TEST AUTOMATION & ADVANCED SELENIUM

TEST AUTOMATION & ADVANCED SELENIUM LAB BOOK

LAB BOOK VERSION 2.0

Table of Contents

Document Revision History 2
Table of Contents 3
Getting Started 4
Lab Demo 1: Example (Basic Selenium IDE Flow) 5
Lab Demo: Example (Selenium IDE) (Cont.) 6
Lab Demo: Example (Selenium IDE) (Cont.) 7
Lab Demo 2: Learning Selenium IDE (Modifications) 8
Lab Demo 3: Learning Selenium IDE(Basic Flow) 9
Lab Demo 4: Learning Selenium IDE(Performing Validations) 10
Lab Demo 5: Create a new account (using Selenium Webdriver) 11
Lab Demo 6: Validations in Selenium (using Selenium Webdriver)..... 13
Lab Demo 7: Alert and window handling (using Selenium Webdriver) 14
Lab Demo 8: WebDriver with JUnit/TestNG (using Selenium Webdriver with Junit and TestNG) 15
Lab Demo 9: Advance Selenium (Chrome Driver and IE Driver) 16
Lab Demo 10: Advance Selenium (RemoteWebDriver)..... 17
Lab Demo 11: Advance Selenium (Page object Model and Page Factory) 18
Lab Demo 12: Advance Selenium (Object Repository using Properties File)..... 19
Lab Demo 13: Advance Selenium (Object Repository using XML file) 20
Lab Demo 14: Advance Selenium (Parametrization using Excel Apache POI) 21
Lab Demo 15: Advance Selenium (Parameterization using CSV file) 22
Lab Demo 16: Advance Selenium (Parameterization using XML and Dataprovider in TestNG) 23
Appendices 24
Appendix A: Selenium Standards 24

Getting Started

Overview

This lab book is a guided tour for learning Test Automation & Advanced Selenium. It comprises solved examples and 'To Do' assignments. Follow the steps provided in the solved examples and work out the 'To Do' assignments given.

Setup Checklist for Selenium

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

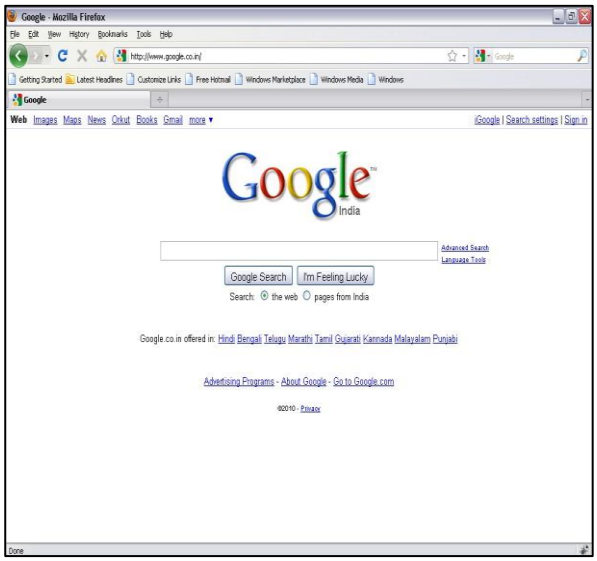
- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP.
- Memory: 512MB of RAM
- Internet Explorer 6.0 or higher
- Mozilla Firefox(Add-ons: Firebug, Firepath, Selenium IDE)

Instructions

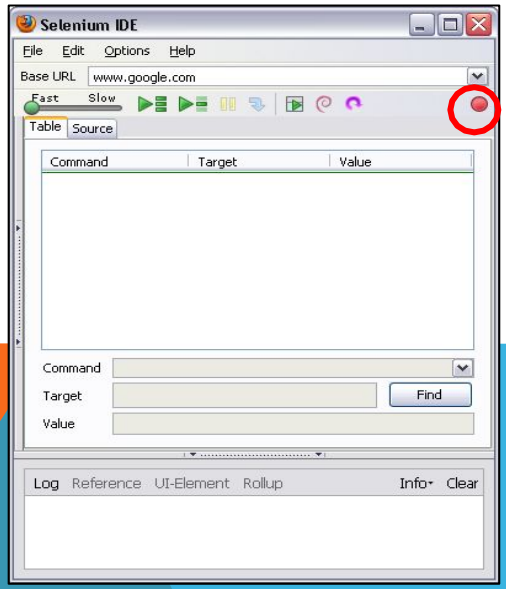
- For all coding standards refer Appendix A. All lab assignments should refer coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory Selenium_Assign. For each lab exercise create a directory as lab <lab number>.

