**Selenium Assignments**

**Note:**

* **Create a project “Selenium Assignments” under “Selenium Assignments” workspace in your machines**
* **Configure required Jar files for your project using MAVEN build tool**

**Pre-Requisites: Chrome/FF/Edge Browser & Chrome/Gecko/Edge Drivers, Latest Eclipse, “Selenium-Java” web driver jars, JAVA (Version: above 1.8)**

**-----------------------------------------------------------------------------------------------------------------------------------------**

**Note: Use all selenium waits wherever it’s requiring**

**Assignment -1: (Browser Commands)**

**Part-1:**

1. Launch new Browser
2. Access the url: <https://netbanking.hdfcbank.com/netbanking/>
3. Get the page title & print in Eclipse Console
4. Get Page URL & Print in Eclipse Console
5. Get Page Source (HTML Source code) Print in Eclipse Console
6. Close the Browser using close () command

(You will notice that only one window will close)

**Part-2:**

1. Launch new Browser
2. Access the url: <https://selectorshub.com/xpath-practice-page/>
3. Maximize the window
4. Click on “Donate” button using -----below line of code

driver. findElement(By.xpath("//div[@class='ast-button']]")).click();

1. Get the current page title and print in Eclipse Console
2. Get Page Source (HTML Source code) Print in Eclipse Console
3. Close the Browser using quit () command

(This time selenium will close both the windows)

**Assignment -2: (Browser Navigation Commands)**

1. Launch new Browser
2. Access the url: <https://www.google.com/> using **get (“URL”)** Command
3. Print the page title in Eclipse console ---i.e., google page title
4. Access the url: <https://www.facebook.com/> using **navigate ().to (“URL”)** Command
5. Print the page title in Eclipse console ---i. e, Facebook page title
6. Come back to Google page (Use ‘**Back’** command)
7. Again, go back to Facebook page (This time use ‘**Forward’** command)
8. Refresh the Browser (Use ‘**Refresh’** command)
9. Close the Browser

**Assignment -3: (Locators)**

**Part-1:**

1. Launch new Browser
2. Access the url: <https://parabank.parasoft.com/parabank/register.htm>
3. Locate the Web elements by using all the locators

(Id, ClassName, name, linkText, partialLinkText, tagName, Xpath )

**Note**: Use Selectors Hub add-on : To locate Web elements

1. Close the Browser

**Part-2:**

1. Launch new Browser
2. Access the url: <https://classic.freecrm.com/register/>
3. Locate the Web elements by using Relative xpath and Xpath functions

**Note**: **Note**: Use Selectors Hub add-on : To locate Web elements

1. Close the Browser

**Assignment -4: (Operations on Web elements—Click (), SendKeys (“”). --etc)**

**Part-1:**

1. Launch new Browser
2. Access the url: <https://parabank.parasoft.com/parabank/index.htm>
3. Click on “**Register**” button (Use xpath locator)
4. Print the current page title
5. Print the current url
6. Enter the below details (Use xpath / name locators)
7. First Name
8. Last Name
9. Address
10. City
11. State
12. Zip Code
13. Phone #
14. SSN
15. Password
16. Confirm
17. Capture the “**Signing up is easy!** Text
18. Close the browser

**Part-2: (Checkbox & Radio Button Operations)**

1. Launch new Browser
2. Access the url: <https://omayo.blogspot.com/>
3. Verify the display of **Home** Tab
4. Verify **Button1** is enabled or not under **Disabled Button** Section
5. Verify **Button2** is enabled or not under **Enabled Button** Section
6. Verify **Book** Check box is selected or not under **Select multiple options** Section
7. Select the **Pen** check box under **Select multiple options** Section
8. Verify Bag check box is selected or not
9. Verify **Bicycle** radio button is selected or not Under **Select a vehicle section**
10. Verify **Car** radio button is selected or not Under **Select a vehicle section**
11. Close the browser

**Assignment -5: (Find Elements & List Example)**

**Part-1:**

1. Launch new Browser
2. Access the url:  <https://www.myntra.com/>
3. Print all the links displayed in the page
4. Close the browser

