Practical No. 03

Aim: Install Selenium server and demonstrate it using a script in Java / PHP.

Solution:

Part 1) Installation of Selenium RC & Eclipse

Download Eclipse

Step 1: Go to URL - http://www.eclipse.org/downbads/

Step 2: Select Eclipse IDE for Java Developers (Click on Windows 32 bit platform).

Step 3: Click on OK button and save to a local drive (i.e. C: or D: ; etc)

Step 4: Unzip the downloaded zip file & rename that to Eclipse.

Step 5: Create one more folder "Eclipse - Workspace" (i.e. C: Eclipse - Workspace) in the same drive where Eclipse is unzipped and renamed

Step 6: Create Eclipse desktop shortcut (go to C: Eclipse folder -> right click Eclipse.exe and then click on "desktop create shortcut").

1. Now we need to create a workspace folder -> C: Eclipse - Workspace Selenium Tests
2. Double click on "Eclipse shortcut on Desktop"
3. This opens the Eclipse
4. Close Eclipse welcome screen
5. Click file menu -> Launch Workspace -> other
6. Now select the C: Eclipse - Workspace Selenium Tests folder.

We have finished setting up the eclipse.

Sundaram FOR EDUCATIONAL USE

Part 2)

Now, we need to download Selenium RC server/ client driver and configure that to Eclipse.

Step 1: Download Selenium server: http://seleniumhq.org/download/

Step 2: Download Selenium Client driver for Java (from Selenium Client Drivers section)

Step 3: Create "Selenium" folder in C: drive and copy the Selenium-server.jar as well as unzip the Selenium Client driver (C: Selenium)

Downloading and unzipping the files into a folder is done

We need to configure the appropriate Selenium Client driver Jar file to the Eclipse.

~~Step 1~~:

Part 3)

Step 1: Go to Eclipse -> Click file -> New -> Project (from various options need to select just "project")

Step 2: In Select Wizard -> Click Java -> "Java Project"

Step 3: Give the project name (e.g. Demo Tests)

Step 4: Click Finish - Click Yes

Step 5: Now we are done with Creation of project and need to configure the Selenium Client driven to this Project.

Step 6: Right Click "DemoTests" project

Step 7: Click "Java Build Path"

Step 8: Click Libraries tab

Step 9: Click "Add External JARs" button

Step 10: Select "Selenium Client Drivers" unzipped in C: Selenium folder (Selenium Server JAR file should not be added)

Sundaram
FOR EDUCATIONAL USE

Step 11 : Click Ok

Step 12 : Reference libraries → contains both the Selenium Client driver jar files.

Step 13 : Create a new class file as "SeleniumDemo" in the "DemoTest" by right click on src folder.

Step 14 : Copy the below code in the class file :-

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

public class hello
{
    Static String driverPath = "D:\\selenium pracs\\gekodriver-
                                v0.21.0-win32\\GeckoDriver.exe";
    public Static WebDriver driver;
    public Static void main (String args[])
    {
        int a =10, b=20;
        System.out.println(" Hi... ");
        System.out.println(a+b);
        System.out.println("Selenium demo.... ");

        System.setProperty("webdriver.gecko.driver", driverPath);
        DesiredCapabilities capabilities = DesiredCapabilities.firefox();
        capabilities.setCapabilities("marionette", true);
        driver = new FirefoxDriver (capabilities);
        driver.get ("https://www.facebook.com/");
        driver.manage().window().maximize();
```

Sundaram
FOR EDUCATIONAL USE

