SOFTWARE TESTING

LABORATORY MANUAL

VI Semester Computer Science & Engineering

2011-12

BY

Mr.RAHUL A.PALAKAR

LECTURER

COMPUTER SCIENCE DEPARTMENT

B.T.PATIL & SONS POLYTECHNIC

Stavanidhi-Nipani. Dist: BELGAUM

FOR ANY FEEDBACK CONTACT TO
Email : rahulpalakar403@gmail.com

Vijay H. Kalmani, M.Tech(CS&E), (Phd), LMISTE
Principal,
B.T.Patil & Sons Polytechnic, Stavanidhi, Nipani, Dist: Belgaum
vijaykalmani@gmail.com

EXPT NO :-1

Understand The Automation Testing Approach (Theory Concept)

Automation

Automation is making a process automatic eliminating the need for human intervention. It is a self-controlling or self-moving process. Automation Software offers automation wizards and commands of its own in addition to providing a task recording and re-play capabilities. Using these programs you can record an IT or business task.

Benefits of Automation

Fast

Reliable

Repeatable

Programmable

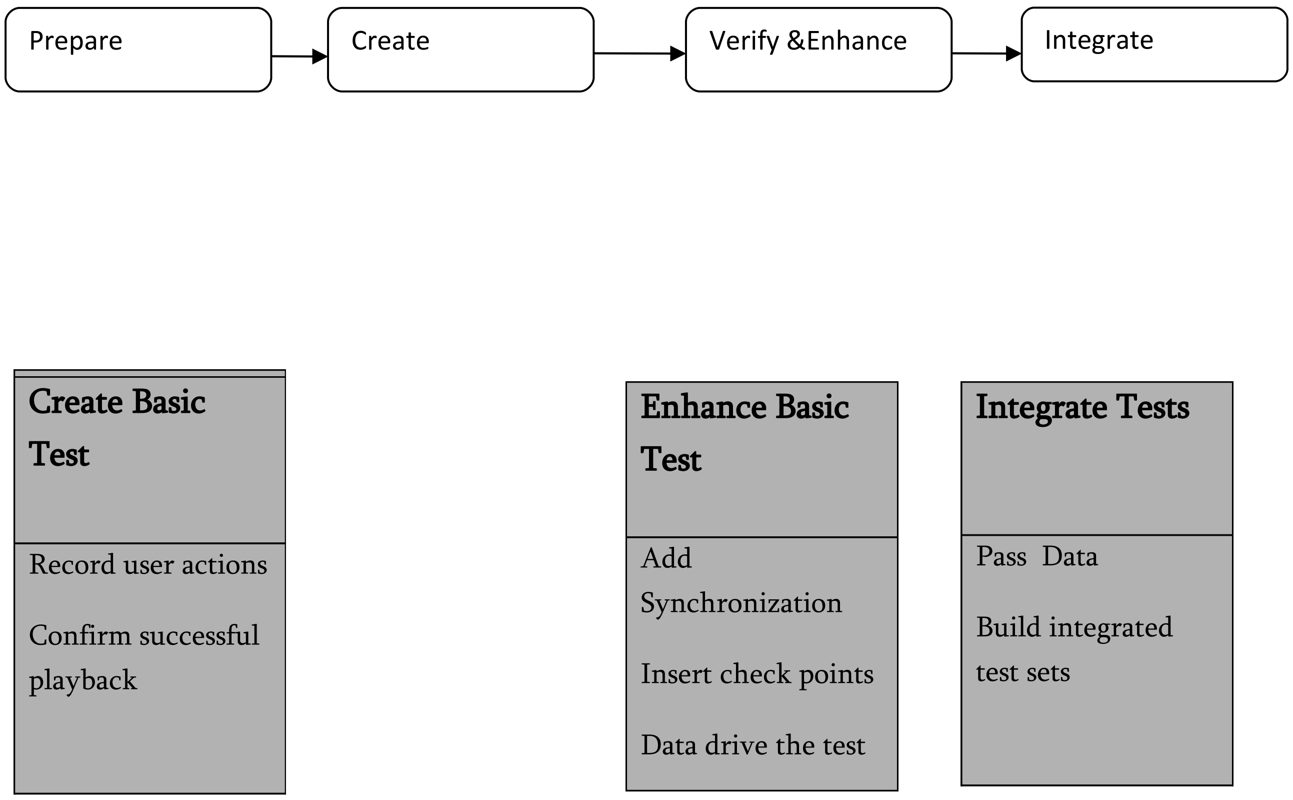
Reusable

Makes Regression testing easy

Enables 24*78 Testing

Robust verification.

Automation Test Workflow



INTRODUCTION TO SELENIUM

1. History of Selenium

- In 2004 invented by Jason R. Huggins and team.
- Original name is JavaScript Functional Tester [JSFT]
- Open source browser based integration test framework built originally by Thoughtworks.
- 100% JavaScript and HTML
- Web testing tool
- That supports testing Web 2.0 applications
- Supports for Cross-Browser Testing(ON Multiple Browsers)
- And multiple Operating Systems
- Cross browser – IE 6/7, Firefox .8+, Opera, Safari 2.0+

2. What is Selenium?

- Acceptance Testing tool for web-apps
- Tests run directly in browser
- Selenium can be deployed on Windows, Linux, and Macintosh.
- Implemented entirely using browser technologies -
    - JavaScript
    - DHTML
    - Frames

3. Selenium Components

- Selenium IDE
- Selenium Core
- Selenium RC

– Selenium Grid

3.1 Selenium IDE

- The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases.
- It is Firefox plug-in
- Firefox extension which allows record/play testing paradigm
- Automates commands, but asserts must be entered by hand
- Creates the simplest possible Locator
- Based on Selenese

3.1.1 Overview of Selenium IDE:

A. Test Case Pane

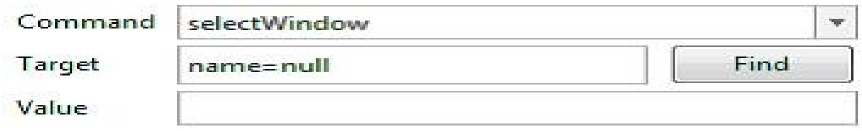
B. Toolbar

C. Menu Bar

D. Log/Reference/UI-Element/Rollup Pane

A. Test Case Pane:

- Your script is displayed in the test case pane.
- It has two tabs.
- one for displaying the command (source)
- and their parameters in a readable "table" format.



Command	selectWindow
Target	name=null
Value	

B. Toolbar: The toolbar contains buttons for controlling the execution of your test cases, including a step feature for

C. **Menu Bar:**

- **File Menu:** The File menu allows you to create, open and save test case and test suite files.
- **Edit Menu:** The Edit menu allows copy, paste, delete, undo and select all operations for editing the commands in your test case.
- **Options Menu:** The Options menu allows the changing of settings. You can set the timeout value for certain commands, add user-defined user extensions to the base set of Selenium commands, and specify the format (language) used when saving your test cases.

D. **Help Menu:**

Introducing Selenium Commands

The command set is often called selenese. Selenium commands come in three “flavors”:
Actions, Accessory and **Assertions.**

a. **Actions:** user actions on application / Command the browser to do something.