**Part-2:**

1. Launch new Browser
2. Access the url:  <https://www.google.co.in/>
3. Enter “selenium” in Google input field
4. Print the suggestions displayed in suggestion list
5. Select any suggestion from the suggestion list
6. Close the browser

**Assignment -6: (Dropdown (i.e., Select Class) & Multiple Select Operations)**

**Part-1: (Select Class)**

1. Launch new Browser
2. Access the url: <https://register.rediff.com/register/register.php>?
3. Enter “Full Name”
4. Enter the “Choose a Rediffmail ID”
5. Enter “Password”
6. Print the maximum allowed length of “Password “field
7. Enter the below details

* Retype password
* Alternate Email Address
* Mobile No

1. Select the “Date of Birth”
2. Verify default gender selected
3. Select Female radio button
4. Print the all-country names from country drop down
5. Print default selected option in country drop down
6. Verify either country drop down is multi selected drop down or not
7. Select “Togo” from country drop down
8. Select “India” from country drop down
9. Select “Bangalore” from City drop down
10. Click on “Create my account” link
11. Close the browser

**Part-2: (Bootstrap Dropdown –i.e.,** **if there is no “select” tag)**

1. Launch new Browser
2. Access the url: <https://www.jquery-az.com/boots/demo.php?ex=63.0_2>
3. Click on “**Output”** dropdown
4. Print all the check boxes text
5. Select all the check boxes
6. Close the Browser

**Assignment -7: (Frame Handling)**

**Part-1:**

1. Launch new Browser
2. Access the url:  <https://the-internet.herokuapp.com/iframe>
3. Refresh the page
4. Enter some text in text area
5. Close the Browser

**Part-2:**

1. Launch new Browser
2. Access the url: <https://the-internet.herokuapp.com/nested_frames>
3. Switch to Left box
4. Switch to Middle box
5. Switch to Right box
6. Switch to Bottom box
7. Close the Browser

**Assignment -8: (Window handles)**

1. Launch new Browser
2. Access the url:  <https://netbanking.hdfcbank.com/netbanking/>
3. Capture current page title
4. Click on “[Register Now](javascript:void(0))” Link
5. Capture the current page title
6. Maximize the window
7. Switch back to previous window
8. Enter **Customer ID/ User ID**
9. Capture **Login to Net Banking** text
10. Close the all opened windows

**Assignment -9: (Different alert popups)**

**Part-1: (Popup with “OK” button)**

1. Launch new Browser
2. Access the url [**https://mail.rediff.com/cgi-bin/login.cgi**](https://mail.rediff.com/cgi-bin/login.cgi)
3. Click on “Go” Button
4. Print the message in popup
5. Click on Ok button
6. Close the browser

**Part-2: (Popup with “OK” and “Cancel” button)**

1. Launch new Browser
2. Access the url [**http://the-internet.herokuapp.com/javascript\_alerts**](http://the-internet.herokuapp.com/javascript_alerts)
3. Click on “Click for JS Confirm” button
4. Print the message in the Popup
5. Dismiss the alert popup (Clicking on cancel button)
6. Again “Click on “Click for JS Confirm” button
7. Accept the alert popup (Clicking on OK button)
8. Close the browser

**Part-3: (Popup with “OK”,” Search Box” and “Cancel” button)**

1. Launch new Browser
2. **Access the url** [**http://the-internet.herokuapp.com/javascript\_alerts**](http://the-internet.herokuapp.com/javascript_alerts)
3. Click on “Click for JS Prompt” button
4. Print the message in the Popup
5. Dismiss the alert popup (I.e., Clicking on cancel button)
6. Again “Click on “Click for JS Prompt” button
7. Enter some text (i.e., Suresh) in search box
8. Accept the alert popup (i.e., Clicking on OK button)
9. Verify the text entered in step no:6 is displayed under Result: (like **You entered: Suresh)**
10. Close the Browser