```
        driver.quit();
    }
}
```

Step 15: Now we run the program.

Output:

Practical No. 04

Aim: Write and test a program to login a specific web page. (Using JUnit)

Solution:-

PART 1) Download JUnit

JUnit Environment Setup

Step 1) You need to set JUNIT_HOME environment variable to point out the base location where you have placed JUnit Jars.

For example, if you have created a JUnit folder in c: drive and placed jars there, then for environment settings you need to open control panel -> advanced -> environment variable.

1. Under environment window clicks on "new" button.
2. When you click on new button in environment variables, it will open another window.

Step 2)

3. Step 2) A "New System Variable" window will open.
4. Provide variable name as "JUNIT_HOME".
5. Provide JUNIT value as JUNIT path where you have copied JUNIT jar files.
6. Click on OK.

When you click on OK, it will create a new System variable with the given name and value.

Step 3) After creating JUNIT_HOME, create another variable with the name CLASSPATH. Again go to Environment Variables and follow the below steps.

1. Click on "new" button
2. When you click on new in environment variables, it will open another window.

Sundaram FOR EDUCATIONAL USE

Step 4) In this step, point out JUNIT_HOME to JUnit.jar which is placed in JUNIT folder as given below:

1. Variable Name: JUNIT_HOME
2. Variable Value: %CLASSPATH%;%JUNIT_HOME%\JUnit4.10.jar;
3. Click on the OK button

Step 5) Once you click on the 'OK' button, you can verify that a new environment variable named "CLASSPATH" can be seen under system variable.

Part 2) Install JUnit jar file in eclispse

Step 1) Right click on project:

1. Click on "build path" and then
2. Click on "Configure build path".

Step 2) In this Step,

1. Go to java build path window as shown in below figure
2. Now click on "Add External JARs" button to add your downloaded JUnit.jar file with eclipse.

After adding a JUnit.jar file, click on 'OK' button to close java build path window.

Part 3) Verifying whether required jar file for JUnit is in my build path

In order to verify JUnit jar file in eclipse, you need to follow below-mentioned.

Steps:

1. Right click on project -> Build Path
2. Click on "Configure build path".
3. Java build path window will appear as shown below.
4. In that window, go to Libraries tab to see all jar file tree view, you need to look for the jar file name

Sundaram
FOR EDUCATIONAL USE

which is starting with JUnit.

5. Once you expand JUnit libraries, you can see java libraries.

6. Now you are ready to use JUnit with eclipse

Part 4) Verify JUnit Setup

7. You can create a simple JUnit test to verify JUnit setup. See below test case:

Step 1)

8. Create a java class named TestJUnit.java and provide a simple assert statement.

```
import org.junit.Test;
import static org.junit.Assert.assertEquals;
public class TestJunit {
  @Test
  public void testSetup(){
    String str = "I am done with JUnit setup";
    assertEquals("I am done with Junit setup", str);
  }
}
```

Step 2) Create a Test Runner class to execute above test

```
import org.junit.runner.JUnitCore;
import org.junit.runner.Result;
import org.junit.runner.notification.Failure;

public class TestRunner {
  public static void main(String [] args){
    Result result = JUnitCore.runClasses(TestJunit.class);
    for (Failure failure : result.getFailures()){
```

Sundaram
FOR EDUCATIONAL USE

```
            System.out.println(failure.toString());
        }
        System.out.println("Result =="+ result.wasSuccessful());
    }
}
```

Step 3) to execute the test, follow below steps:

1. Right click on TestRunner.java and click on "Run As" as shown below
2. Another window will be open once you click on "Run As", click on "1 JUnit Test"

Step 4) Here is the output or result for your test. If it is successfully executed, it will show a green check mark in front of it.

Sundaram

FOR EDUCATIONAL USE

Practical No. 05

Aim: Write & test a program to update 10 student records into table into Excel file (using TestNG).

Solution :

Step 1: Installing TestNG in eclipse

1. Select Help / Software updates / Find and Install.
2. Search for new features to install.
3. New remote site.
4. For Eclipse 3.4 and above, enter http://beust.com/eclipse.
5. For Eclipse 3.5 and below, enter http://beust.com/eclipse1.
6. Make sure the check box next to URL is checked and click Next.
7. Eclipse will then guide you through the process.

Step 2: Create 2 Excel sheet.

1. ~~Excet~~ Open Excel Sheet → File → Save As → Name: Book1 → select xls: Exel 97-2003 Workbook → Finish
2. Enter details in Book 1 Exel sheet.
3. Open Excel sheet → File → Save As → Desktop → Name: Book2 → Select xls: Exel 97-2003 Workbook → Finish.

Sundaram

FOR EDUCATIONAL USE

Step 3: For Adding Jar Files

1. Right click on project:
2. Click on "build path" and then
3. Click on "Configure build path".
4. In that window, go to Libraries tab to see all jar files.
5. Now click on "Add External JARs" button
6. Then Select Selenium and geckodriver
7. Apply and Then OK
8. Add libraries from lib folder.

Step 4: Program.