Actions are commands that generally manipulate the state of the application.

1. Click link- click / Clickandwait
2. Selecting items

b. **Accessors**: Accessors examine the state of the application and store the results in variables, e.g. "storeTitle".

c. **Assertions:** For validating the application we are using Assertions

1. For verifying the web pages
2. For verifying the text
3. For verifying alerts

Assertions can be used in 3 modes:

- assert
- verify
- waitFor

Example: "assertText","verifyText" and "waitForText".

NOTE:

1. When an "assert" fails, the test is aborted.
2. When a "verify" fails, the test will continue execution
3. "waitFor" commands wait for some condition to become true

Commonly Used Selenium Commands

These are probably the most commonly used commands for building test.

open - opens a page using a URL.

click/clickAndWait - performs a click operation, and optionally waits for a new page to load.

verifyTitle/assertTitle - verifies an expected page title.

verifyTextPresent- verifies expected text is somewhere on the page.

verifyElementPresent -verifies an expected UI element, as defined by its HTML tag, is present on the page.

verifyText - verifies expected text and it's corresponding HTML tag are present on the page.

verifyTable - verifies a table's expected contents.

waitForPageToLoad -pauses execution until an expected new page loads. Called automatically when clickAndWait is used.

waitForElementPresent -pauses execution until an expected UI element, as defined by its HTML tag, is present on the page.

3.1.2 Recording and Run settings

When Selenium-IDE is first opened, the record button is ON by default.
During recording, Selenium-IDE will automatically insert commands into your test case based on your actions.

a. **Remember Base URL MODE** - Using Base URL to Run Test Cases in Different Domains

b. **Record Absolute recording mode** – Run Test Cases in Particular Domain.

3.1.3 Running Test Cases

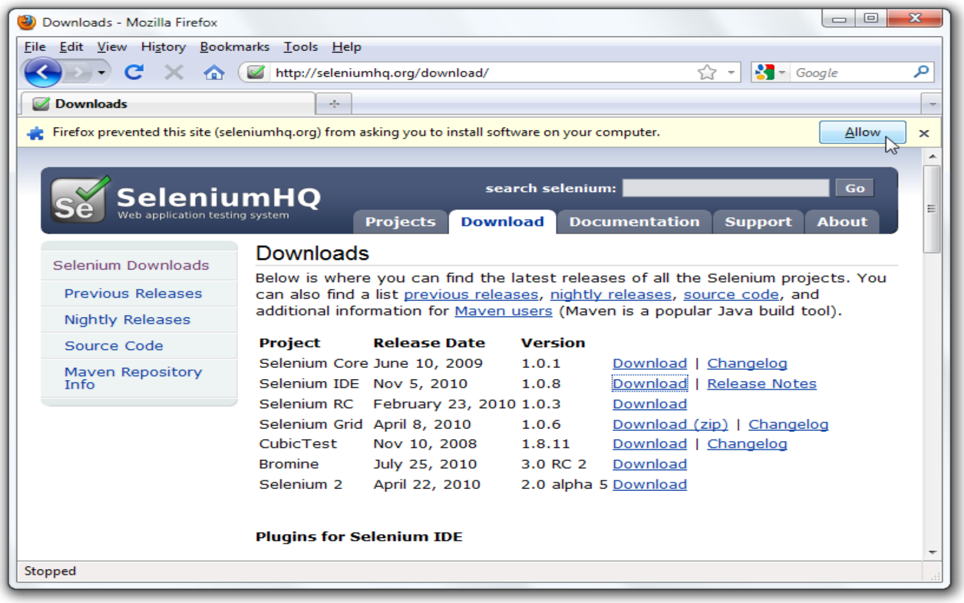
Run a Test Case Click the Run button to run the currently displayed test case.
Run a Test Suite Click the Run All button to run all the test cases in the currently loaded test suite.
Stop and Start The Pause button can be used to stop the test case while it is running. The icon of this button then changes to indicate the Resume button. To continue click Resume.
Stop in the Middle You can set a breakpoint in the test case to cause it to stop on a particular command. This is useful for debugging your test case. To set a breakpoint, select a command, right-click, and from the context menu select Toggle Breakpoint.
Start from the Middle You can tell the IDE to begin running from a specific command in the middle of the test case. This also is used for debugging. To set a startpoint, select a command, right-click, and from the context menu select Set/Clear Start Point.
Run Any Single Command Double-click any single command to run it by itself. This is useful when writing a single command. It lets you immediately test a command you are constructing, when you are not sure if it is correct. You can double-click it to see if it runs correctly. This is also available from the context menu.

Test Suite:

A test suite is a collection of tests. Often one will run all the tests in a test suite as one continuous batch-job.
When using Selenium-IDE, test suites also can be defined using a simple HTML file. The syntax again is simple. An HTML table defines a list of tests where each row defines the filesystem path to each test.

Installing the IDE

Using Firefox, first, download the IDE from the SeleniumHQ [downloads page](http://seleniumhq.org/download/)
Firefox will protect you from installing addons from unfamiliar locations, so you will need to click 'Allow' to proceed with the installation, as shown in the following screenshot.



Firefox prevented this site (seleniumhq.org) from asking you to install software on your computer.

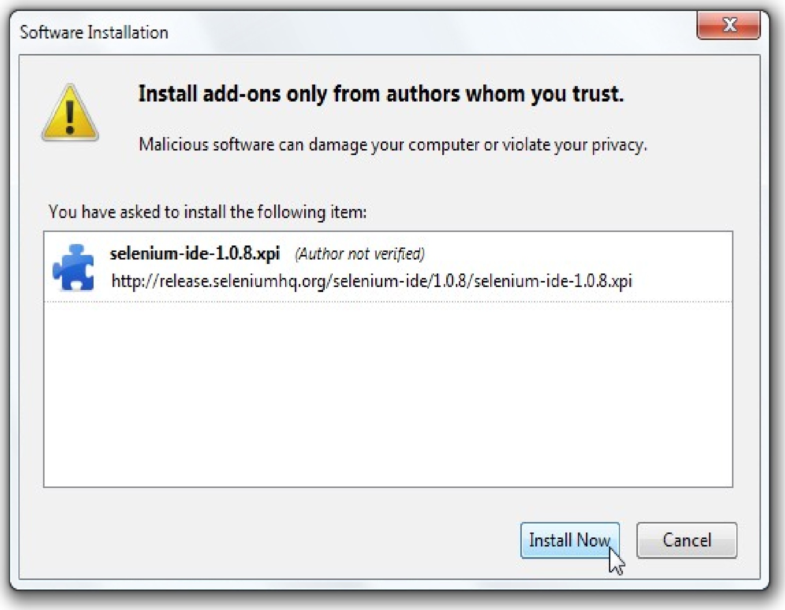
Downloads

Below is where you can find the latest releases of all the Selenium projects. You can also find a list previous releases, nightly releases, source code, and additional information for Maven users (Maven is a popular Java build tool).

