



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

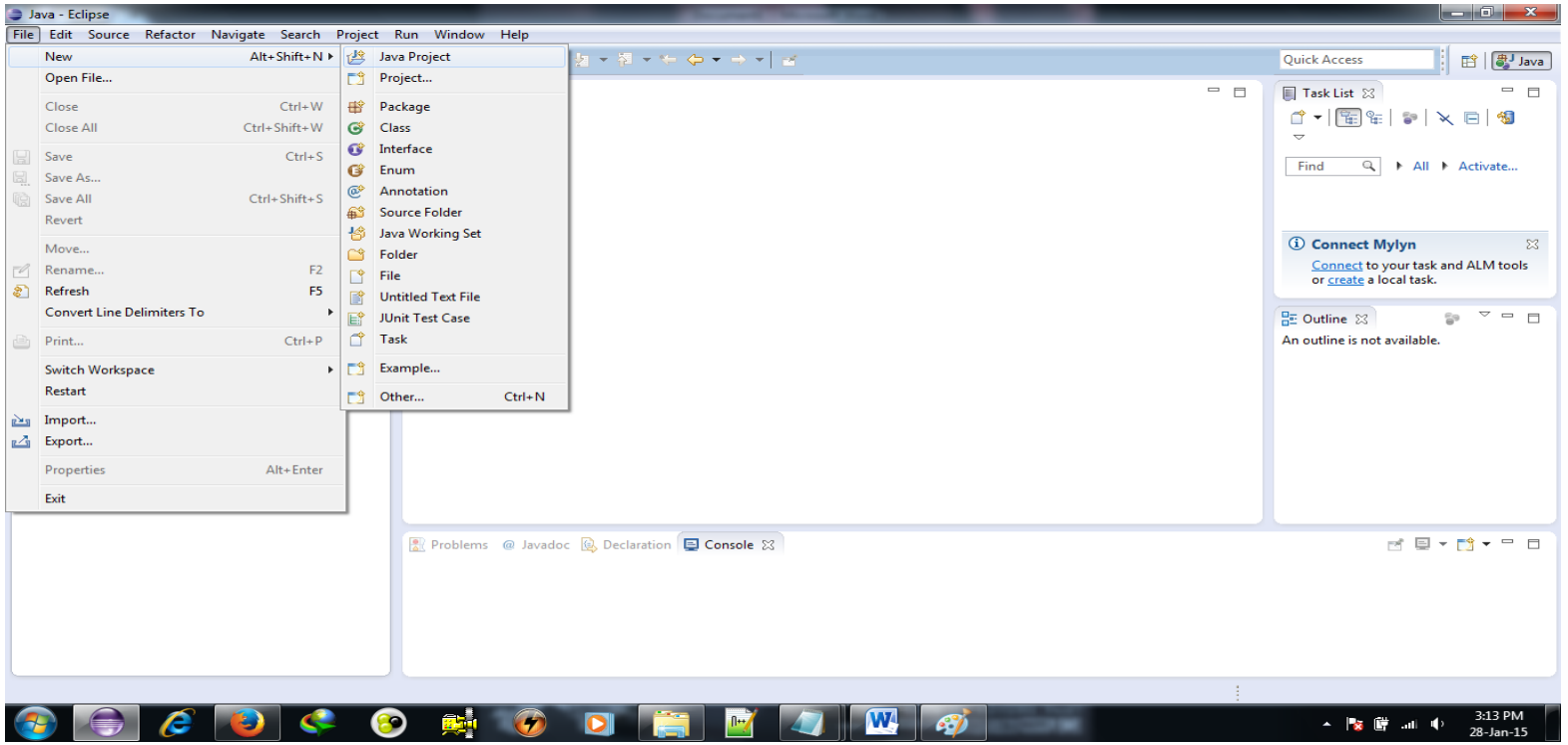
Implementation of Selenium Webdriver

- Open Eclipse.



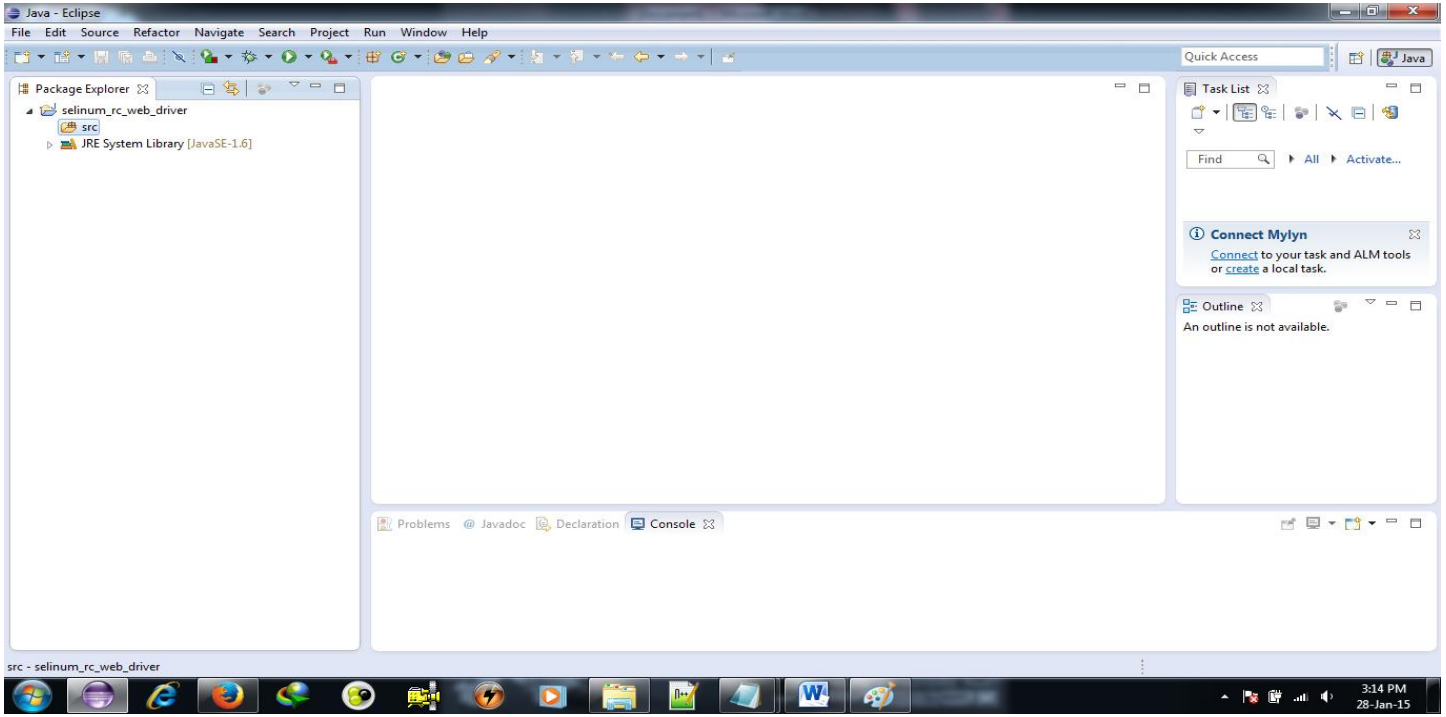
Implementation of Selenium Webdriver contd….

