DAY 1 – Selenium, Java, Eclipse Installation And Sample Selenium Script Creation Assignment

Selenium - Environment Setup

In order to develop Selenium RC or WebDriver scripts, users have to ensure that they have the initial configuration done. Setting up the environment involves the following steps.

* Download and Install Java
* Download and Configure Eclipse
* Configure FireBug and FirePath
* Configure Selenium RC
* Configure Selenium WebDriver

Download and Install Java

We need to have JDK (Java Development Kit) installed in order to work with Selenium WebDriver/Selenium. Let us see how to download and install Java.

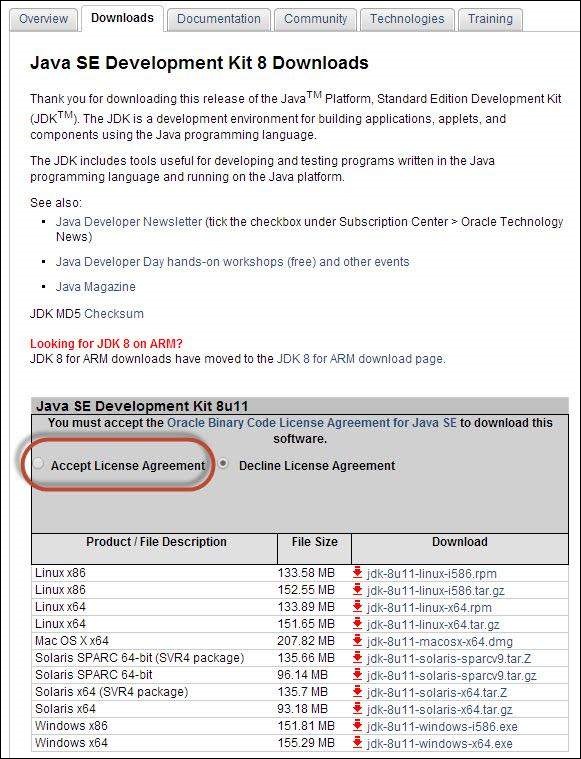
**Step 1** − Navigate to the UR.

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

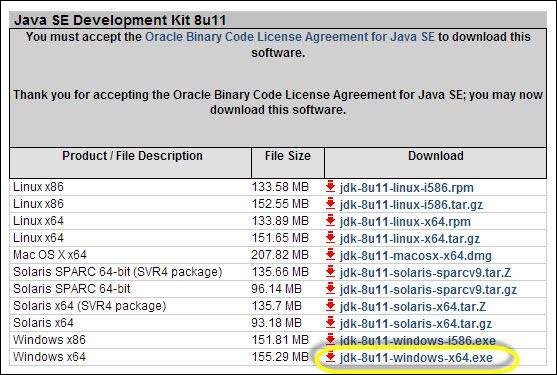
**Step 2** − Go to "Downloads" section and select "JDK Download".



**Step 3** − Select "Accept License Agreement" radio button.



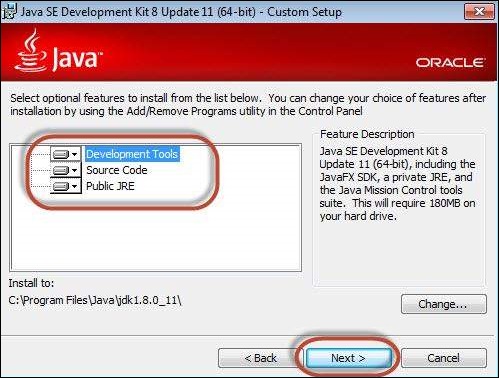
**Step 4** − Select the appropriate installation. In this case, it is 'Windows 7-64' bit. Click the appropriate link and save the .exe file to your disk.



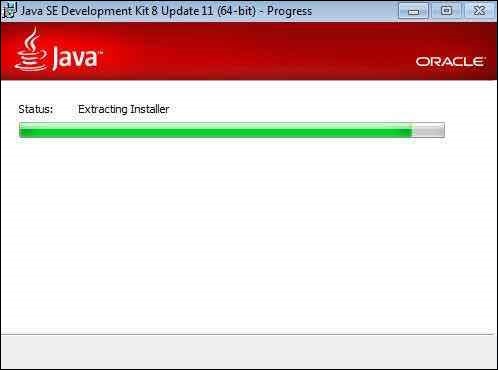
**Step 5** − Run the downloaded exe file to launch the Installer wizard. Click 'Next' to continue.