```
import jxl. Cell;
import jxl. Sheet;
import jxl. Workbook;
import jxl. read. biff. BiffException;

//Code to update 10 student records into the table into
  Excel files.
import org. testng. annotations. BeforeClass;
import org. testng. annotations. Test;
import jxl. *;
import jxl. read. *;
import jxl. write. *;
import java. io. *;

public class updatestudrecords {
    @ Before Class
    public void f1() { }
```

Sundaram
FOR EDUCATIONAL USE

```
@ Test
public void testImportexport1() throws Exceptions {
 FileInputStream fi = new FileInputStream("D:\\selenium
                                pracs\\Book1.xls");
 Workbook w = Workbook.getWorkbook(fi);
 Sheet s = w.getSheet(0);
 String a[][] = new String[s.getRows()][s.getColumns()];
 FileOutputStream fo = new FileOutputStream("D:\\
                                Selenium pracs\\Desktop\\Book2.
                                xls");
 WritableWorkbook wwb = Workbook.createWorkbook(fo);
 WritableSheet ws = wwb.createSheet("result 1",0);
 for (int i=0; i<s.getRows(); i++)
    for(int j=0, j<s.getColumns(); j++)
    {
       a[i][j] = s.getCell(j,i).getContents();
       Label l2 = new Label(j, i, a[i][j]);
       ws.addCell(l2);
       Label l1 = new Label(6,0,"Result");
       ws.addCell(l1);
    }
 for (int i=1; i<s.getRows(); i++) {
    for (int j=2; j<s.getColumns(); j++)
    {
       a[i][j] = s.getCell(j,i).getContents();
       int x = Integer.parseInt(a[i][j]);
       if(x>35)
       {
          Label l1 = new Label(6,i,"pass");
          ws.addCell(l1);
       }
```

Sundaram
FOR EDUCATIONAL USE

```
                else
                {
                    label l1 = new label (6, i, "fail");
                    ws. addCell (l1);
                    break ;
                }
            }
        }

        wwb. write ();
        wwb. close ();
    }
}
```

Step 5: Add path of Book1.xls in FileInputStream and add path of Book2.xls in FileOutput-Stream.

Step 6: Click on "Run as TestNG Suite" and your test will start running.

Sundaram

FOR EDUCATIONAL USE

Practical No. 06

Aim: Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).

Solution:

Step 1: Create 2 Excel Sheet

1. Open Excel Sheet → File → Save As → Select Desktop → Name : Book 11 → select xls : Exel 97-2003 Workbook → Finish.
2. Enter the details of Student in Book 11 Exel sheet.
3. Open Excel Sheet → File → Save As → Select Desktop → Name: Book 12 → select xls: Exel 97-2003 Workbook → finish.

Step 2: For New project.

1. Open File → New project → Name : practical6 → Finish.

~~2. For~~

Step 3: For Adding Jar files

1. Right Click on Project
2. Click on " build path " and then
3. Click on " Configure build path "
4. In that window, go to Libraries tab to See all jar files.
5. Now Click on " Add External JARS " button.
6. Then Select Selenium and geckodriver
7. Apply and Then Ok.

FOR EDUCATIONAL USE

Sundaram

Step 4: For Creating class to write program.

1. ~~Sel Right~~ Click on Project.
2. Right Click on Src Folder. → New → class
   → Name: countstuds → Finish

Step 5: 1. Create a lib folder in project
2. Copy all libraries and paste in lib folder in your project

Step 6: Program.

```
import jxl.*;
import jxl.read.*;
import jxl.write.*;
import java.io.*;
import org.testng.annotations.Test;
public class countstuds {
    @Test
    public void testImportexport1() throws Exception {
        FileInputStream f1 = new FileInputStream("D:\\
                                Selenium pracs\\Desktop\\Book11.xls");
        Workbook w = Workbook.getWorkbook(f1);
        Sheet s = w.getSheet(0);
        String a[][] = new String[s.getRows()][s.getColumns()];
        FileOutstream fo = new FileOutputStream("D:\\Selenium
                                pracs\\Desktop\\Book12.xls");
        Writeable Workbook wwb = Workbook.createWorkbook
                                (fo);
        Writeable Sheet ws = wwb.createSheet("result", 0);
        int c=0;
```

Sundaram®
FOR EDUCATIONAL USE