- Go to new->JAVA project.



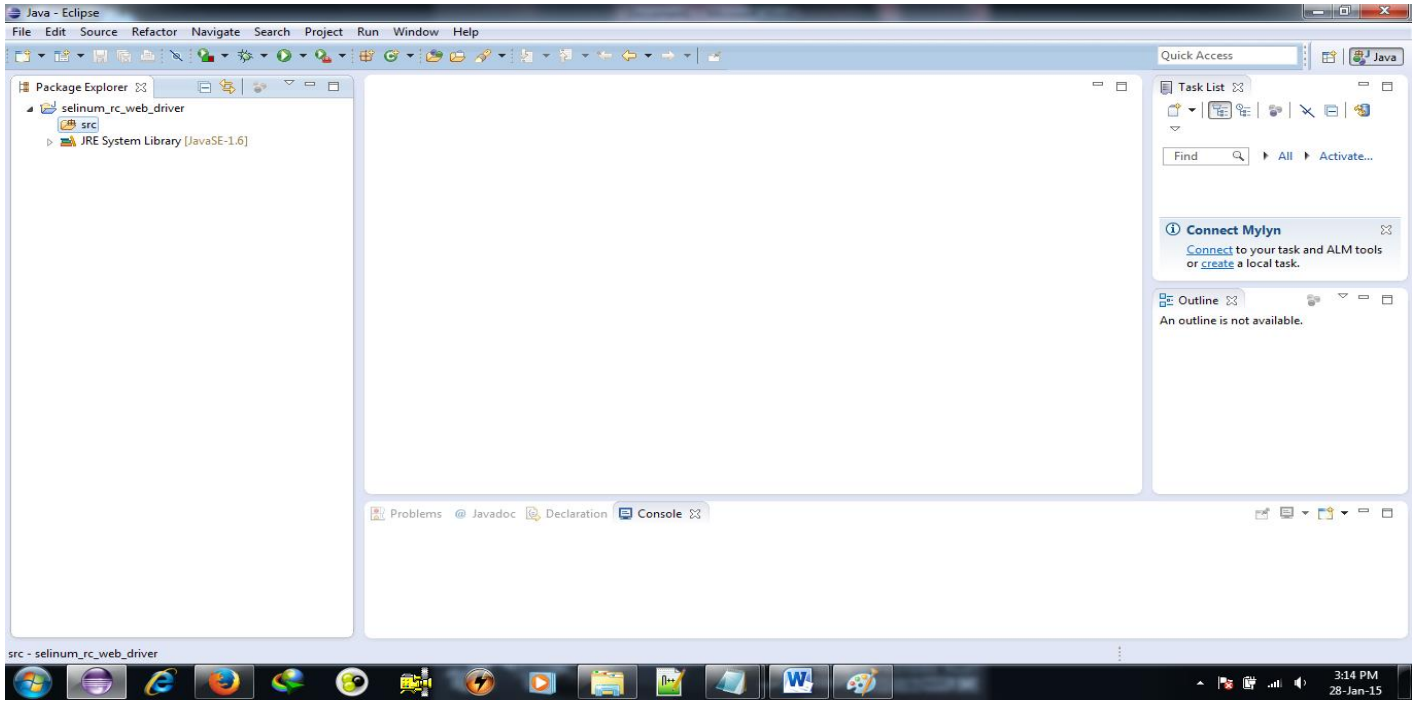
Implementation of Selenium Webdriver contd….

- Create a new project.