Lab Demo 1: Example (Basic Selenium IDE Flow)

Goals	• Understand the process of automation testing of a web application on Selenium IDE • Learn to manage document spacing
Time	60 minutes

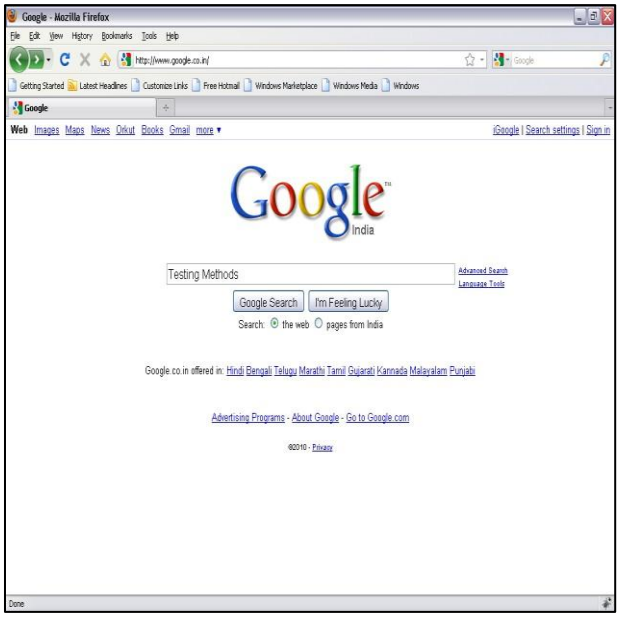


Go to the Web Page for which you want to carry out the test

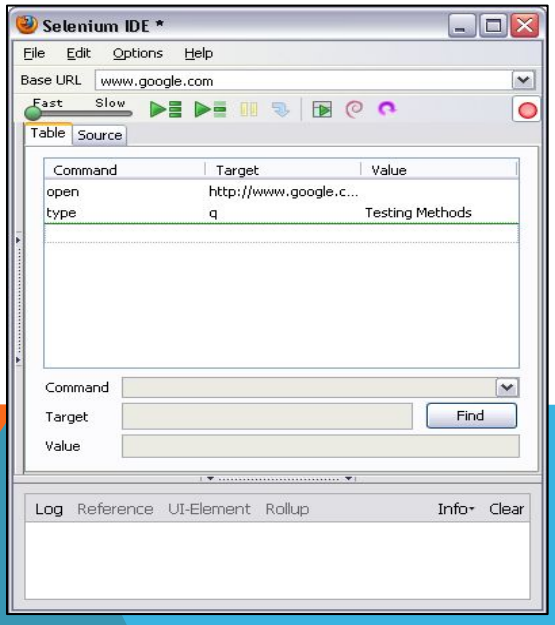


Hit the record button on IDE

Lab Demo: Example (Selenium IDE) (Cont.)