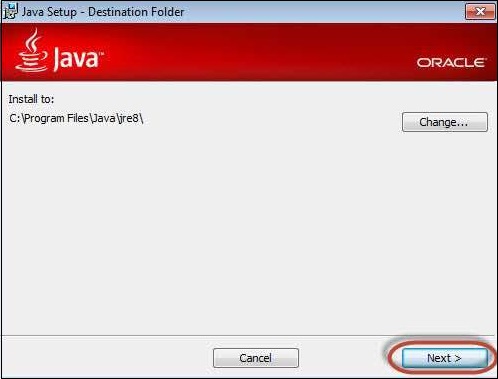
**Step 6** − Select the features and click 'Next'.



**Step 7** − The installer is extracted and its progress is shown in the wizard.



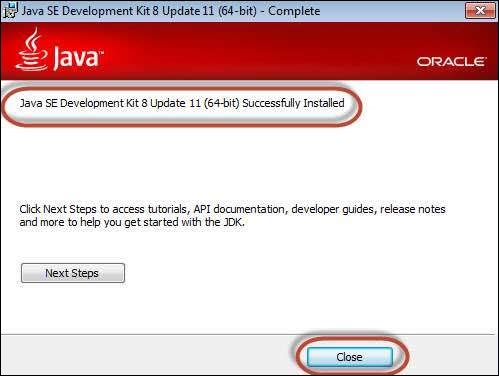
**Step 8** − The user can choose the install location and click 'Next'.



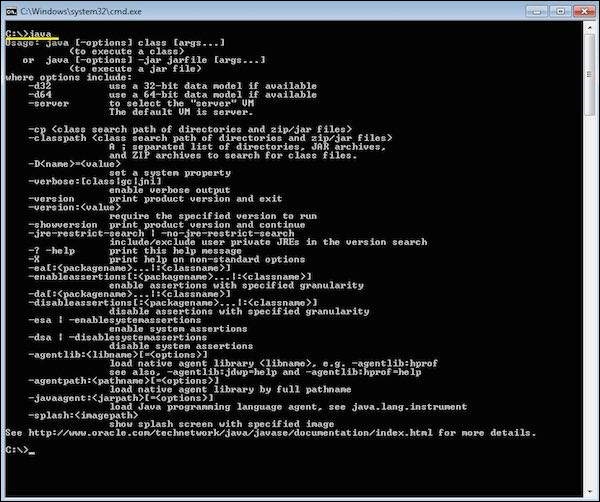
**Step 9** − The installer installs the JDK and new files are copied across.



**Step 10** − The Installer installs successfully and displays the same to the user.

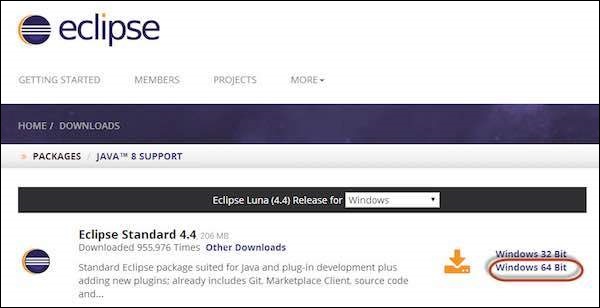


**Step 11** − To verify if the installation was successful, go to the command prompt and just type 'java' as a command. The output of the command is shown below. If the Java installation is unsuccessful or if it had NOT been installed, it would throw an "unknown command" error.



Download and Configure Eclipse

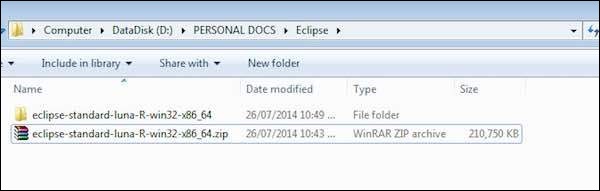
**Step 1** − Navigate to the URL: <https://www.eclipse.org/downloads/> and download the appropriate file based on your OS architecture.