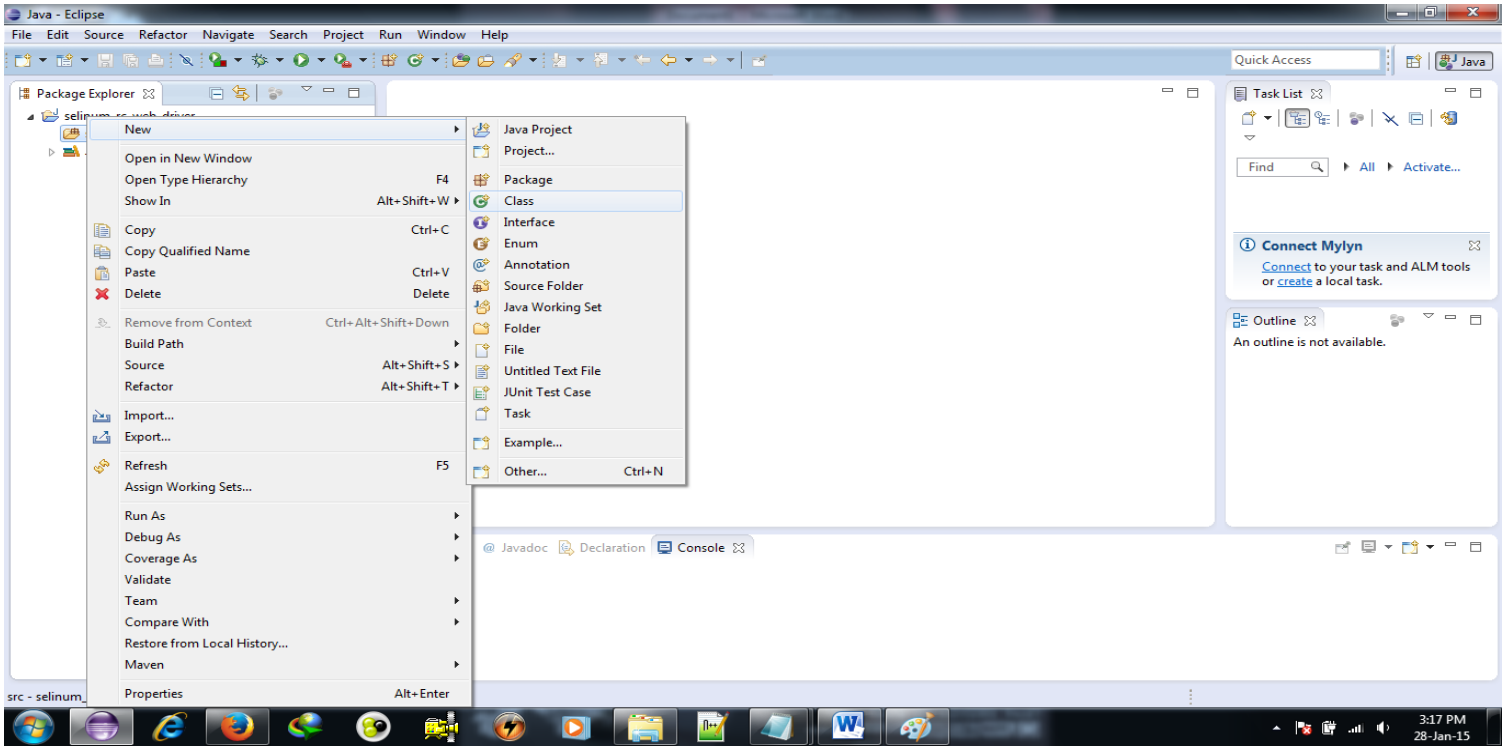
Implementation of Selenium Webdriver contd….

- New project will be created.



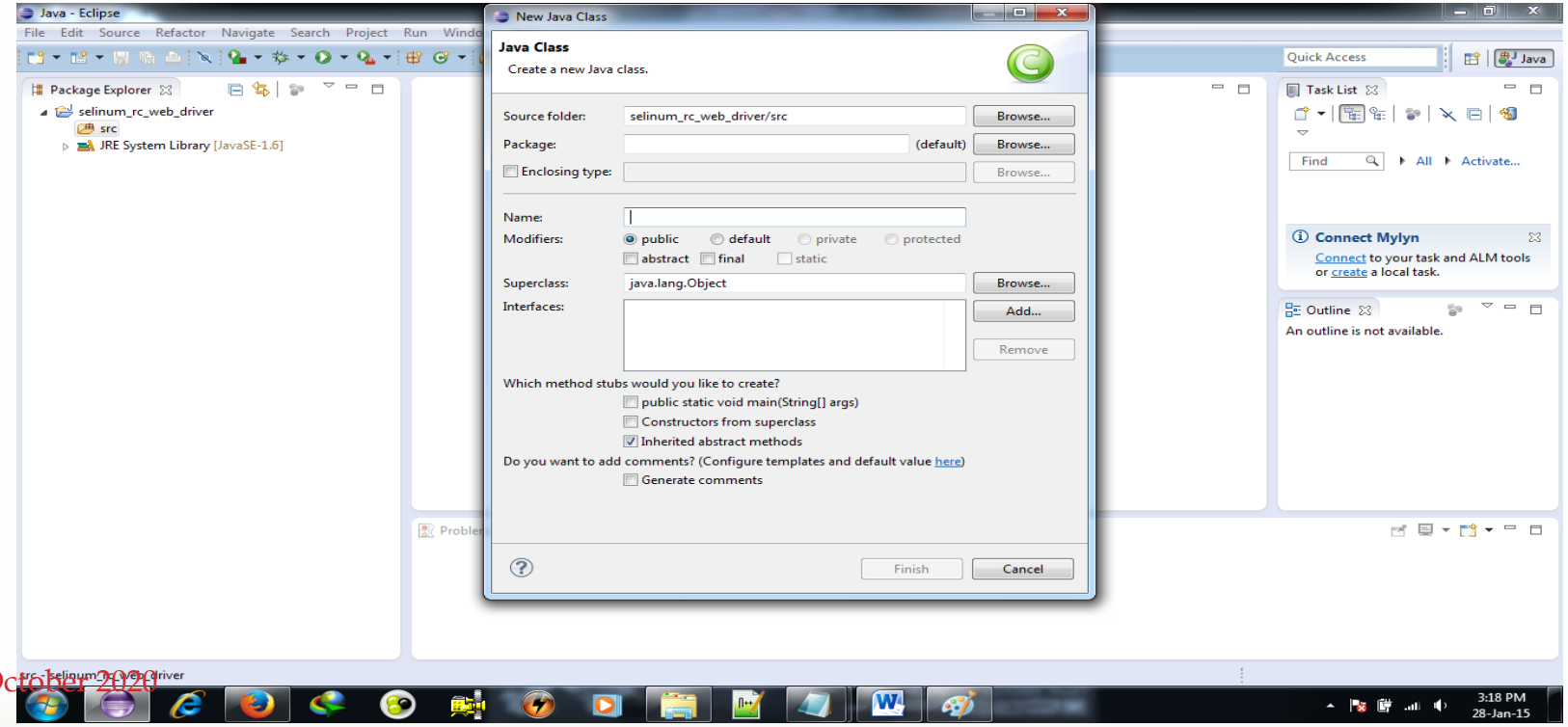
Implementation of Selenium Webdriver contd….