Project	Release Date	Version	
Selenium Core	June 10, 2009	1.0.1	Download | Changelog
Selenium IDE	Nov 5, 2010	1.0.8	Download | Release Notes
Selenium RC	February 23, 2010	1.0.3	Download
Selenium Grid	April 8, 2010	1.0.6	Download (zip) | Changelog
CubicTest	Nov 10, 2008	1.8.11	Download | Changelog
Bromine	July 25, 2010	3.0 RC 2	Download
Selenium 2	April 22, 2010	2.0 alpha 5	Download

Plugins for Selenium IDE

When downloading from Firefox, you'll be presented with the following window.


Software Installation

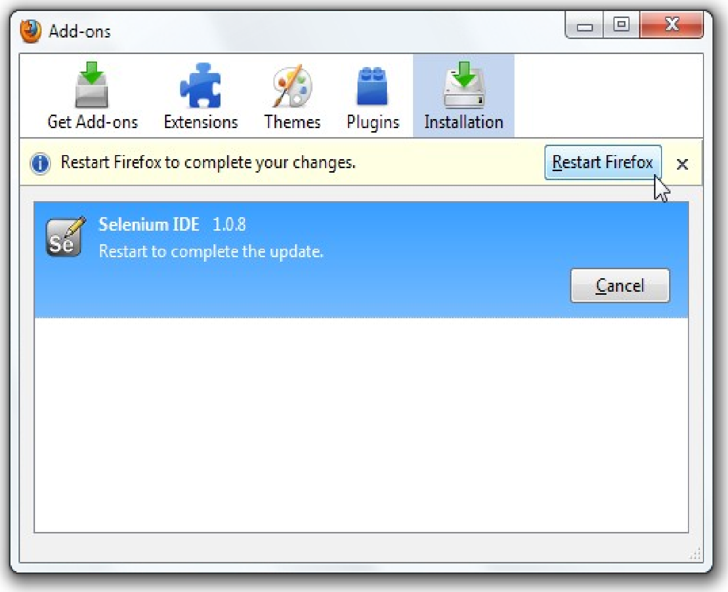
Install add-ons only from authors whom you trust.

Malicious software can damage your computer or violate your privacy.

You have asked to install the following item:

selenium-ide-1.0.8.xpi *(Author not verified)*
http://release.seleniumhq.org/selenium-ide/1.0.8/selenium-ide-1.0.8.xpi

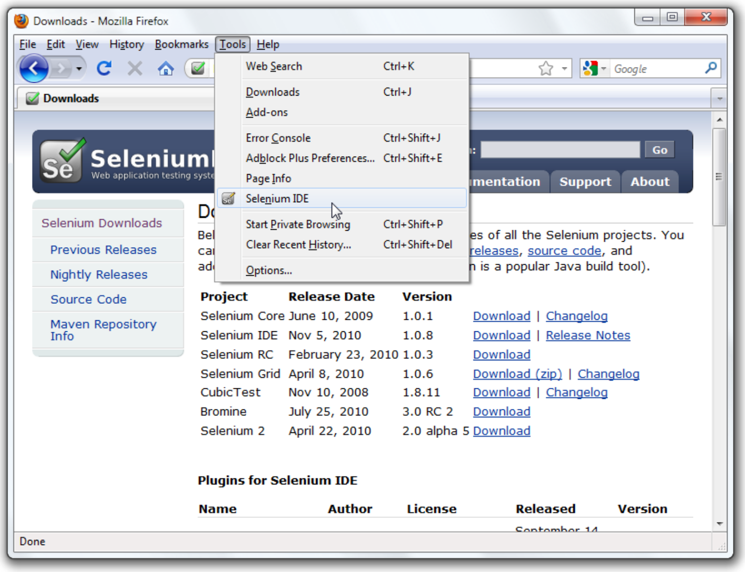
Select Install Now. The Firefox Add-ons window pops up, first showing a progress bar, and when the download is complete, displays the following.


Add-ons

Restart Firefox to complete your changes.

Selenium IDE 1.0.8
Restart to complete the update.

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


Downloads - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Web Search Ctrl+K
Downloads Ctrl+J
Add-ons
Error Console Ctrl+Shift+J
Adblock Plus Preferences... Ctrl+Shift+E
Page Info
Selenium IDE
Start Private Browsing Ctrl+Shift+P
Clear Recent History... Ctrl+Shift+Del
Options...
Google
Downloads
Selenium
Web application testing syste
Go
mentation
Support
About
Selenium Downloads
Previous Releases
Nightly Releases
Source Code
Maven Repository Info
Done

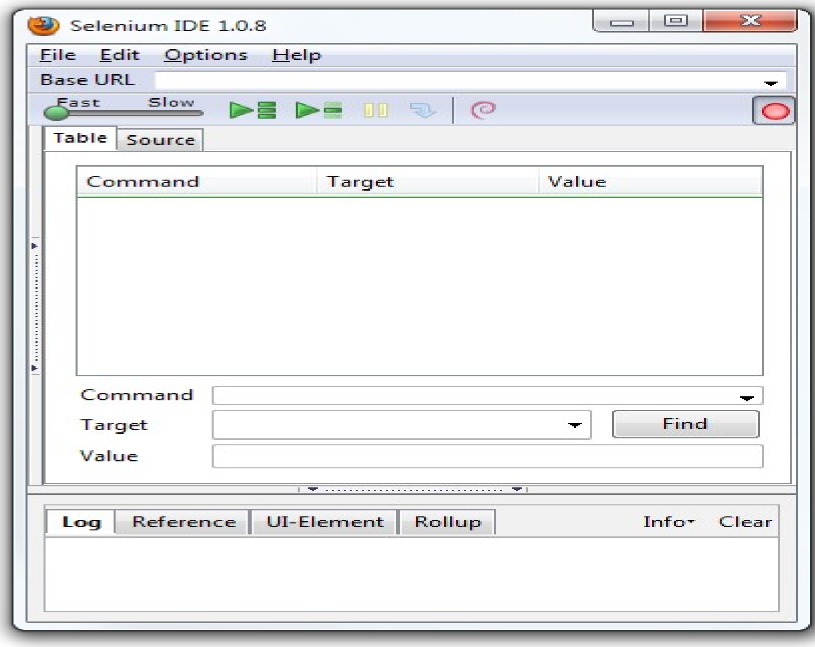
Project	Release Date	Version	
Selenium Core	June 10, 2009	1.0.1	Download | Changelog
Selenium IDE	Nov 5, 2010	1.0.8	Download | Release Notes
Selenium RC	February 23, 2010	1.0.3	Download
Selenium Grid	April 8, 2010	1.0.6	Download (zip) | Changelog
CubicTest	Nov 10, 2008	1.8.11	Download | Changelog
Bromine	July 25, 2010	3.0 RC 2	Download
Selenium 2	April 22, 2010	2.0 alpha 5	Download

Plugins for Selenium IDE

Name	Author	License	Released	Version

Opening the IDE

To run the Selenium-IDE, simply select it from the Firefox Tools menu. It opens as follows with an empty script-editing window and a menu for loading, or creating new test cases.