```
for (int i=0; i<s.getRows();  i++) {
  for (int j=0; j<s.getColumns(); j++) {
      if (i>=1)
      {
          String b = new String();
          b=s.getCell(3, i).getContents();
          int x = Integer.parseInt(b);
          if (x<60)
          {
              c++;
              break;
          }
      }
          a[i][j] =s.getCell(j,i).getContents();
          Label l2 = new Label(j, i-c, a[i][j]);
          ws.addCell(l2);
      }
  }
  wwb.write();
  wwb.close();
}
}
```

Step 7: Add path of Book11.xls in FileInputStream and add path of Book12.xls in ~~Out~~ FileOutputStream.

Step 8: Click on "Run as TestNG Suite" and your test will start running.

Sundaram

FOR EDUCATIONAL USE

Practical No. 07

Aim: Write and test a program to provide total number of links available on the page.

Solution:

Step 1: Create a new project, name: nlinks.

Step 2: For Adding Jar Files.

1. Right Click on Project
2. Click on "build path" and then
3. Click on "Configure build path"
4. In that window, go to Libraries tab to see all jar files.
5. Now Click on "Add External JARs" button.
6. Then select Selenium and geckodriver
7. Select Apply and then ok.

Step 3: Create a lib folder in project.

1. Go to File Explorer.
2. Open the lib folder.
3. Copy all the libraries
4. And paste in the lib folder of project.

Step 4: Create class

1. Select project
2. Right Click on src → New → Class → Name: nlinks → Next → Finish.

Step 5: Program

Sundaram
FOR EDUCATIONAL USE

```
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.remote.DesiredCapabilities;

public class nlinks {
  Static String driverPath = "D:\\Selenium pracs\\geckodriver
                  - v0.21.0-win32\\Geckodriver.exe";
  public static WebDriver driver;
  public static void main(String args[])
  {

    System.setProperty("webdriver.gecko.driver", driverPath);
    DesiredCapabilities capabilities = DesiredCapabilities.firefox();
    capabilities.setCapabilities("marionette", true);
    driver = new FireFoxDriver(capabilities);
    driver.get("http://toolsqa.wpengine.com/");

    java.util.List<WebElement> links =
    driver.findElements(By.tagName("a"));
    System.out.println("Total links are"+links.size());
    for (int i=0; i< links.size(); i=i+1)
    {
      System.out.println("Link"+i+" link name"+links.get(i).
                                    getText());
    }
  }
}
```

Step 6: Run the code

Sundaram
FOR EDUCATIONAL USE

Practical No. 8

Aim: Write and test a program to get the number of items in a list / combo box.

Solution:

Step 1: Create a new project

Step 2: For Adding Jar Files

1. Right Click on Project
2. Click on "build path" and then
3. Click on "Configure build path"
4. In that window, go to Libraries tab to see all ~~the~~ jar files
5. Now Click on "Add External JARS" button.
6. Then Select Selenium and geckodriver.
7. Select Apply and then ok.

Step 3: Create a lib folder in project

1. Go to file Explorer
2. Open the lib folder
3. Copy all the libraries
4. And paste in the lib folder of project.

Step 4: Create class

1. Select project
2. Right Click on src → New → Class → Name: list → Next → Finish.

Sundaram

FOR EDUCATIONAL USE

Part A) For the list Program

Step 5: Create html page code

list.html

```
<html>
<body>
<ol>
<li> Coffee 1 </li>
<li> Tea 2 </li>
<li> Milk 3 </li>
</ol>
<ul>
 <li> Coffee </li>
 <li> Tea </li>
 <li> Milk </li>
</ul>
</body>
</html>
```

Step 6: Program Code

```
import org.openga.selenium.By;
import org.openga.selenium.WebDriver;
import org.openga.selenium.WebElement;
import org.openga.selenium.firefox.FirefoxDriver;
import org.openga.selenium.remote.DesiredCapabilities;

public class no_of_list {
    Static String driverPath = "C:\\SW\\ST-SW\\geckodriver-v0.
                                11.1-win32.zip";
    public static void main (String [] args) {
```

FOR EDUCATIONAL USE

Sundaram

```
System.setProperty("webdriver.gecko.driver", driverPath);
DesiredCapabilities capabilities = DesiredCapabilities.firefox();
Capabilities.setCapabilities("marionette", true);
driver = new FirefoxDriver(capabilities);
~~driver = new firefoxDriver(co~~
driver.get("C:\\users\\user\\Desktop\\Sanika\\list.html");
java.util.List <WebElement> list = driver.findElements
                                    (By.tagName("li"));
for (int i=0; i<list.size(); i=i+1)
{
    System.out.println("List Item " + i + "Item Name="
                    + list.get(i).getText());
}
}
}
```

Step 6: Run the list code.

Part B) For the combo Box Program.

Step 1: Continue the Step 1 to Step 4.

Step 2: Create html page code
comboBox.html