- Create new java class.

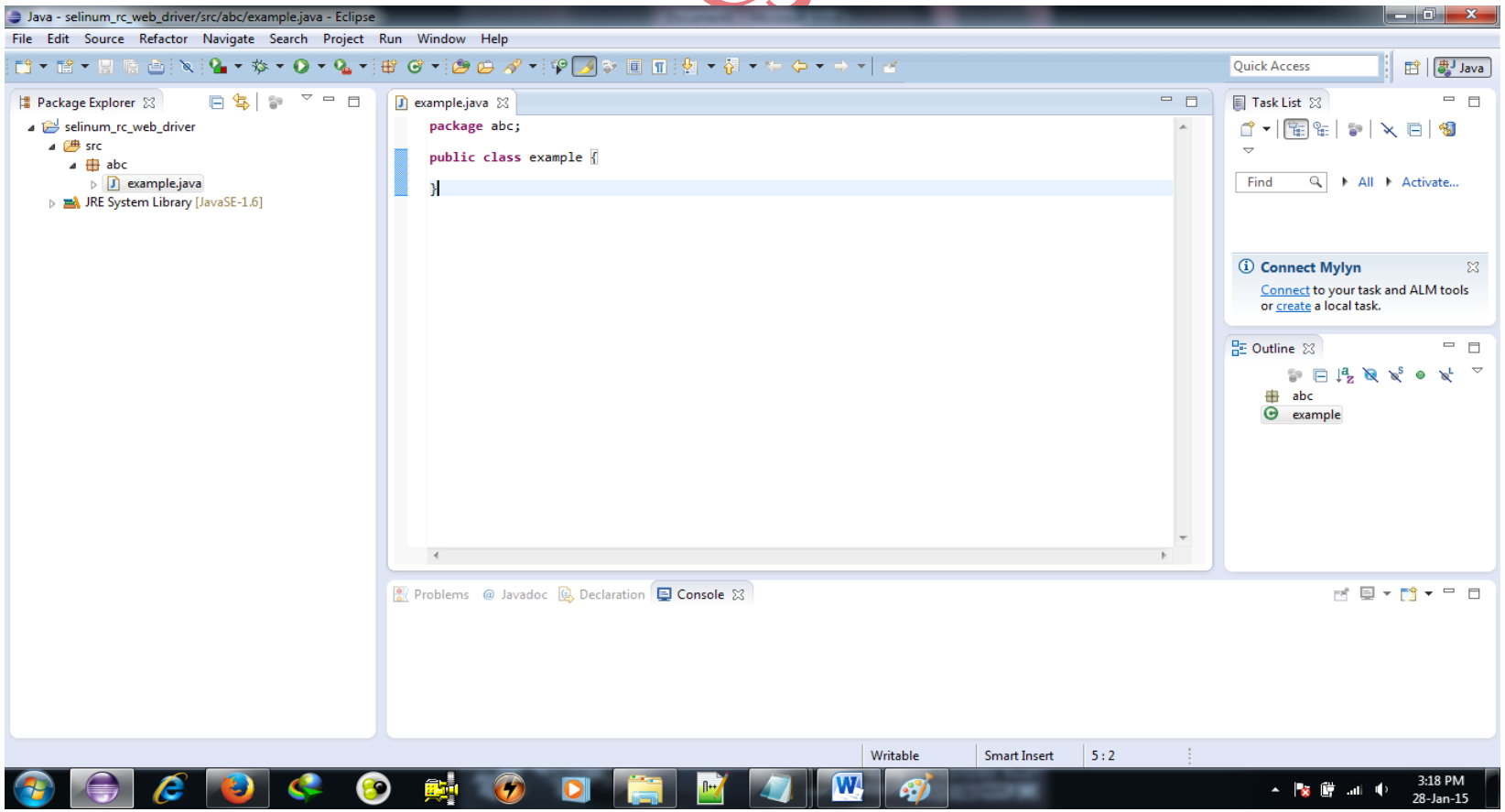


Implementation of Selenium Webdriver contd….

- Enter java class name and package name and then click finish.