EXPT NO:-2 Using Selenium IDE, Write a test suite containing minimum 4 test cases.

TC'S #1: Manual Steps:

- Open (Example : Type www.google.com)
- Type "energy efficient" in the Google Search Input Box
- Click outside on an empty spot
- Click Search Button
- Verify the Text Present as "energy efficient"
- Assert the Title as "energy efficient - Google Search"
- Save the test case with .HTML Extension.

TC'S #2:

- Open (Example : Type www.google.com)
- Type "Selenium RC" in the Google Search Input Box
- Click outside on an empty spot
- Click Search Button
- Verify the Text Present as "Selenium RC"
- Assert the Title as "Selenium RC - Google Search"
- Save the test case with .HTML Extension.

Steps for creating test suite:

1. Create more Tc's save each Test Case with <.html> extension.

2. Open Firefox

3. Open Tools → Selenium IDE

4. File → Open → new Test Suite

5. File → Open → Add Test cases

6. Add more test cases

7. Save Suite with <.Html> extensions.

8. Run the test suite

EXPT NO:-3 Conduct a test suite for nay two web sites.

TC'S #1: Manual Steps:

- Open (Example : Type www.google.com)
- Type "energy efficient" in the Google Search Input Box
- Click outside on an empty spot
- Click Search Button
- Verify the Text Present as "energy efficient"
- Assert the Title as "energy efficient - Google Search"
- Save the test case with .HTML Extension.

TC#2:

1: Open Firefox Web Browser

2: In the address bar, Type http://www.yahoo.com

3: In the search input button, Type "energy efficient"

4: Click on the "Web Search" submit button

5: Wait for Search Results to come on "http:/search.yahoo.com"

6: Verify "energy efficient" text is present anywhere in the search results: (Select and

highlight anywhere in the search results page, "energy efficient" text is present.)

7: Verify the browsers title has the value "energy efficient - Yahoo! Search Results"

8. Save the test case with .HTML Extension.

Steps for creating test suite:

1. Create more Tc's save each Test Case with <.html> extension.

2. Open Firefox

3. Open Tools → Selenium IDE

4. File → Open → new Test Suite

5. File → Open → Add Test cases

6. Add more test cases

7. Save Suite with <.Html> extensions.

8. Run the test suite

Selenium-RC

1. Introduction

Selenium-RC is the solution for tests that need more than simple browser actions and linear execution.
Selenium-RC uses the full power of programming languages to create more complex tests like reading and writing files, querying a database, emailing test results.
You'll want to use Selenium-RC whenever your test requires logic not supported by Selenium-IDE.
What logic could this be? For example, Selenium-IDE does not directly support:

- condition statements
- iteration
- logging and reporting of test results
- error handling, particularly unexpected errors
- database testing
- test case grouping
- re-execution of failed tests
- test case dependency
- screenshot capture of test failures

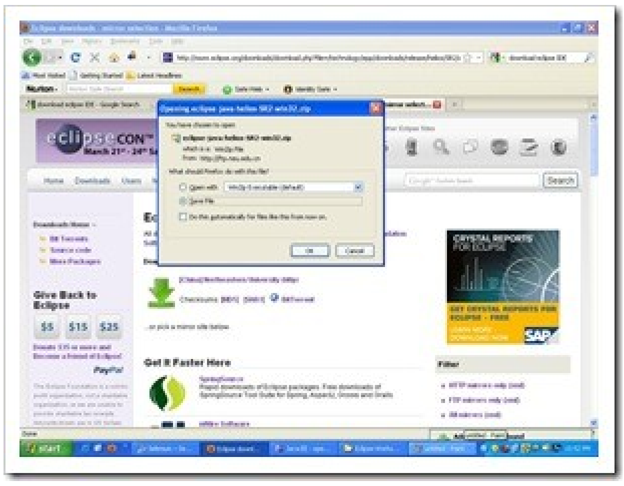
Although these tasks are not supported by Selenium directly, all of them can be achieved by using programming techniques with a language-specific Selenium-RC client library.

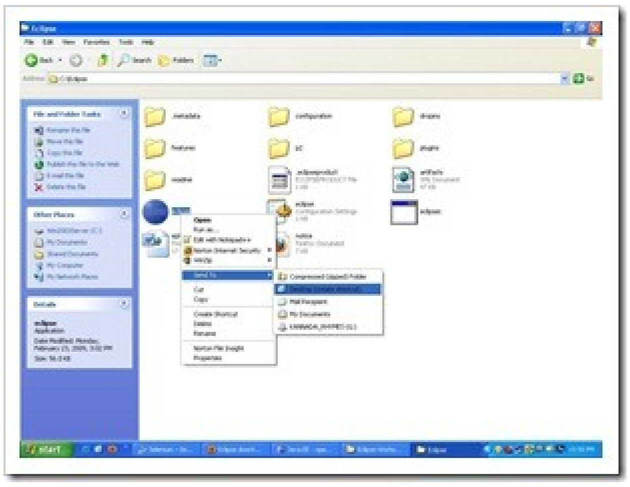
EXPT NO :-4 Install Selenium server and demonstrate it using a script in Java/PHP

Installation of Selenium RC and Eclipse

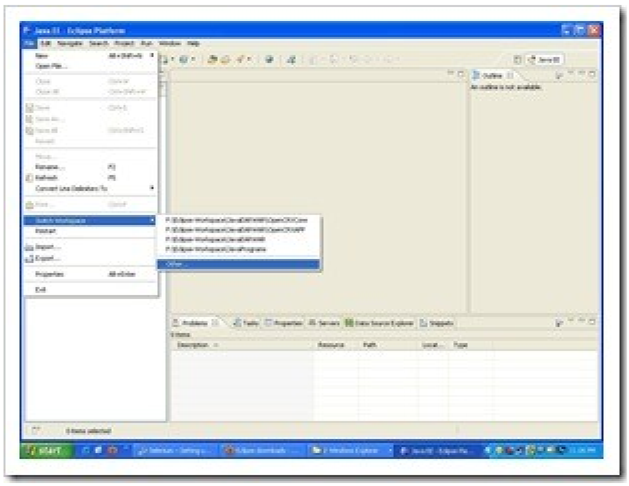
Download Eclipse

1. Go to URL – http://www.eclipse.org/downloads/
2. Select Eclipse IDE for Java Developers (Click on Windows 32 bit platform)
3. Click on OK button and save to a local drive (i.e. C: or D:, etc)
4. Unzip the downloaded zip file and rename that to **Eclipse**
5. Create one more folder “Eclipse-Workspace” (i.e. C:Eclipse-Workspace)in the same drive where Eclipse is unzipped and renamed.
6. Create Eclipse desktop shortcut (go to C:Eclipse folder –> right click Eclipse.exe and then click on “desktop create shortcut”) as demonstrated in the below pictures.