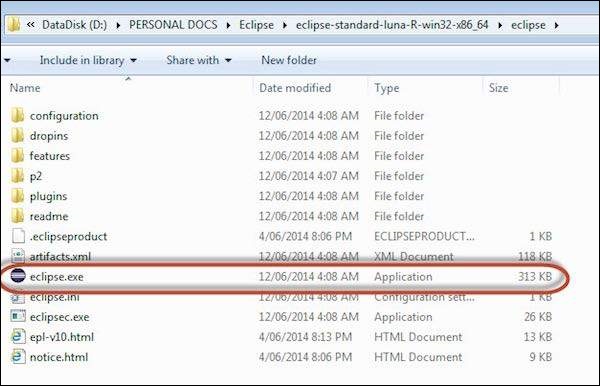
**Step 2** − Click the 'Download' button.



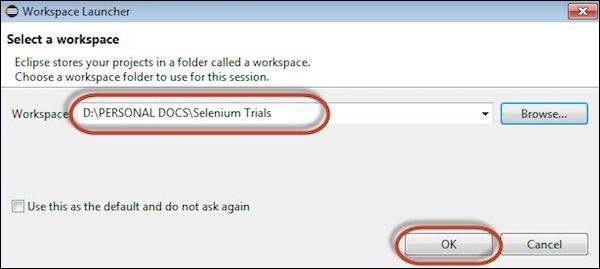
**Step 3** − The download would be in a Zipped format. Unzip the contents.



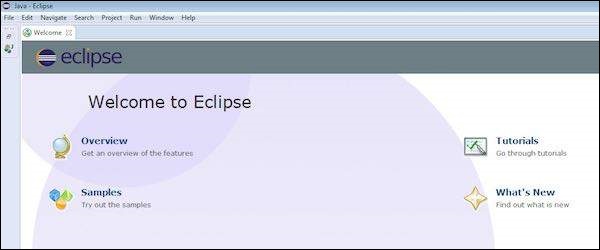
**Step 4** − Locate Eclipse.exe and double click on the file.



**Step 5** − To configure the workspace, select the location where the development has to take place.



**Step 6** − The Eclipse window opens as shown below.



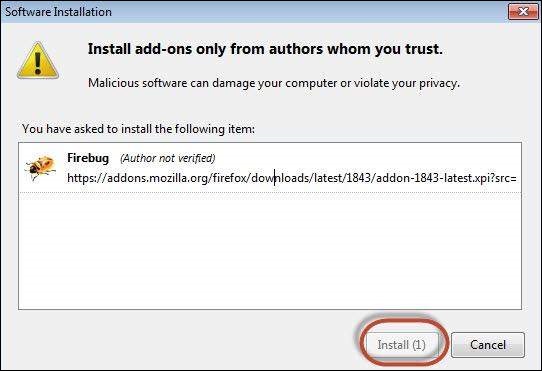
Configure FireBug and FirePath

To work with Selenium RC or WebDriver, we need to locate elements based on their XPath or ID or name, etc. In order to locate an element, we need tools/plugins.

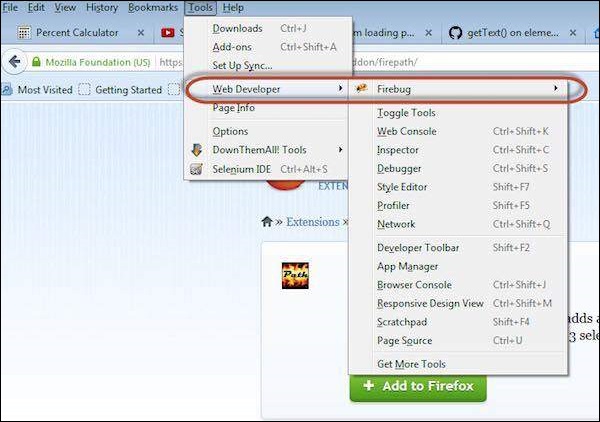
**Step 1** − Navigate to the URL : <https://addons.mozilla.org/en-US/firefox/addon/firebug/> and download plugin.



**Step 2** − The add-on installer is shown to the user and it is installed upon clicking the 'Install' button.



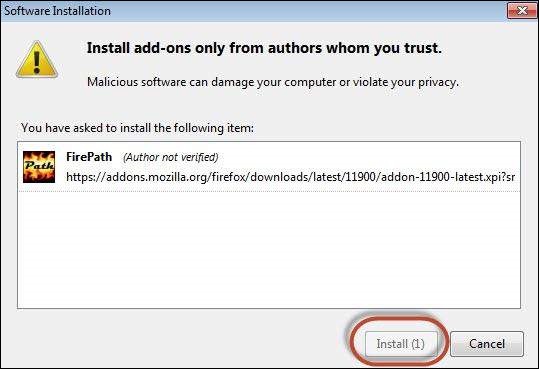
**Step 3** − After installing, we can launch the plugin by navigating to "Web Developer" >> "Firebug".