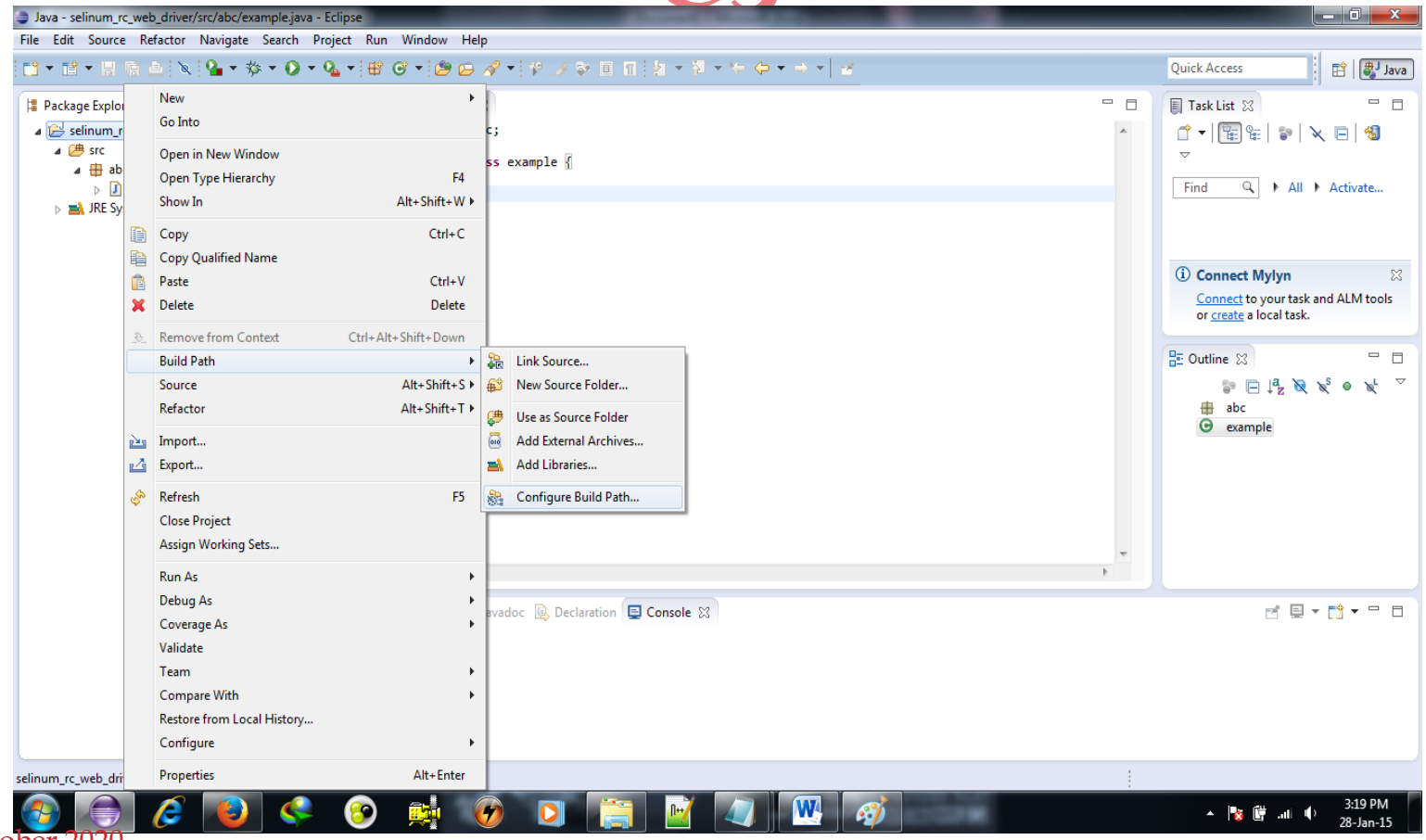
Implementation of Selenium Webdriver contd….



Implementation of Selenium Webdriver contd….

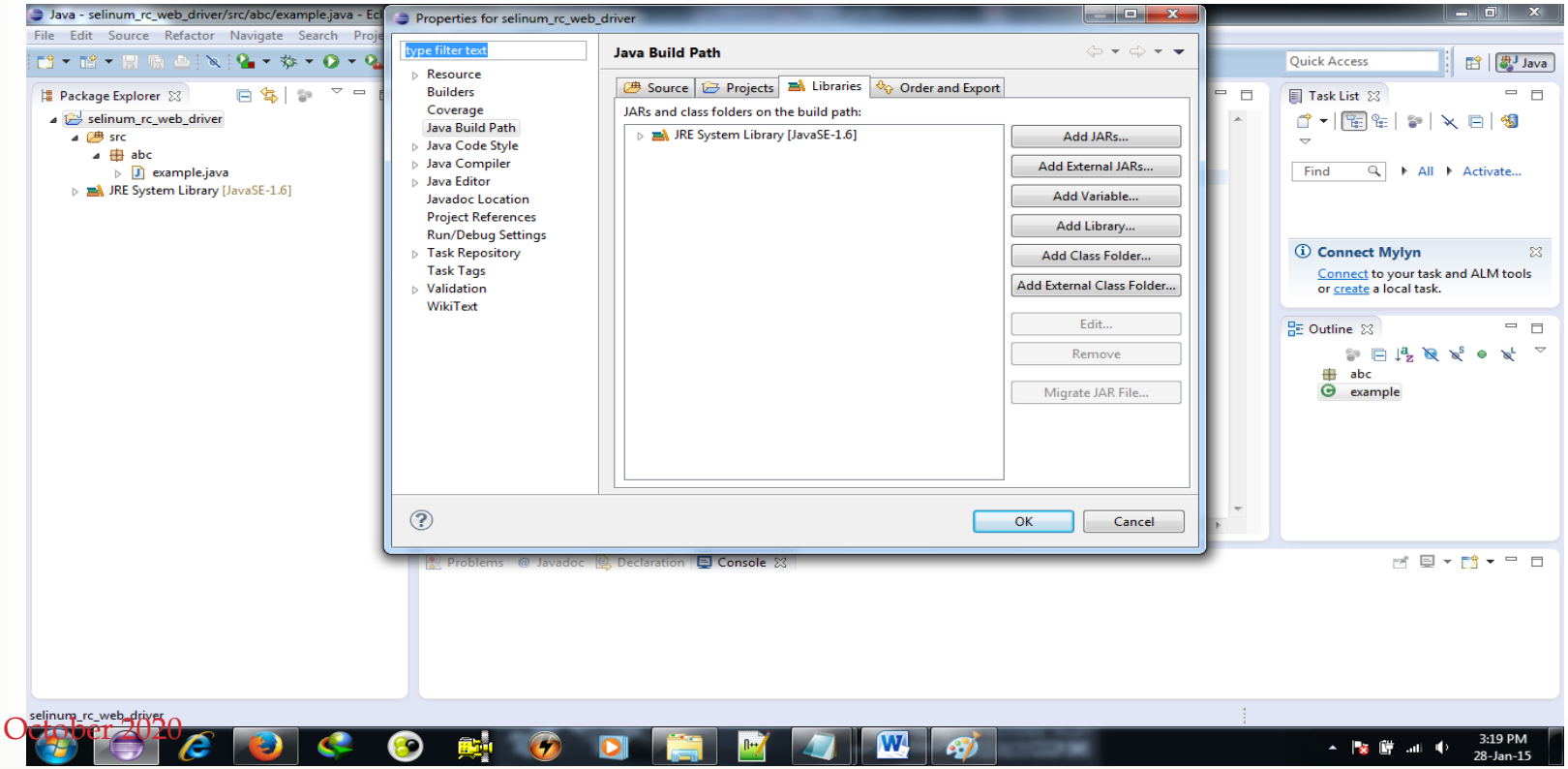
- Now we have to add different library files for our project. For adding different library files the step are
- Right click on project→build path→configure bulid path

Implementation of Selenium Webdriver contd….