1. Now we need to create a workspace folder –> C:Eclipse-WorkspaceSeleniumTests
2. Double click on "Eclipse shortcut on Desktop"
3. This opens the Eclipse
4. Close Eclipse welcome screen
5. Click File menu –> Switch Worspace –> other
6. Now Select the C:Eclipse-WorkspaceSeleniumTests folder (These steps are demonstrated in the following figure)

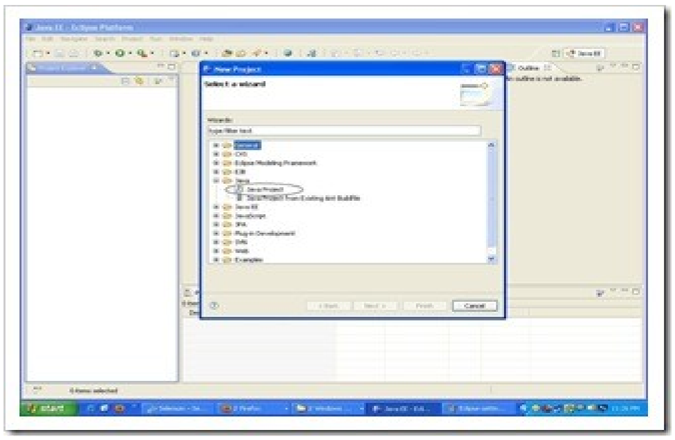


We have finished setting up the eclipse. Now, we need to download Selenium RC server / client driver and configure that to Eclipse

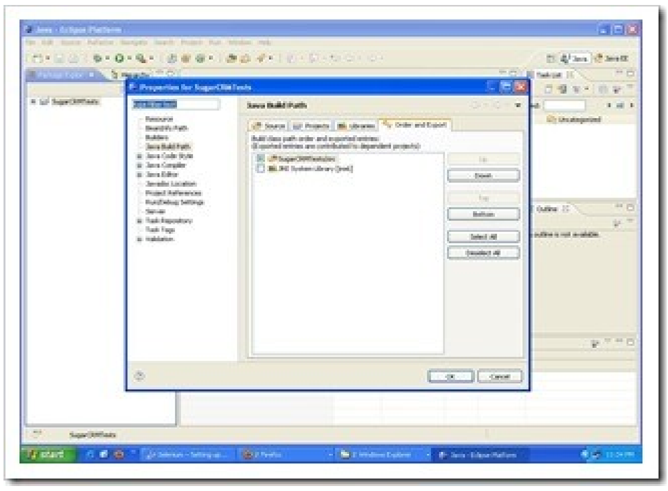
1. Download Selenium server: http://seleniumhq.org/download/
2. Download Selenium Client driver for Java (from Selenium Client Drivers section)
3. Create "Selenium" folder in C: drive and copy the Selenium-server.jar as well as unzip the Selenium Client driver (C:Selenium)

Downloading and unzipping the files into a folder is done. We need to configure the appropriate Selenium Client driver Jar file to the Eclipse.

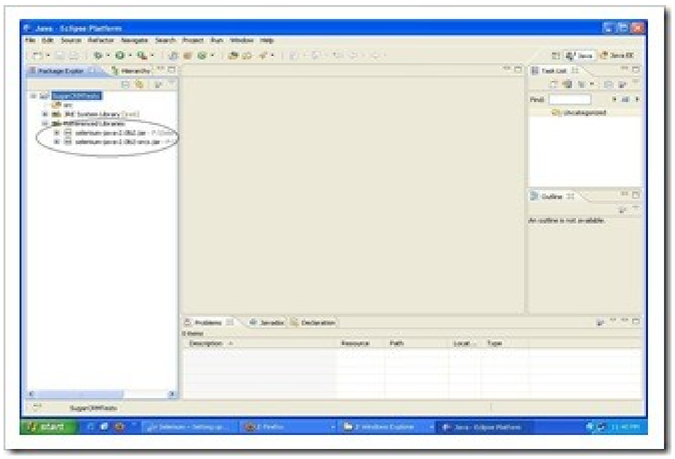
1. Go to Eclipse –> Click File –> New –> Project (from various options need to select just "project")
2. In Select Wizard –> Click Java –> "Java Project" (demonstrated in the below figure)



3. Give the project name (e.g. SugarCRMTests)
4. Click Finish – Click Yes
5. Now we are done with creation of project and need to configure the Selenium Client driver to this Project
6. Right Click “SugarCRMTests” project



7. Click “Java Build Path”
8. Click Libraries tab
9. Click “Add External JARs” button
10. Select “Selenium Client Drivers” unzipped in C:Selenium folder (Selenium Server JAR file should not be added)
11. Click OK
12. Referenced libraries –> contains both the Selenium Client driver jar files as shown in the below picture.



First script-Selenium RC-JUnit-Java

Selenium RC allows to write the automated tests in any language. Selenium Scripts can be written in various ways:

1. Extending the Selenese Test Case
2. Extending the TestCase-Junit and defining an object to the DefaultSelenium Class

Before you start with your script, the following are the steps that should be performed.
Method1: Extending the Selenese Test Case

1. Install any IDE (Eclipse, IntelliJ, IDEA), create a project.
2. Download the selenium RC file from Selenium website.
3. Extract the selenium-remote-control-1.0.3.zip file and add the selenium-remote-control-1.0.3\selenium-java-client-driver-1.0.1\selenium-java-clientdriver.jar and selenium-remote-control-1.0.3\selenium-server-1.0.3\selenium-server.jar file to the build path.
4. Create a new package and a class in the newly created package.
5. Copy the below code:

```
import com.thoughtworks.selenium.SeleneseTestCase;
public class SeleniumTest1 extends SeleneseTestCase {
public void setUp() throws Exception {
setUp("http://www.hexbytes.com/", "*firefox");
}
public void testNew() throws Exception{
selenium.open("/");
selenium.type("searchbox", "selenium rc");
selenium.click("css=input[type='submit'][value='Search']");
selenium.waitForPageToLoad("30000");
assertTrue(selenium.isTextPresent("selenium rc"));
}
}
```

6.Start the selenium Server.

7.Click on the RunAs button and run the script as JUnit Test. If you don't find the option Install the JUnit plug-in and run the script.

Method2: Extending the DefaultSelenium Class.

Some times we need the selenium scripts to contact the server running on a different port and different machine. You can configure the selenium scripts to handle such situations easily by extending your selenium script from TestCase(Junit) and defining the selenium instance as an object to DefaultSelenium class.

```
package test;
import junit.framework.*;
import com.thoughtworks.selenium.DefaultSelenium;
public class DefaultSelenium1 extends TestCase{
private DefaultSelenium selenium;
public void testing(){
   selenium=new
DefaultSelenium("localhost",4444,"*iexplore","http://hexbytes.com");
   selenium.start();
   selenium.open("/");
   selenium.type("searchbox", "selenium rc");
   selenium.click("css=input[type='submit'][value='Search']");
   selenium.waitForPageToLoad("30000");
   Assert.assertTrue(selenium.isTextPresent("selenium rc"));
   }
}
```