```
<html>
<body>
<select name="Cars" id="Cars">
  <option value="volvo">volvo</option>
  <option value="saab">saab</option>
```

Sundaram
FOR EDUCATIONAL USE

```
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
</select>
</body>
</html>
```

Step 3: Program Code for Combo_box

```
import org.openga.selenium.By;
import org.openga.selenium.WebDriver;
import org.openga.selenium.WebElement;
import org.openga.selenium.firefox.FirefoxDriver;
import org.openga.selenium.remote.DesiredCapabilities;

public class no_of_combo_box {
    Static String driverPath = "C:\\SW\\ST_SW\\geckodriver-
                    V0.11.1-win32.zip";
    public Static WebDriver driver;
    public Static void main (String [] args) {
        System.setProperty ("webdriver.gecko.drivers",
                                driverPath);
        DesiredCapabilities ("marionette", true);
        driver = new FirefoxDriver (capabilities);
        driver.get("C:\\Users\\user\\Desktop\\Sanikall
                                combo_box.html");
        java.util.List<webElement> list = driver.findElements (By.
                                tagName ("option"));
        for (int i=0 ; i< list.size (); i=i+1) {
            System.out.println ("list Item "+i+ "Item Name = "+
                                list.get(i).getText ());
    }}}
```

Sundaram
FOR EDUCATIONAL USE

Step 4: Run the code.

Practical No. 9

Aim: Write and test a program to count the number of check boxes on the page

Solution:

Step 1: Create a new project
Step 2: For Adding Jar files
1. Right Click on Project
2. Click on "build path" and then
3. Click on "Configure build path"
4. In that window, go to Libraries tab to see all jar files
5. Now Click on "Add External JARs" button.
6. Then Select Selenium and geckodriver.
7. Select Apply and then ok.

Step 3: Create a lib folder in project
1. Go to file Explorer
2. Open the lib folder
3. Copy all the libraries
4. And paste in the lib folder of project.

Step 4: Create Class
1. Select project
2. Right Click on src → New → Class → Name: .checkbox → Next → finish

Part A) For the ~~combo~~ Checkbox Program.
Step 1: Create html page code
check_box.html.

Sundaram
FOR EDUCATIONAL USE

```
<html>
<body>
<input type="checkbox" id="vechicle1" name="vechicle1"
                                value="Bike">
<label for="vechicle 1"> I have a bike </label><br>
<input type="checkbox" id="vechicle2" name="vechicle2" value="Car">
<label for="vechicle 2"> I have a Car </label> <br>
<input type="checkbox" id="vechicle 3" name="vehicle 3" value="boat">
<label for "vechicle 3" > I have a boat </ label> <br>
</ body>
</ html>
```

Step 2: Program for Checkbox

```
import org. openga . selenium. By;
import org. openga. selenium . WebDriver;
import org. openga . Selenium. WebElement;
import org. openga . selenium. firefox . firefox Driver;
import org. openga . Selenium. remote. Desired Capabilities;

public class no_of_checkbox {
  Static String driver Path = " C: \\ SW \\ ST_ SW \\
                    geckodriver_V0.11.1-win32.zip";
  public Static void main (String [] args){
    System . setProperty ("webdriver . gecko . driver",
                                  driver Path);
    Desired Capabilities  capabilities = Desired Capabilities .
                                  firefox ();
    capabilities . setCapabilities ("marionette", true);
    driver = new firefox Driver ( capabilities);
    driver . get (" C: \\ users \\ user \\ Desktop \\
                                  Check_box. html");
```

FOR EDUCATIONAL USE

```
java.util.list <webElement> list = driver.findElements
                                    (By.tagName("label"));
for (int i=0; i< list.size(); i=i+1)
{
    System.out.println("List Item" + i + " Item
                                    Name = " + list.get(i).getText());
    }
  }
}
```

Step 3: Run the code.

Part B) Code for ~~and~~ No. of Radios

Step 1: For Part (B) follow the Step 1 to Step 4.

Step 2: Create the html page code.

radio.html

```
<html>
```

Sundaram®

FOR EDUCATIONAL USE