**Part-4: (Authentication popup)**

1. Launch new Browser
2. Access the url: [**http://the-internet.herokuapp.com/basic\_auth**](http://the-internet.herokuapp.com/basic_auth)
3. Handle the popup by passing user name and password (admin/admin) in url itself
4. (Format: http://username:password@test.com)
5. Verify the message in logged in page
6. Close the Browser

**Assignment -10: (Action Class)**

**Part-1: (Mouse Hover)**

1. Launch new Browser
2. Access the url: <https://book.spicejet.com/RetrieveBooking.aspx?AddMeal=true>
3. Mouse hover on the “Login / Signup” tab
4. Click on “**SME Travellers**” option
5. Switch to SME Travellers page
6. Click on “**SIGNUP”** Link
7. Switch to “**Signup”** Page
8. Fill all the information in **Registration Form**
9. Click on “submit” button
10. Close the Browser

**Part-2: (Drag and Drop)**

1. Launch new Browser
2. Access the url: <https://jqueryui.com/droppable/>
3. Perform Dragan drop operations

**I.e.,** Move “Drag me to my target” Element to “Drop here” Element

1. Close the Browser

**Part-3: (Right Click)**

1. Launch new Browser
2. Access the url: <https://swisnl.github.io/jQuery-contextMenu/demo.html>
3. Right Click on “**right click me**” button
4. Click on Delete option
5. Handle the alert popup
6. Close the Browser

**Part-3: (Double Click)**

1. Launch new Browser
2. Access the url: <https://api.jquery.com/dblclick/>
3. Scroll down the page
4. Capture the color of **Demo: block ( Output : rgba value of blue )**
5. Double click on “**Demo: block**
6. Capture the color of **Demo: block ( Output : rgba value of yellow )**
7. Close the Browser

**Assignment -11: (Keyboard operations)**

**Part-1:**

1. Launch new Browser
2. Access the url: <https://www.google.com/advanced_search>
3. Enter the text in “all these words:” field
4. Perform Tab operation from Keyboard
5. Enter the text in “this exact word or phrase:” field
6. Select & copy the entered text in “this exact word or phrase’ field using keyboard operation
7. Perform Tab operation from Keyboard
8. Paste above copied text into “any of these words:” field using keyboard operation
9. Close the Browser

**Part-2: (Screen shot)**

1. Launch new Browser
2. Access the url: <https://ttdsevaonline.com/#/login>
3. Take the screenshot of the page
4. Click on “**SignUp**” button
5. Take the screenshot of the page
6. Close the Browser

**Assignment -12: (File Upload)**

**Part-1: (type=’file’--🡪 Using SendKeys(“”) )**

1. Launch new Browser
2. Access the url:  <http://sme.spicejet.com/Airline/SMERegister>
3. Enter all mandatory fields
4. “Upload Image” in **Upload pan card** Field
5. Close the Browser

**Part-2: (if type! =’file’---Use “Robot” Class)**

1. Launch new Browser
2. Access the url: <https://imgbb.com/>
3. Click on **Start uploading** Button
4. Upload file using **Robot Class**
5. Close the Browser

**Assignment -13: (File Download)**

**Part-1: (Using “Click()”)**

1. Launch new Browser
2. Access the url:  <https://file-examples.com/index.php/sample-documents-download/sample-xls-download/>
3. Download any of the file ( Ex: 10 rows Excel File )
4. Close the Browser

**Part-2: (Using “Browser profile settings & MIME Type”) -🡪 Out dated :No need to do**

1. Launch new Browser
2. Access the url: <https://file-examples.com/index.php/sample-documents-download/sample-xls-download/>
3. Download “climate\_data.xls” file using profile settings & MIME Type

**Note**: Refer the url: <https://www.freeformatter.com/mime-types-list.html>   for MIME Types

1. Close the Browser

**TestNG**

**Pre-Requisites:** Install TestNG from “Eclipse Market place” In Eclipse IDE

**-**----------------------------------------------------------------------------------------------------------------------