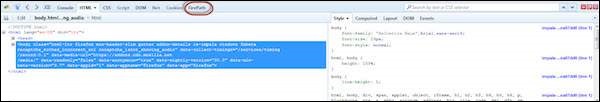
**Step 4** − FirePath, a plugin that works within Firebug, helps users to grab the 'XPath' of an element. Install FirePath by navigating to "<https://addons.mozilla.org/en-US/firefox/addon/firepath/>"



**Step 5** − The add-on installer is shown to the user and it is installed upon clicking the 'Install' button.



**Step 6** − Now launch "Firebug" by navigating to "Tools" >> "Webdeveloper" >> "Firebug".



Example

Now let us understand how to use FireBug and FirePath with an example. For demonstration, we will use www.google.com and capture the properties of the text box of "google.com".

**Step 1** − First click on the arrow icon as highlighted in the following screenshot and drag it to the object for which we would like to capture the properties. The HTML/DOM of the object would be displayed as shown below. We are able to capture the 'ID' of the input text box with which we can interact.



**Step 2** − To fetch the XPath of the object, go to 'firepath' tab and perform the following steps.

* Click the Spy icon.
* Select the Control for which we would like to capture the XPath.
* XPath of the selected control would be generated.



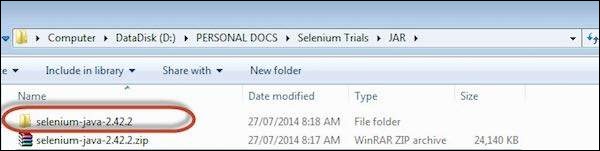
Configure Selenium WebDriver

Now let us look at how to configure Selenium WebDriver. We will understand how to develop scripts with Selenium WebDriver in later chapters, however for now, we will understand just the configuration part of it.

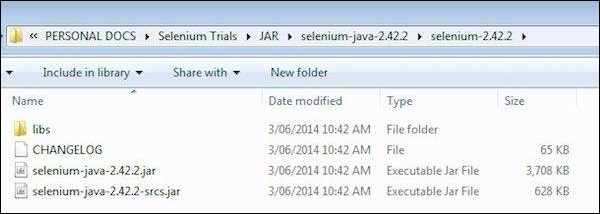
**Step 1** − Navigate to the selenium downloads section <http://www.seleniumhq.org/download/> and download Selenium WebDriver by clicking on its version number as shown below.



**Step 2** − The downloaded file is in Zipped format and one has to unzip the contents to map it to the project folder.



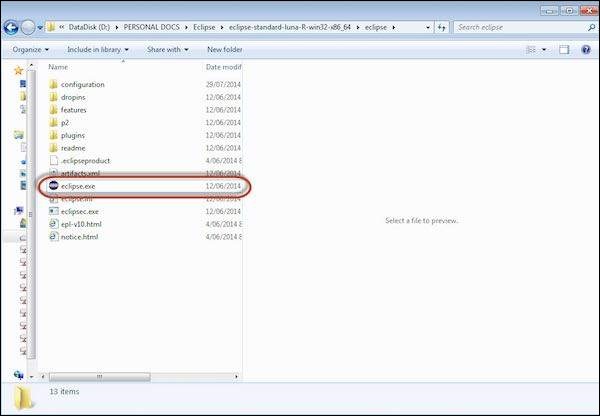
**Step 3** − The Unzipped contents would be displayed as shown below. How to map it to the project folder and how to start scripting would be dealt in the webDriver chapter.



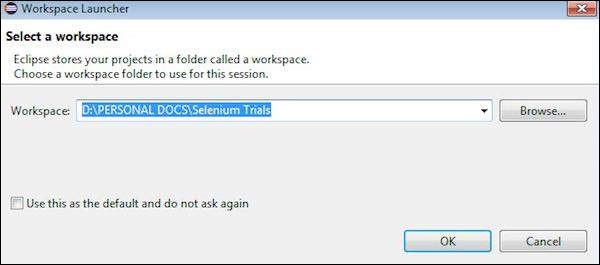
Scripting using WebDriver

Let us understand how to work with WebDriver. For demonstration, we would use <https://www.calculator.net/>. We will perform a "Percent Calculator" which is located under "Math Calculator". We have already downloaded the required WebDriver JAR's. Refer the chapter "Environmental Setup" for details.