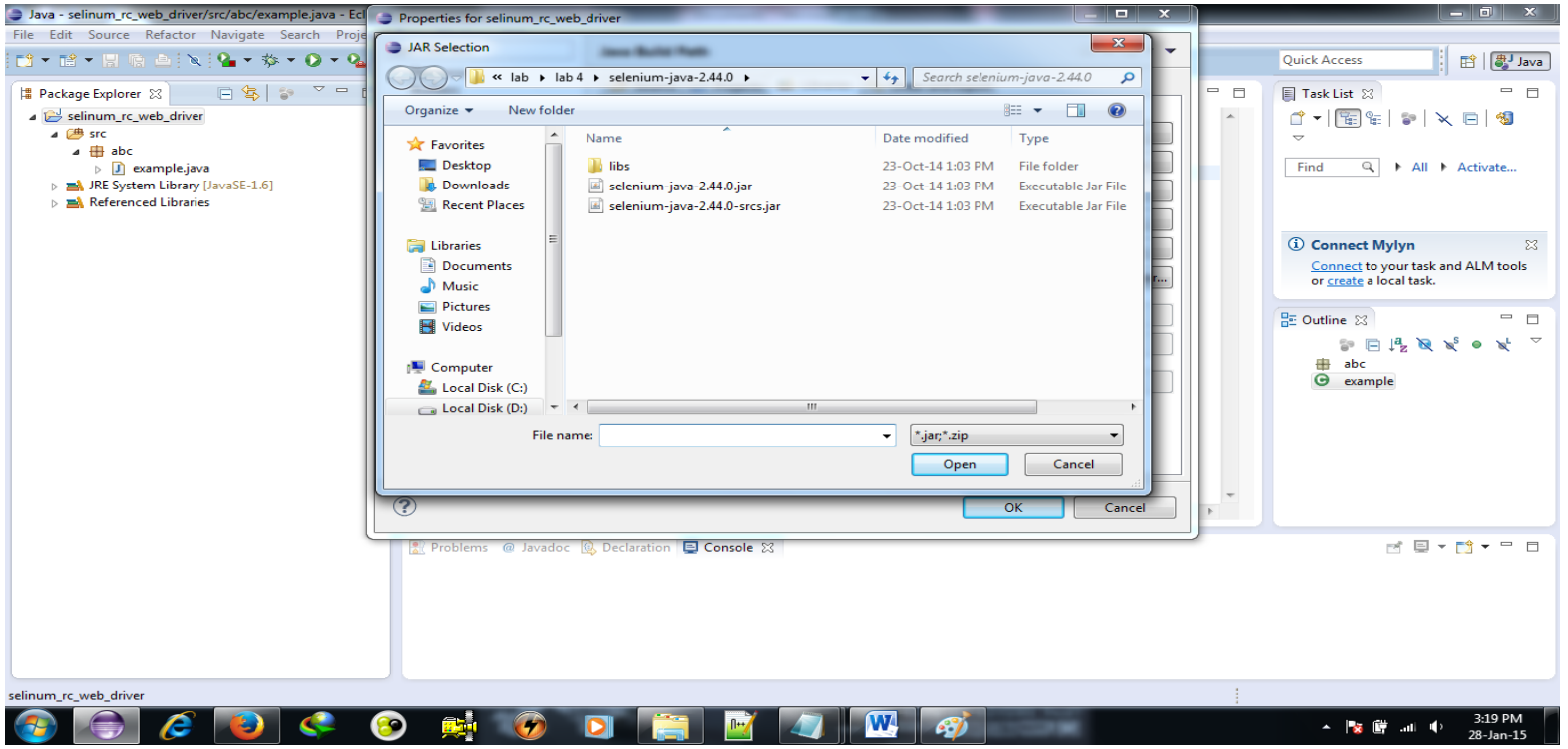
Implementation of Selenium Webdriver contd….

- Then click on libraries and add external jar files.