EXPT NO:-5 Write and test a program to login a specific web page.

```
import com.thoughtworks.selenium.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import java.util.regex.Pattern;

public class exp5 extends SeleneseTestCase {
        @Before
        public void setUp() throws Exception {
                selenium = new DefaultSelenium("localhost", 4444, "*chrome",
"http://demo.opensourcecms.com/");
                selenium.start();
        }

        @Test
        public void testExp5() throws Exception {
                selenium.open("/wordpress/wp-login.php");
                selenium.type("id=user_login", "admin");
                selenium.type("id=user_pass", "demo123");
                selenium.click("id=wp-submit");
                selenium.waitForPageToLoad("30000");
        }

        @After
        public void tearDown() throws Exception {
                selenium.stop();
        }
}
```

TestNG

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use, such as:

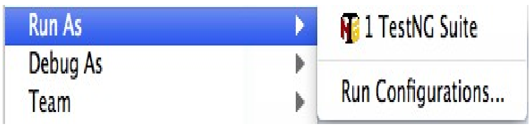
- Annotations.
- Run your tests in arbitrarily big thread pools with various policies available (all methods in their own thread, one thread per test class, etc...).
- Test that your code is multithread safe.
- Flexible test configuration.
- Support for data-driven testing (with @DataProvider).
- Support for parameters.
- Powerful execution model (no more TestSuite).
- Supported by a variety of tools and plug-ins (Eclipse, IDEA, Maven, etc...).
- Embeds BeanShell for further flexibility.
- Default JDK functions for runtime and logging (no dependencies).
- Dependent methods for application server testing.

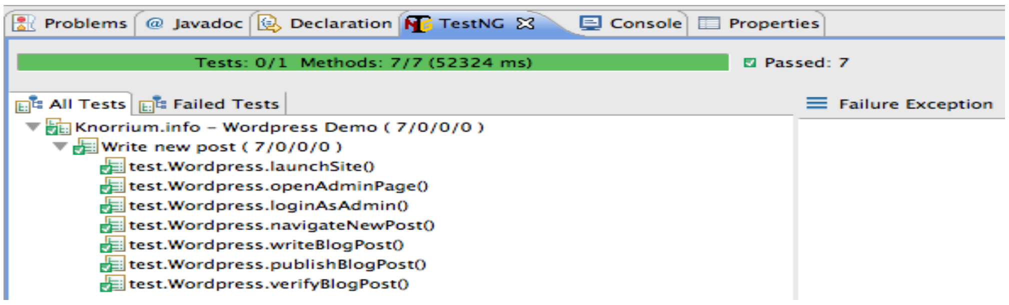
TestNG is designed to cover all categories of tests: unit, functional, end-to-end, integration, etc...

Installing TestNG in eclipse

- Select *Help / Software updates / Find and Install.*
- *Search for new features to install.*
- *New remote site.*
- For Eclipse 3.4 and above, enter http://beust.com/eclipse.
- For Eclipse 3.3 and below, enter http://beust.com/eclipse1.
- Make sure the check box next to URL is checked and click *Next.*
- Eclipse will then guide you through the process.

Launching your tests in Eclipse

- We finished writing our tests, now how can we run them?
- You can launch TestNG from the command line, using a Eclipse plugin or even programatically. We are going to use the Eclipse plugin. Follow the steps described on the official TestNG documentation over here
- If you installed TestNG correctly, you will see this menu when you right click on the XML file:
- 

- Click on "Run as TestNG Suite" and your test will start running. You will then see this nice results tree:
- 

```
Tests: 0/1  Methods: 7/7 (52324 ms)     Passed: 7
Knorrium.info - Wordpress Demo ( 7/0/0/0 )
  Write new post ( 7/0/0/0 )
    test.Wordpress.launchSite()
    test.Wordpress.openAdminPage()
    test.Wordpress.loginAsAdmin()
    test.Wordpress.navigateNewPost()
    test.Wordpress.writeBlogPost()
    test.Wordpress.publishBlogPost()
    test.Wordpress.verifyBlogPost()
```

Selenium Tests with Microsoft Excel

Parameterizing a test from external sources such as Microsoft Excel is always recommended in order to handle large amount of test data. To read data from Excel, we need APIs which support opening file, reading data, and writing data into Excel. We should know various classes and methods which support above mentioned operations. In this post, let us try to figure out which is the API that supports all the activities we need to do during execution of a test.

Jxl.jar is an open source Java API which supports read Excel spreadsheets and to write into Excel spreadsheets. Below are some of the operations that we can handle with this API.

1. Read data from Excel spreadsheet
2. Read and write formulas into spreadsheets
3. Generate spreadsheets

4. Supports formatting of font, number, and date
5. Supports coloring of cells

To access the methods and classes provided by this API inside Eclipse we need to add this JAR file to the Java Build Path. (I have explained steps to add external Jar files to Java Build Path in previous posts)

Download the jxl.jar from "***http://jexcelapi.sourceforge.net/***"

Add the JAR file to Java Build Path

Add **import** statements to the .java file as below to read from an Excel spreadsheet

import jxl.Cell;

import jxl.Sheet;

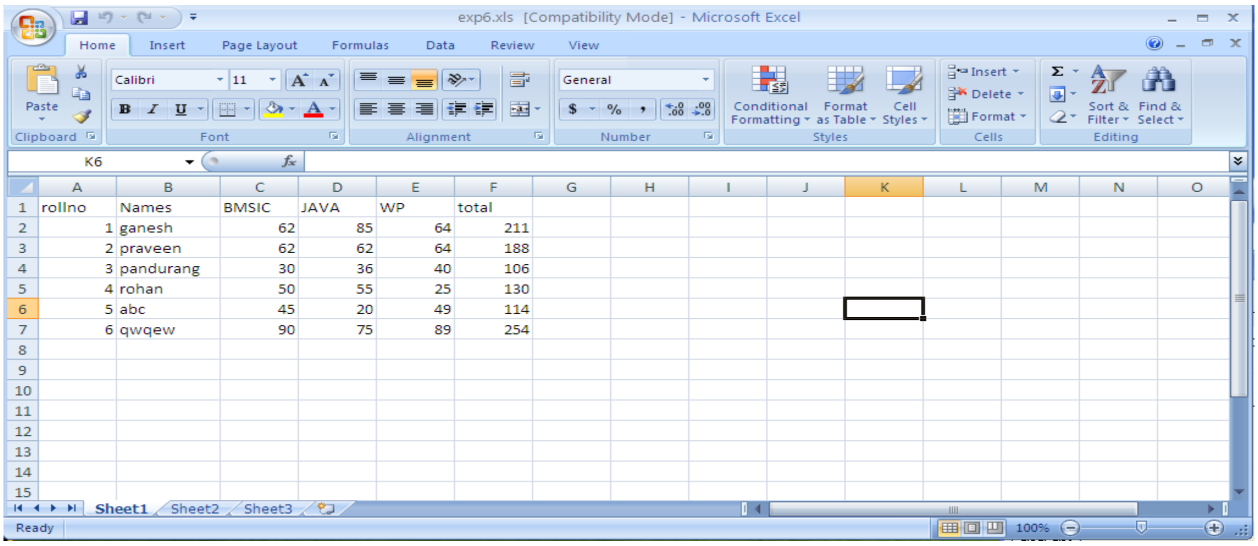
import jxl.Workbook;

import jxl.read.biff.BiffException;

EXPT NO :-6 Write and test a program to update 10 student records into table into Excel file .