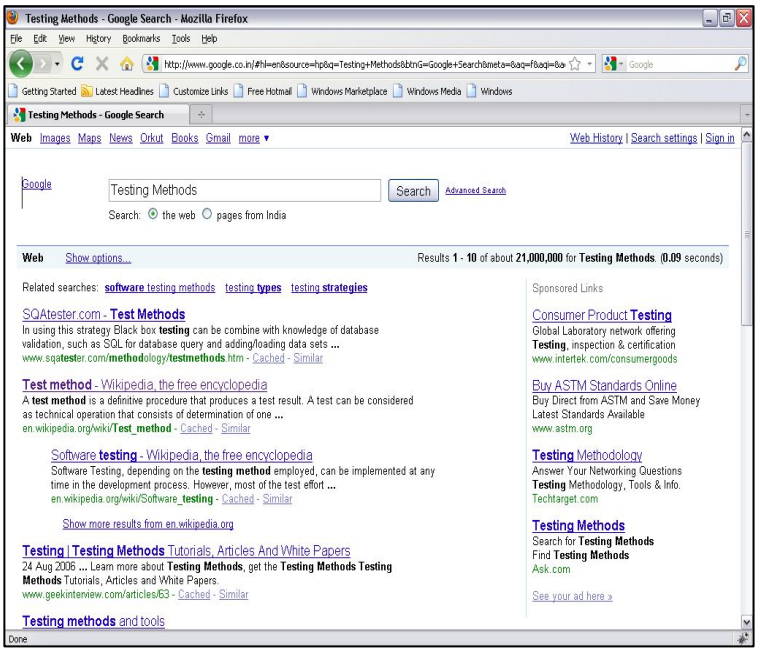


Enter the text on Web Page and submit

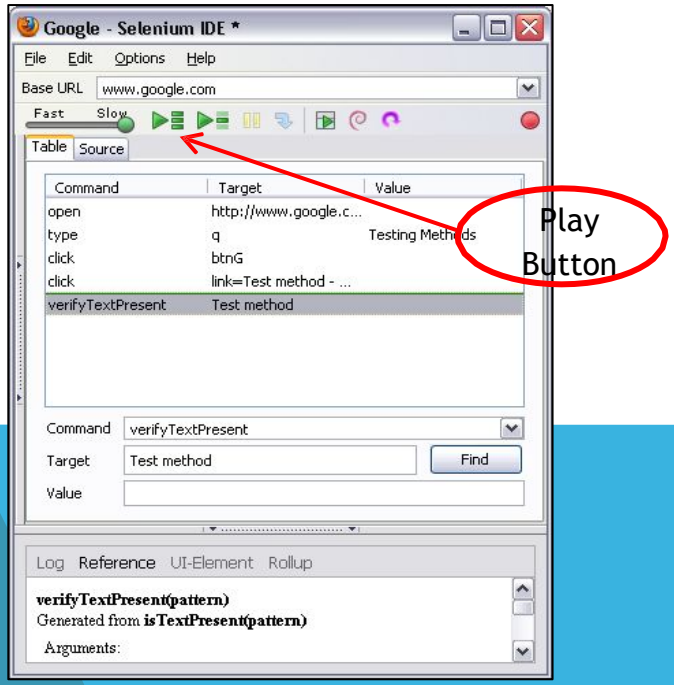


IDE should be updated, stop the recorder and add the assertions

Lab Demo: Example (Selenium IDE) (Cont.)



Perform operations on the web page



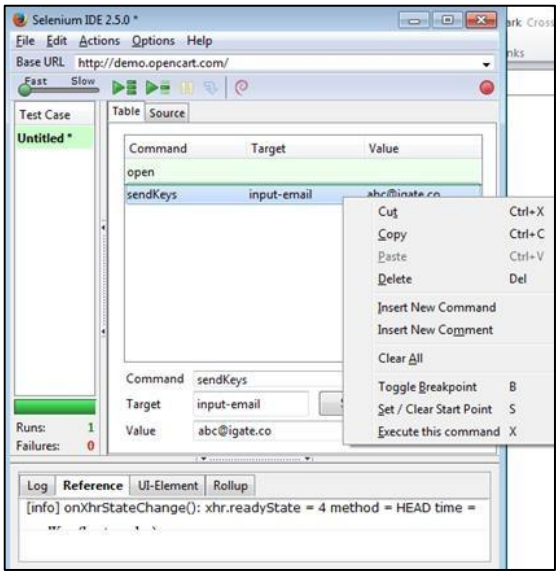
Hit the play button to play the recorded scripts

Lab Demo 2: Learning Selenium IDE (Modifications)