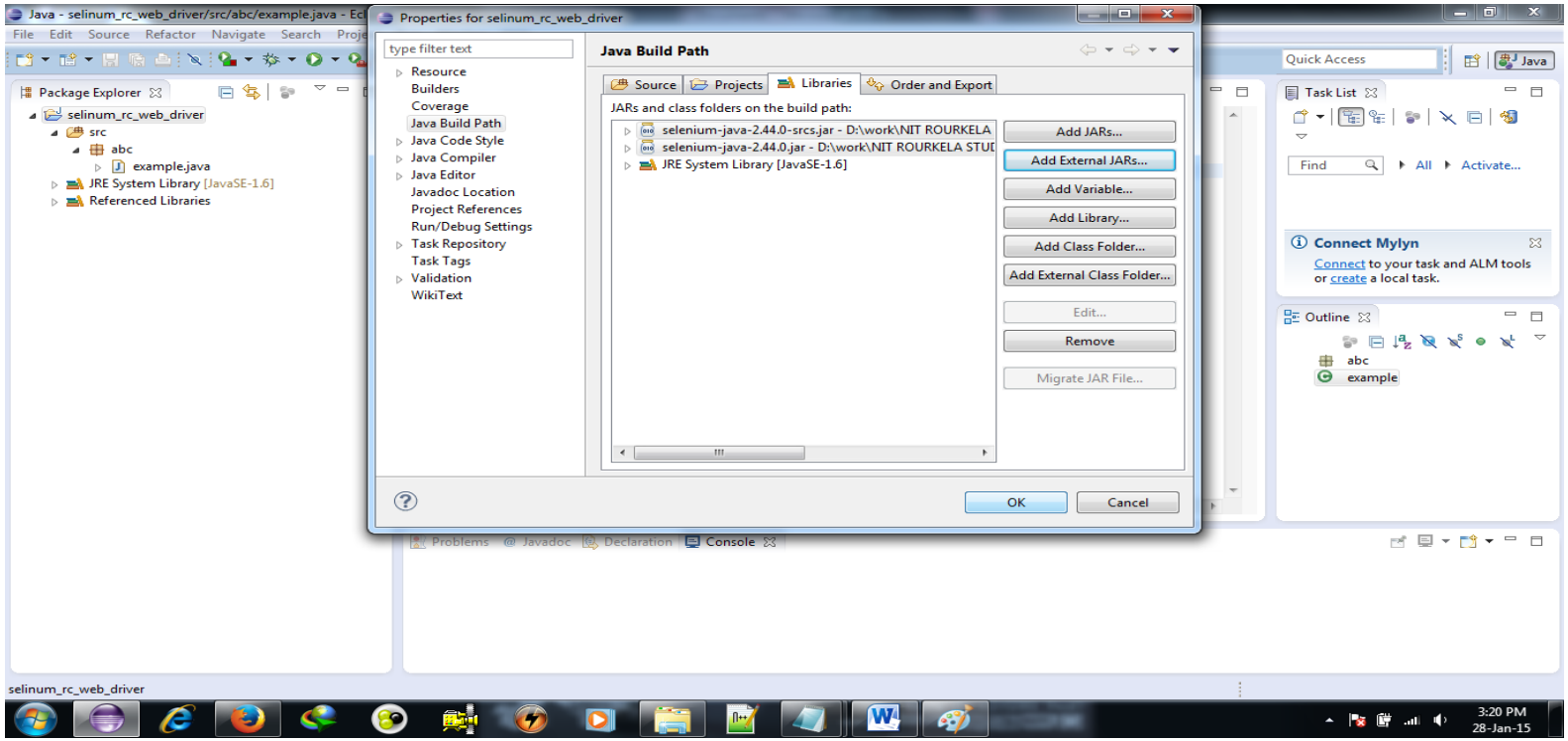
Implementation of Selenium Webdriver contd….

- Select all the associated jar files and click open.



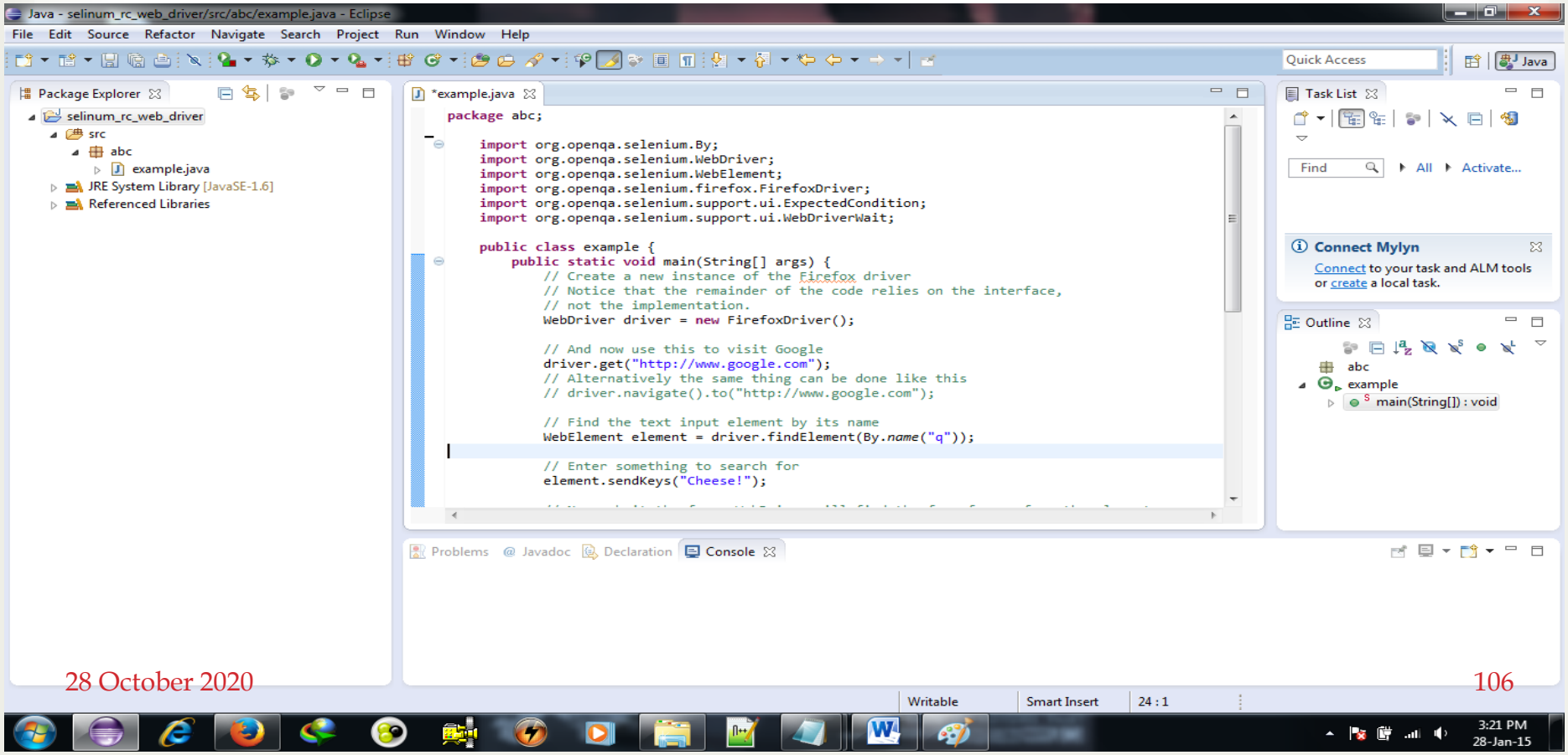
Implementation of Selenium Webdriver contd….