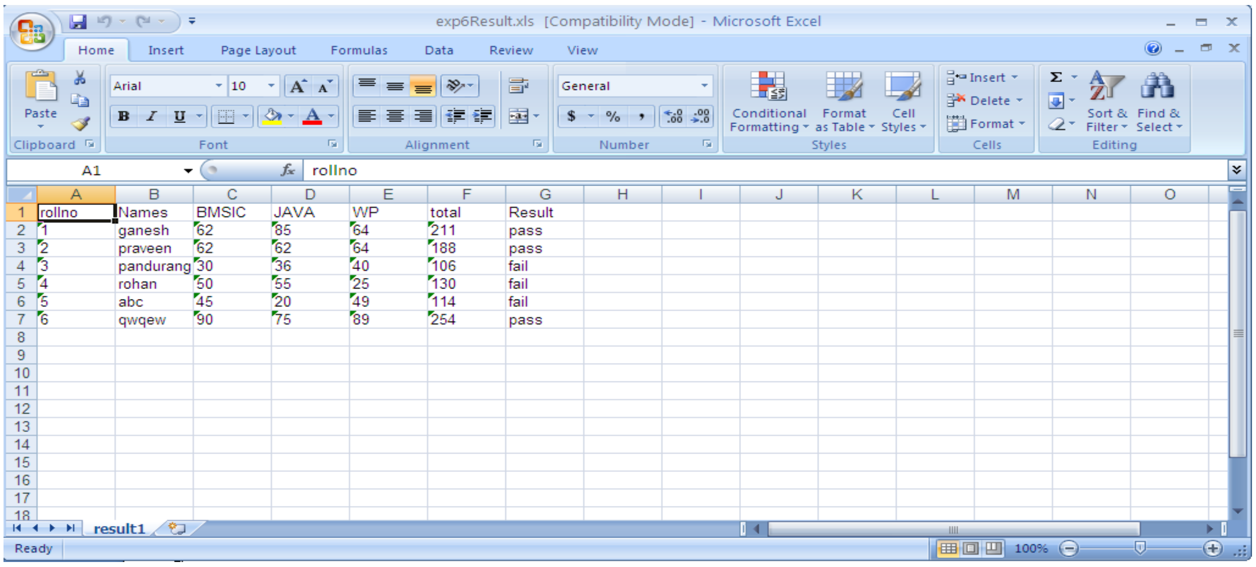
```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import org.testng.annotations.*;
public class newj {

@BeforeClass
public void setUp() throws Exception {}
@Test
public void testImportexport1() throws Exception {
FileInputStream fi = new FileInputStream("D:\\exp6.xls");
Workbook w = Workbook.getWorkbook(fi);
Sheet s = w.getSheet(0);
String a[][] = new String[s.getRows()][s.getColumns()];
FileOutputStream fo = new FileOutputStream("D://exp6Result.xls");
WritableWorkbook wwb = Workbook.createWorkbook(fo);
WritableSheet ws = wwb.createSheet("result1", 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);
        }
        else
        {
                Label l1 = new Label(6, i, "fail");
                ws.addCell(l1);
                break;
        }
                }
}
wwb.write();
wwb.close();
}
}
```

INPUT



	A	B	C	D	E	F
1	rollno	Names	BMSIC	JAVA	WP	total
2	1	ganesh	62	85	64	211
3	2	praveen	62	62	64	188
4	3	pandurang	30	36	40	106
5	4	rohan	50	55	25	130
6	5	abc	45	20	49	114
7	6	qwqew	90	75	89	254

OUTPUT

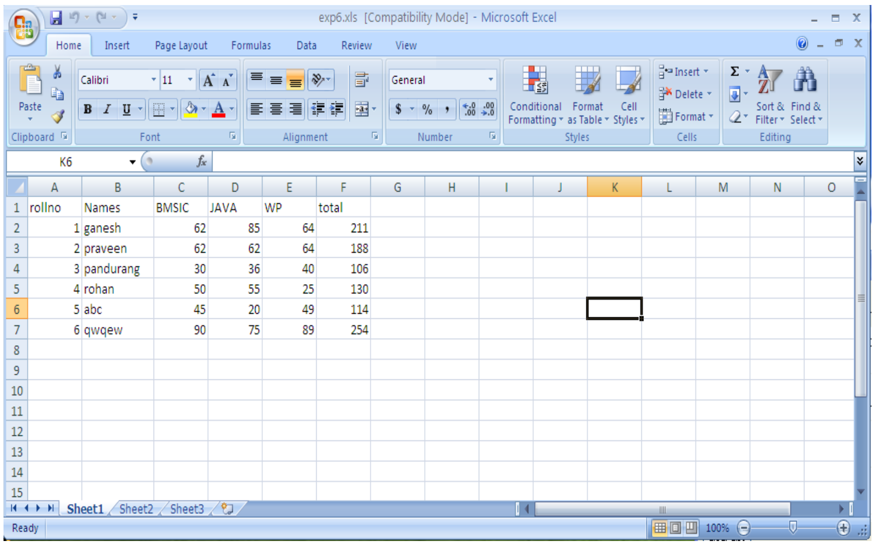


	A	B	C	D	E	F	G
1	rollno	Names	BMSIC	JAVA	WP	total	Result
2	1	ganesh	62	85	64	211	pass
3	2	praveen	62	62	64	188	pass
4	3	pandurang	30	36	40	106	fail
5	4	rohan	50	55	25	130	fail
6	5	abc	45	20	49	114	fail
7	6	qwqew	90	75	89	254	pass

EXPT NO :-7 Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).

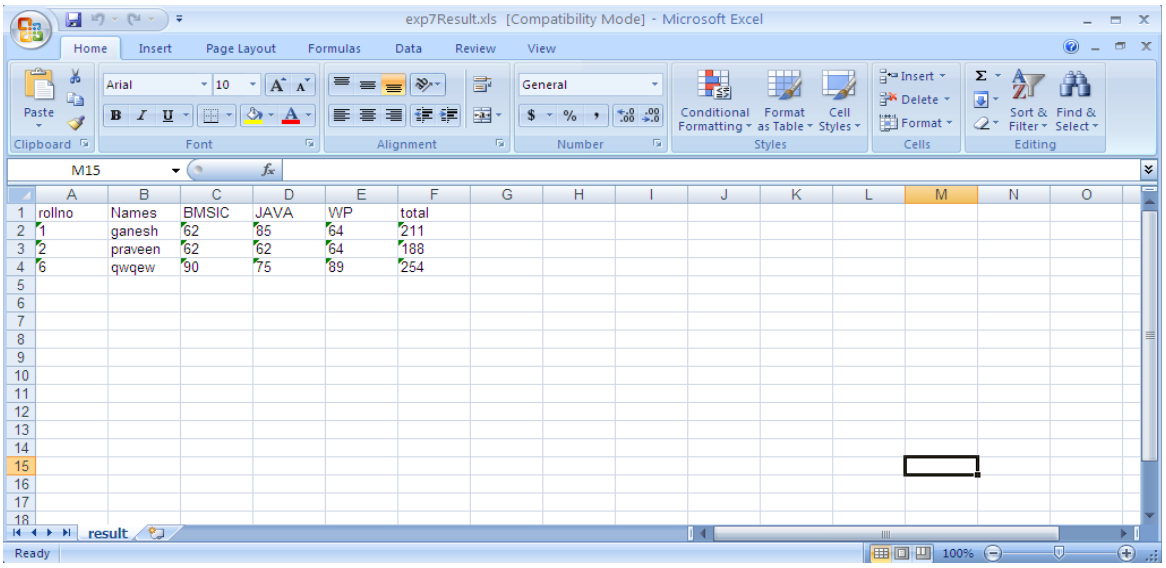
```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import org.testng.annotations.*;
public class exp7 {

@BeforeClass
public void setUp() throws Exception {
}
@Test
public void testImportexport1() throws Exception {
FileInputStream fi = new FileInputStream("D:\\exp6.xls");
Workbook w = Workbook.getWorkbook(fi);
Sheet s = w.getSheet(0);
String a[][] = new String[s.getRows()][s.getColumns()];
FileOutputStream fo = new FileOutputStream("D://exp7Result.xls");
WritableWorkbook wwb = Workbook.createWorkbook(fo);
WritableSheet ws = wwb.createSheet("result", 0);
int c=0;
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();
}
}
```