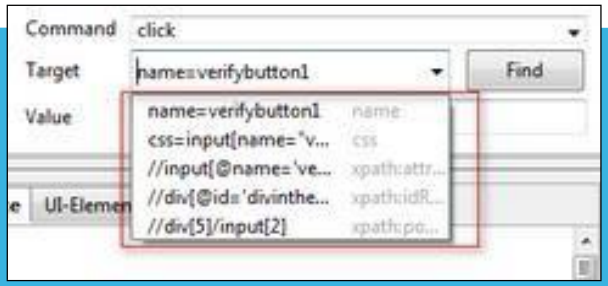
Goals	• Understand the process of further modifications and validations in automation testing of a web application on Selenium IDE • Learn to manage document spacing
Time	60 minutes

Basic URL: http://demo.opencart.com/

1. Follow the steps from Lab Demo 1
2. In order to add new command, go to the command where the new command needs to be added and then do right click , click on 'Insert New Command'



3. 'Command' dropdown provides options/ keywords to perform the operation

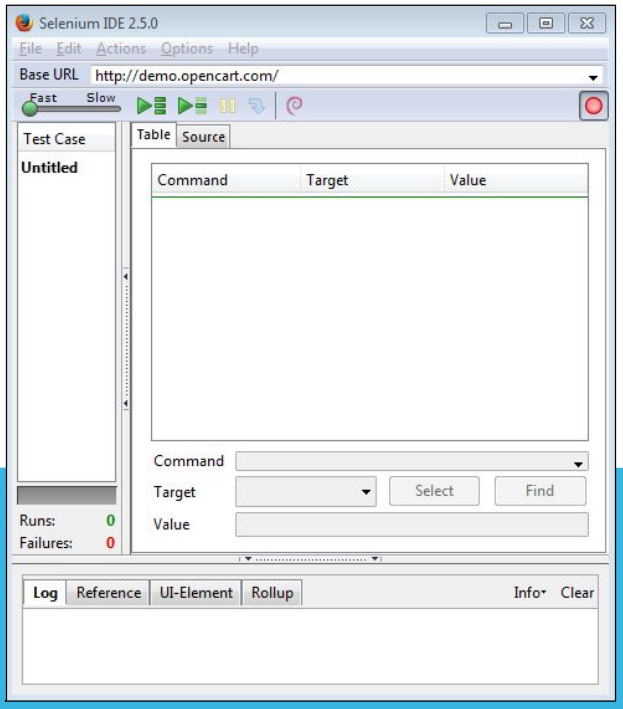


Lab Demo 3: Learning Selenium IDE(Basic Flow)

Goals	• Understand the basic process of automation testing of a web application in Selenium IDE • Learn to manage document spacing
Time	60 minutes

Basic URL: http://demo.opencart.com/

1. Open the URL on Firefox
2. Start recording on Selenium IDE
3. Go to 'Desktops' tab
4. Click on 'Mac'
5. Select 'Name(A-Z)' from the 'Sort By' dropdown
6. Click on 'Add to Cart' button
7. Stop the recording on Selenium IDE
8. Playback the whole test case



Lab Demo 4: Learning Selenium IDE(Performing Validations)

Goals	• Understand the process of automation testing of a web application • Learn to manage document spacing
Time	60 minutes

Basic URL: http://demo.opencart.com/

1. Open the URL on Firefox
2. Start recording on Selenium IDE
3. Verify title of the page
4. Go to 'Desktops' tab
5. Click on 'Mac'
6. Select 'Name(A-Z)' from the 'Sort By' dropdown
7. Click on 'Add to Cart' button
8. Enter ‘Mobile’ in ‘Search’ text box and click on ‘Search’ button
9. Wait for page to load
10. Clear the text from ‘Search Criteria’ text box
11. Click on ‘Search in product descriptions’ check box and click on ‘Search’ button
12. Stop the recording on Selenium IDE
13. Add the step after Step 5 where verify the 'Mac' heading
14. Change the value from ‘Mobile’ to ‘Monitors’
15. Save the test case
16. Playback the whole test case.
17. Playback set by step
18. Add the step after Click on 'Mac' where verify the 'Mac' heading
19. Save the test case
20. Export the test case as 'Java/JUnit/webDriver'
21. Run the whole Test case and check the 'Pass' Status
22. Create another test case with the same flow
23. Create the test suite for the above test cases

Lab Demo 5: Create a new account (using Selenium Webdriver)

Goals	• Understanding the scenario from end to end and automating the same • Analyze the requirement and perform validations accordingly
Time	120 minutes

Basic URL: http://demo.opencart.com/

Please ensure that the variables have to be defined before it is being used.