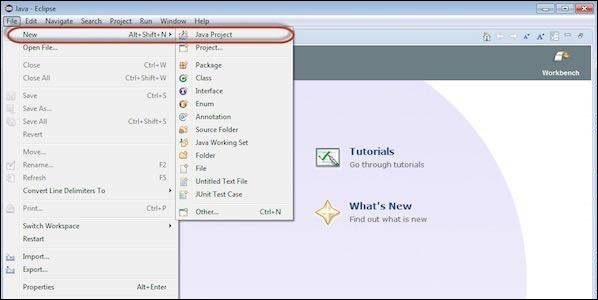
**Step 1** − Launch "Eclipse" from the Extracted Eclipse folder.



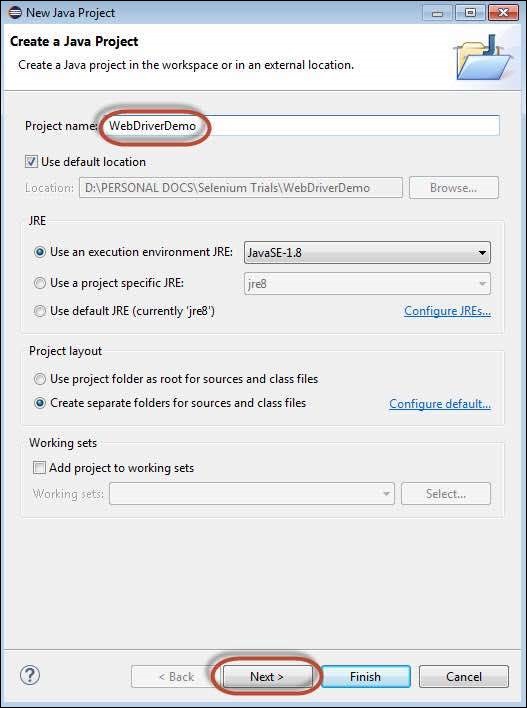
**Step 2** − Select the Workspace by clicking the 'Browse' button.



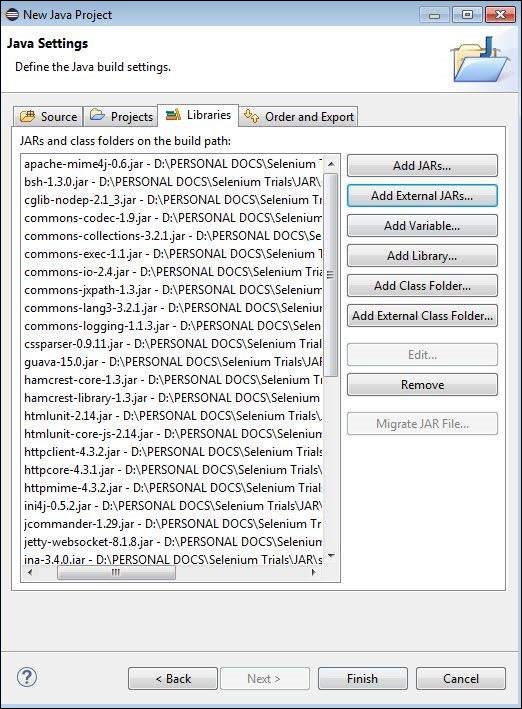
**Step 3** − Now create a 'New Project' from 'File' menu.



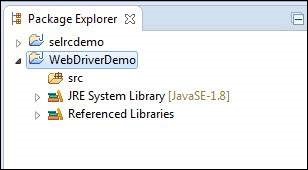
**Step 4** − Enter the Project Name and Click 'Next'.