**Assignment -14: (TestNg Annotations)**

**Part-1:**

* Explore with all TestNG Annotations

**Part-2:**

* Verify the Execution sequence of TestNG Annotations

**Assignment -15: (About Attributes in @Test Method)**

* Explore “Lists of attributes that we can pass to Test method:”
* **priority**----------------------**Eg:** @Test (priority =1)
* **dependsOnGroups** ----- **Eg:** @Test (groups = “City”) --group name is anything
* **dependsOnMethods**----**Eg**: @Test (dependsOnMethods = “OpenBrowser”)
* description----------------**Eg:** @Test (description = “test method”)
* invocationCount---------**Eg**: @Test (invocationCount = 7)
* invocationTimeOut----- **Eg**: @Test (invocationCount =7, invocationTimeOut = 30)
* alwaysRun ---------- -----**Eg**: @Test (alwaysRun = true)
* dataProvider ------------- **Eg**: @Test (dataProvider = “getData”). ---“getData” @DataProvider method name

**Assignment -16: (Validations—Using “Assert” Class)**

**Part-1:**

* Explore all the methods available in “Assert” Class

**Part-2:**

1. Launch new Browser
2. Access the url:  <https://www.google.com/>
3. Get the page title
4. Validate the page title using Assert class
5. Close the Browser

**Assignment -17: (Validations—Using “Assert” and “SoftAssert”)**

* Do the “**Assignment -4”** using TestNg

**Note:**

* Create “Part-1 & Part-2” under different @Test methods
* Use TestNgAssertions for validations
* Use SoftAssert and observe the difference between “HardAssert & SoftAssert”

**Assignment -18: (“Test Suite” Creation)**

* Do the “**Assignment -7”** using TestNg
* Create test suite (**Assignment -17** + **Assignment -18**)
* Generate “testng.xml” file
* Execute as “TestNg suite”
* Verify the test output

Note: For validations use Assertion concept

**Assignment -19: (“Cross Browser” Execution---Using “Parameter” tag in textng.xml file)**

i.e., about **@Parameters** Annotation

* Do the “**Assignment -8”** using TestNg
* Generate testng.xml file
* Execute the script in “Chrome and Firefox” browser using “Parameter” tag in xml file

Note: For validations use Assertion concept

**Assignment -20: (Parameterization using “@DataProvider” Annotation)**

**Part-1: (Excel Reader---using “apache POI” jars)**

* Store some data into excel sheet and read
* Print the data in Eclipse Console

**Part-2: (DataProvider)**

1. Launch new Browser
2. Access the url: <https://demo.applitools.com/index.html>
3. Store “username” and “pwd” of 2 different users in Excel
4. Read data from Excel
5. Enter username and Pwd using DataProvider concept
6. Click on Login button
7. Close the browser

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

**Assignment -21: (Framework Design)**

Hybrid Framework (Using “Data Driven “+ “Page Object Model”)

**Part-1:**

* Explore “How to read properties files” using java properties class

**Part-2:**

* Explore about ANT
* Explore about Build.xml
* Explore about “How to generate build.xml file”

**Part-3:**

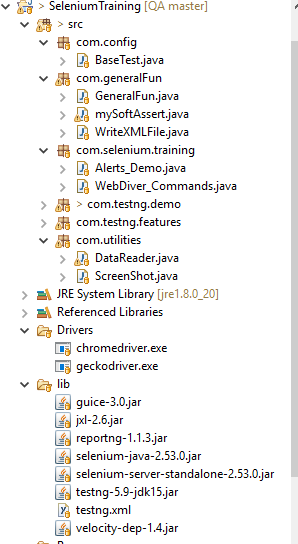
* Implement the **Framework** as mentioned in “Framework Structure “image
* Explore about executing and generating reports using build.xml

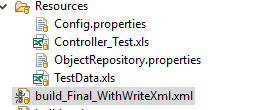
------------------------------------------------------------------------------------------------------------------------------------------

**Required code files for Frame work design:**

* mySoftAssert.java
* WriteXMLFile.java
* DataReader.java
* ScreenShot.java
* Controller\_Test.xls
* build\_Final\_WithWriteXml.xml

**Framework Structure**





**Best of luck 🙂 Happy learning and Happy coding.**

Suresh Reddy Kadenti 😊