Part 1: Launch Application

1. Launch the URL on Firefox
2. Verify 'Title' of the page
3. Click on 'My Account' dropdown
4. Select 'Register' from dropdown
5. 'Register Account' page will open up, verify the heading 'Register Account'
6. Click on 'Continue' button at the bottom of the page
7. Verify warning message 'Warning: You must agree to the Privacy Policy!'

Automate and validate the different sections of 'Register Account' page:

Part 2: For 'Your Personal Details'

1. Enter data in 'First Name' text box
2. Verify if 33 characters can be entered in 'First Name' text box by clicking on 'Continue' button.
3. If not, verify error message.
4. Enter data in 'Last Name' text box
5. Verify if 33 characters can be entered in 'First Name' text box by clicking on 'Continue' button.
6. If not, verify error message.
7. Enter valid 'E-mail'.
8. Enter 'Telephone' which must be between 3 and 32 characters.

Part 3: For 'Your Address'

1. ~~Enter 'Address 1' which should contain characters between 3 and 128~~
2. Enter 'City' which should contain characters between 2 and 128
3. Enter 'Post Code' which should contain characters between 2 and 10

4. Select 'India' from 'Country' Dropdown
5. Select 'Region/State' from dropdown

Part 4: For 'Password'

1. Enter 'Password' which must be between 4 and 20 characters.
2. Enter 'Password Confirm'.

Part 4: For 'Newsletter'

1. Click on 'Yes' Radio button
2. Click on checkbox for 'I have read and agree to the Privacy Policy'.
3. Click on 'Continue' button
4. Verify message 'Your Account Has Been Created!'
5. Click on 'Continue'
6. Click on link 'View your order history' under 'My Orders'

Lab Demo 6: Validations in Selenium (using Selenium Webdriver)

Goals	• Understanding the scenario from end to end and automating the same
Time	120 minutes

Basic URL: http://demo.opencart.com/

After login on the 'Open Cart', create a test script following the below mentioned steps:

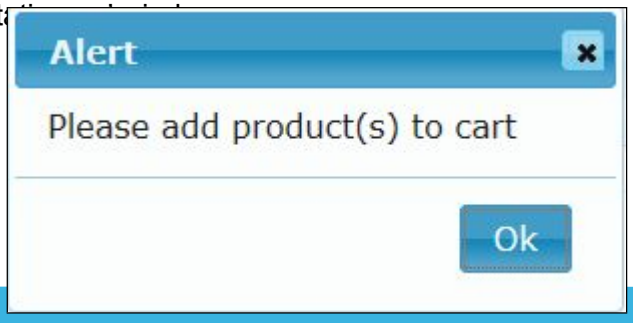
1. Login with credentials created in Lab 1
2. Go to 'Components' tab and click
3. Select 'Monitors'
4. Select 25 from 'Show' dropdown
5. Click on 'Add to cart' for the first item
6. Click on 'Specification' tab
7. Verify details present on the page
8. Click on 'Add to Wish list' button.
9. Verify message 'Success: You have added Apple Cinema 30" to your wish list!'
10. Enter 'Mobile' in ' Search' text box.
11. Click on 'Search' button
12. Click on 'Search in product descriptions' check box
13. Click on link 'HTC Touch HD' for the mobile 'HTC Touch HD'
14. Clear '1' from 'Qty' and enter '3'
15. Click on 'Add to Cart' button
16. Verify success message 'Success: You have added HTC Touch HD to your shopping cart!'
17. Click on 'View cart' button adjacent to search button
18. Verify Mobile name added to the cart
19. Click on 'Checkout' button
20. Click on 'My Account' dropdown
21. Select 'Logout' from dropdown
22. Verify 'Account Logout' heading
23. Click on 'Continue'

Lab Demo 7: Alert and window handling (using Selenium Webdriver)

Goals	Learning alert handling and window handling basics in selenium webdriver
Time	120 minutes