- Then click on OK



Implementation of Selenium Webdriver contd....

- Now write the code for execution and run it.



Implementation of Selenium Webdriver contd….

```
package org.openqa.selenium.example;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.WebDriverWait;
```

Implementation of Selenium Webdriver contd….

```
public class Selenium2Example{
public static void main(String[] args){
// Create a new instance of the Firefox driver
// Notice that the remainder of the code relies on the interface,
// not the implementation.
WebDriver driver=new FirefoxDriver();
// And now use this to visit Google
driver.get("http://www.google.com");
// Alternatively the same thing can be done like this
// driver.navigate().to("http://www.google.com");
```

Implementation of Selenium Webdriver contd….

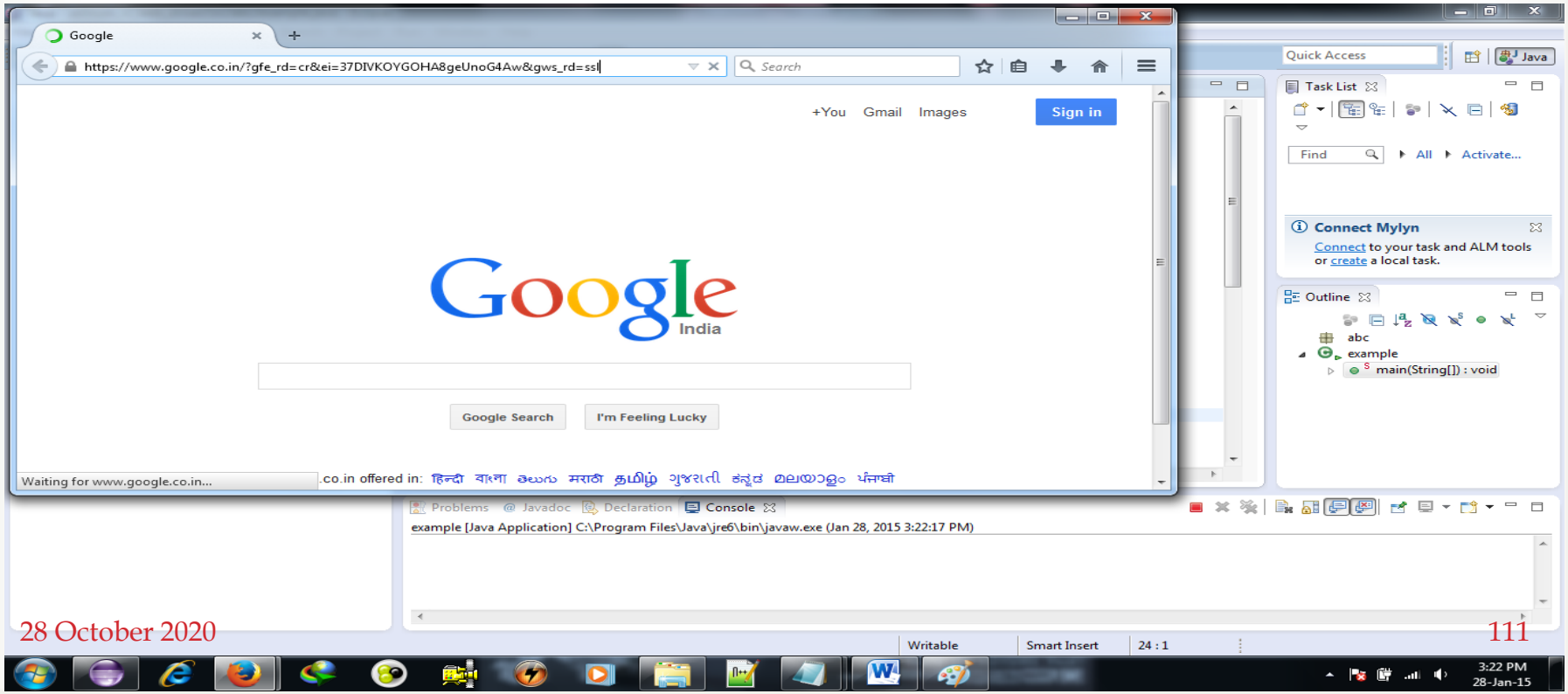
```
// Find the text input element by its name
WebElement element=driver.findElement(By.name("q"));
// Enter something to search for
element.sendKeys("Cheese!");
// Now submit the form. WebDriver will find the form for us from the element
element.submit();
// Check the title of the page
System.out.println("Page title is: "+driver.getTitle());
// Google's search is rendered dynamically with JavaScript.
```

Implementation of Selenium Webdriver contd….

```
// Wait for the page to load, timeout after 10 seconds
(new WebDriverWait(driver,10)).until(new
ExpectedCondition<Boolean>(){
public Boolean apply(WebDriver d){
return d.getTitle().toLowerCase().startsWith("cheese!");
}
});
// Should see: “cheese! - Google Search”
System.out.println(“Page title is: “+driver.getTitle());
//Close the browser
driver.quit();
}
}
```

Implementation of Selenium Webdriver contd….

- Browser will be opened automatically after the execution of code which is written in eclipse :



Thank You