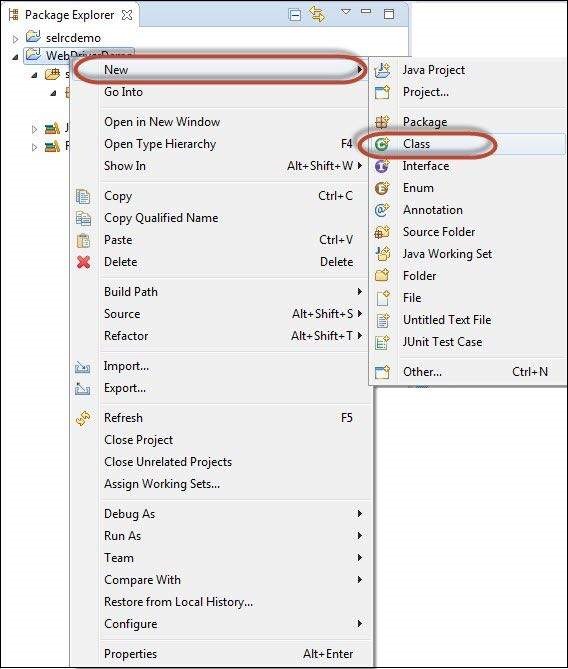
**Step 5** − Go to Libraries Tab and select all the JAR's that we have downloaded. Add reference to all the JAR's of Selenium WebDriver Library folder and also selenium-java-2.42.2.jar and selenium-java-2.42.2-srcs.jar.



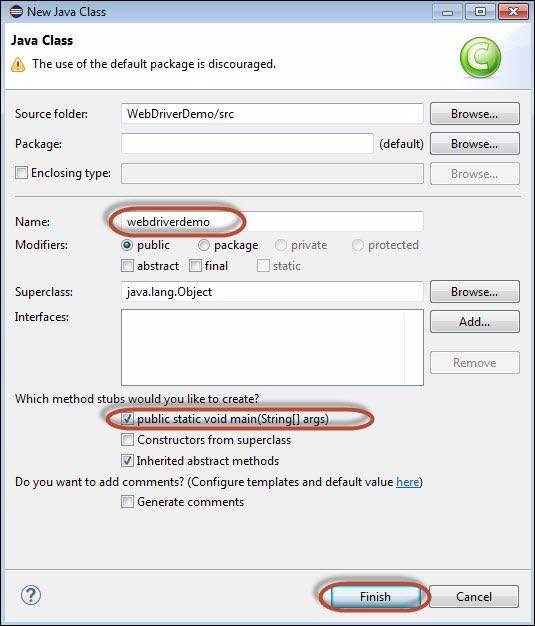
**Step 6** − The Package is created as shown below.



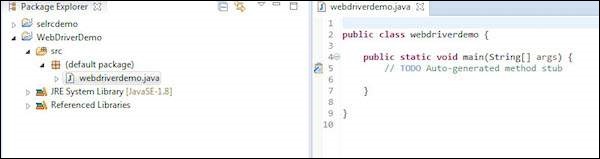
**Step 7** − Now right-click on the package and select 'New' >> 'Class' to create a 'class'.



**Step 8** − Now name the class and make it the main function.



**Step 9** − The class outline is shown as below.



**Step 10** − Now it is time to code. The following script is easier to understand, as it has comments embedded in it to explain the steps clearly. Please take a look at the chapter "Locators" to understand how to capture object properties.

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.\*;

import org.openqa.selenium.firefox.FirefoxDriver;

public class webdriverdemo {

public static void main(String[] args) {

WebDriver driver = new FirefoxDriver();

//Puts an Implicit wait, Will wait for 10 seconds before throwing exception

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

//Launch website

driver.navigate().to("http://www.calculator.net/");

//Maximize the browser

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

// Click on Math Calculators

driver.findElement(By.xpath(".//\*[@id = 'menu']/div[3]/a")).click();

// Click on Percent Calculators

driver.findElement(By.xpath(".//\*[@id = 'menu']/div[4]/div[3]/a")).click();

// Enter value 10 in the first number of the percent Calculator

driver.findElement(By.id("cpar1")).sendKeys("10");

// Enter value 50 in the second number of the percent Calculator

driver.findElement(By.id("cpar2")).sendKeys("50");

// Click Calculate Button

driver.findElement(By.xpath(".//\*[@id = 'content']/table/tbody/tr[2]/td/input[2]")).click();

// Get the Result Text based on its xpath

String result =

driver.findElement(By.xpath(".//\*[@id = 'content']/p[2]/font/b")).getText();

// Print a Log In message to the screen

System.out.println(" The Result is " + result);

//Close the Browser.

driver.close();

}

}

**Step 11** − The output of the above script would be printed in Console.