Base URL: https://ispace.ig.capgemini.com/sitepages/index.aspx

1. Launch the URL
2. Go to 'Application' tab
3. Click on checkbox 'Stationery Request'
4. Verify the new title ‘Stationary’
5. On ‘Stationery’ tab, click on 'Submit to collect your Stationery >>>' link
6. Switch to alert as shown below.
7. Verify text on the alert 'Please add product(s) to cart'
8. Verify if 'Ok' button is present on the alert
9. Click on 'Ok' button
10. Click on ‘Photocopy’ tab, click on ‘Save Request’ button
11. Switch to alert and verify text ‘No changes made’
12. Verify if ‘Ok’ button is present
13. Click on ‘Ok’ button present on alert
14. Click on ‘Logout’ button
15. Close the ‘St



Lab Demo 8: WebDriver with JUnit/TestNG (using Selenium Webdriver with Junit and TestNG)

Goals	Learning how to write webdriver automation testcases using Junit and TestNG
Time	120 minutes

1. Consider the flow mentioned in Lab 3 & Lab 4, complete the task using Selenium WebDriver and JUnit.
   [**NOTE:** All the verifications should be using JUnit Assertions]

2. Consider the flow mentioned in Lab 3 & Lab 4, complete the task using Selenium WebDriver and TestNG.
   [**NOTE:** All the verifications should be using TestNG Assertions. Testcases should have proper TestNG Reporter logging as well.]

3. Create one TestNG test suite for both the testcases created for question number 2 along with **testing.xml** and execute the test suite. Provide the **Reports** as well.

Lab Demo 9: Advance Selenium (Chrome Driver and IE Driver)

Goals	Learning how to write webdriver automation testcases for Chrome and IE browser
Time	120 minutes

1. Consider the flow mentioned in Lab 3 & Lab 4, make necessary changes to execute the same flow on both Chrome and Internet Explorer browser.

2. Consider the Question number 1 and make it JUnit test case.
   [**NOTE:** All the verifications should be using JUnit Assertions]

Lab Demo 10: Advance Selenium (RemoteWebDriver)

Goals	Learning how execute Selenium scripts using RemoteWebDriver
Time	120 minutes

1. Consider the flow mentioned in Lab 3 & Lab 4, make necessary changes to execute the same flow using RemoteWebDriver and Selenium Grid on Firefox, Chrome and Internet Explorer browser.
   [NOTE: Set Platform, BrowserName and Version to DesiredCapabilities]

2. Consider the flow mentioned in Lab 4 and take screenshot after each steps. Save all the screenshots inside a folder called '**Screenshots**' in the root of the Java project.

Lab Demo 11: Advance Selenium (Page object Model and Page Factory)

Goals	Learning how to use page object model and page factory to make the test maintenance easy
Time	121 minutes

1. Consider the flow mentioned in Lab 3 & Lab 4, complete the task using page object model.
   Hints : create one class for locators and other classes for Test cases.

2. Consider the flow mentioned in Lab 3 & Lab 4, complete the task using page object model with page factory

Lab Demo 12: Advance Selenium (Object Repository using Properties File)

Goals	Learning how to create Object Repository using Properties File
Time	122 minutes

1. Consider the flow mentioned in Lab 4, complete the task using properties file object repository.

Steps to create Properties File in eclipse

1. Create java project
2. Add Selenium webdriver jar file
3. Right click on project and create folder – “configuration”
4. Right click on configuration folder -> New -> File and give a file name – “config.property”
5. Enter keys and values in property file

*config.property TestLoginPage.java

```
chromedriver=C:\\Selenium Lib\\chromedriver.exe
IEdriver=C:\\Selenium Lib\\IEDriverServer.exe
url=https://www.google.com/
username=un
password=//input[@id='pass']
login_button=login
```

Lab Demo 13: Advance Selenium (Object Repository using XML file)

Goals	Learning how to create Object repository using XML file
Time	123 minutes

Consider the flow mentioned in Lab 4, complete the task using properties file object repository.

Steps to create XML file in eclipse

1. Create Java project
2. Right click on project -> New -> Other -> XML - XML File
3. Name xml file - ObjectRepository.XML
4. Write XML code
5. Import jaxen.jar and dom4j-1.6.jar file in eclipse to read XML file

Find below example for XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<login_detail>
    <username>un</username>
    <password>//input[@id='pass']</password>
    <login_button>login</login_button>
</login_detail>
```

Lab Demo 14: Advance Selenium (Parametrization using Excel Apache POI)

Goals	Learning how to use parameterization using Excel and Apache POI lib., to achieve regression testing.
Time	60 minutes

Basic URL: http://demo.opencart.com/

1. **Create "**UserDetails.xls" file with First_Name, Last_Name, E-Mail, Telephone, Password and Password Confirm columns and insert data in the sheet.

Follow below flow to write Selenium Webdriver script

1. Open the URL on Firefox/chrome/IE browser
2. Verify if the title "Your store" of the application is correct
3. Click on "My Account" menu option
4. Select "Register" option
5. Verify the text present on web page as "Register account
6. Enter all the details in the First name, Last Name, E-Mail, Telephone, Password, Password Confirm from the excel sheet (UserDetails.xls)
7. Select "I have read and agree to the Privacy Policy" check box
8. Click on "Continue" button
9. Verify the acknowledgement message "Your Account Has Been Created"

Hint : Use Appache POI lib file to connect with excel sheet through Selenium Webdriver.

Lab Demo 15: Advance Selenium (Parameterization using CSV file)

Goals	Learning how to use parametrization using CSV file to achieve data driven testing
Time	60 minutes

Basic URL: http://demo.opencart.com/

1. **Create "**UserDetails.csv" file with First_Name, Last_Name, E-Mail, Telephone, Password, Pass_Confirm insert 2 rows of data in the CSV file.

Follow below flow to write Selenium Webdriver script

1. Open the URL on Firefox/chrome/IE
2. Verify if the title "Your store" of the application is correct
3. Click on "My Account" menu option
4. Select "Register" option
5. Verify the text present on web page as "Register account
6. Enter all the details in the First name, Last Name, E-Mail and Telephone from the CSV file (UserDetails.csv)
7. Select "I have read and agree to the Privacy Policy" check box
8. Click on "Continue" button
9. Verify the acknowledgement message "Your Account Has Been Created"

Note : Use notepad to create CSV file and save it with .CSv extention

Lab Demo 16: Advance Selenium (Parameterization using XML and Dataprovider in TestNG)

Goals	Learning how to use XML and Dataprovider annotation in TestNG to achieve parameterization
Time	60 minutes

Basic URL: http://demo.opencart.com/

Steps to create TestNg.XML file

1. **Create class in java project**
2. **Use @ Test annotation and create test case**
3. **Right click on class -> TestNG ->Convert to TestNG**
4. **Give the proper class name and pass the parameter in XML file**

1. Follow below flow to write Parameterized Selenium Webdriver script using XML in TestNG

1. Open the URL on Firefox/chrome/IE browser
2. Verify if the title “Your store” of the application is correct
3. Click on “My Account” menu option
4. Select “Register” option
5. Verify the text present on web page as “Register account”
6. Enter all the details in the First name, Last Name, E-Mail, Telephone, password and password confirm fields from the excel sheet (UserDetails.xls) using TestNG
7. Select “I have read and agree to the Privacy Policy” check box
8. Click on “Continue” button
9. Verify the acknowledgement message “Your Account Has Been Created”

14.2 Consider above flow and insert data into the application using DataProvider annotation in TestNg

Note : Use @DataProvider(name= “UserDetails”) annotation

Appendices

Appendix A: Selenium Standards

Key points to keep in mind:

Selenium standards help you reach the widest possible audience.
There are many technologies that are associated with HTML because they are used on web pages or in conjunction with HTML.

For each of the above, please follow coding conventions specified by that technology.

Sometimes you are going to have to break rules and use non-standard syntax for good reason. Try to keep this to the minimum.

How to follow Selenium standards:

See the W3C site for more information on locator types and keyword statements.

The important thing to remember is that proper keywords are essential to assist validation software in checking your document.
Use Selenium IDE to find the locators for the elements.

Refer to W3C for technical and syntax information.

Always use the comments with the codes for proper understanding.