INPUT



rollno	Names	BMSIC	JAVA	WP	total
1	ganesh	62	85	64	211
2	praveen	62	62	64	188
3	pandurang	30	36	40	106
4	rohan	50	55	25	130
5	abc	45	20	49	114
6	qwqew	90	75	89	254

OUTPUT



exp7Result.xls [Compatibility Mode] - Microsoft Excel

M15

rollno	Names	BMSIC	JAVA	WP	total
1	ganesh	62	85	64	211
2	praveen	62	62	64	188
6	qwqew	90	75	89	254

EXPT NO :-8 Write and test a program to provide total number of objects present / available on the page .

```
package testscripts;
import com.thoughtworks.selenium.*;
import org.openqa.selenium.server.*;
import org.testng.annotations.*;

public class exp8 {
public Selenium selenium;
public SeleniumServer seleniumserver;

@BeforeClass
public void setUp() throws Exception {
RemoteControlConfiguration rc = new RemoteControlConfiguration();
seleniumserver = new SeleniumServer(rc);
selenium = new DefaultSelenium("localhost", 4444, "*firefox","http://");
seleniumserver.start();
selenium.start();
}

@Test
public void testDefaultTNG() throws Exception {

selenium.open("http://www.google.co.in/");
selenium.windowMaximize();
String lc[]=selenium.getAllLinks();
System.out.println("TOTAL NO OF LINKS="+lc.length);

String bc[]=selenium.getAllButtons();
System.out.println("TOTAL NO OF BUTTONS="+bc.length);
String fc[]=selenium.getAllFields();
System.out.println("TOTAL NO OF INPUT FIELDS="+fc.length);
}
@AfterClass
public void tearDown() throws Exception {
selenium.stop();
seleniumserver.stop();

}
}
```

OUTPUT

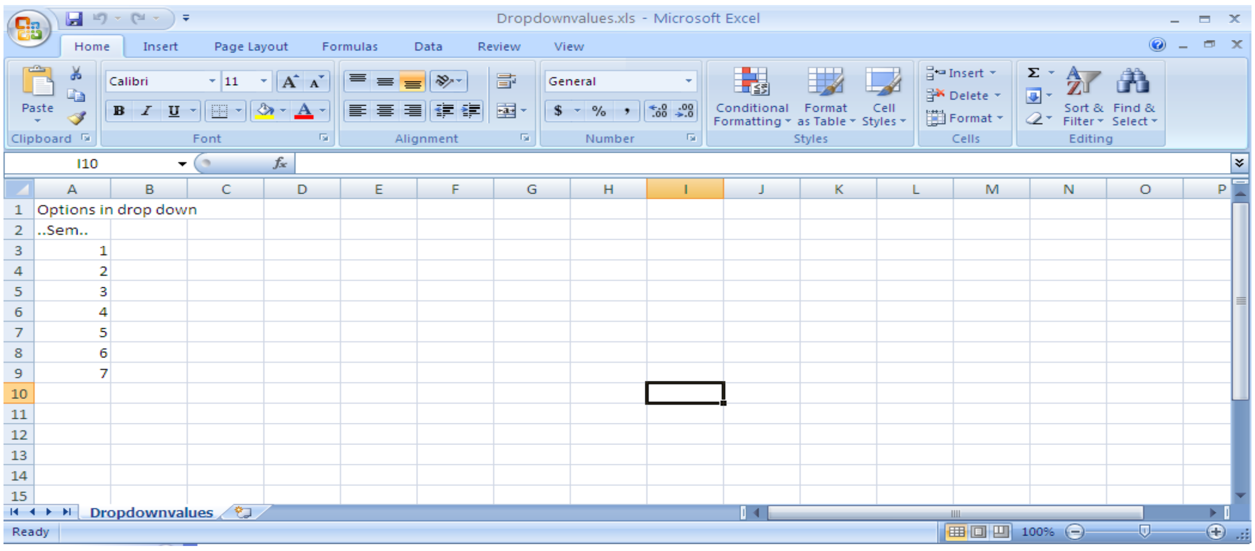
```
TOTAL NO OF LINKS=41
TOTAL NO OF BUTTONS=1
TOTAL NO OF INPUT FIELDS=3
```

EXPT NO :-9 Write and test a program to get the number of list items in a list / combo box.

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import com.thoughtworks.selenium.*;
import org.openqa.selenium.server.*;
import org.testng.annotations.*;
public class Dropdowngoogle {
public Selenium selenium;
public SeleniumServer seleniumserver;

  @BeforeClass
  public void setUp() throws Exception {
RemoteControlConfiguration rc = new RemoteControlConfiguration();
seleniumserver = new SeleniumServer(rc);
selenium = new DefaultSelenium("localhost", 4444, "*firefox", "http://");
seleniumserver.start();
selenium.start();
}
@Test
public void testDropdowngoogle()throws Exception {
selenium.open("http://www.schools9.com/kdiploma11022011.htm");
selenium.waitForPageToLoad("3000");
   String[] option = selenium.getSelectOptions("name=course");
   File file = new File("C:/Dropdownvalues.xls");//excel
   File file1 = new File("C:/Dropdownvalues.txt");//notepad
   BufferedWriter out = new BufferedWriter(new FileWriter(file));//excel
   BufferedWriter out1 = new BufferedWriter(new FileWriter(file1));//notepad
   out.write("Options in drop down\n");//Excel
   out1.write("Options in drop down\n");//notepad
   for (int i = 0; i < option.length; i++) {
      System.out.println("Option: " + i + " is" + option[i]);
      out.write(""+option[i]);
```

```
        out.newLine();
        out1.write(""+option[i]);
        out1.newLine();
    }
    out.close();//excel
    out1.close();//notepad
  }
 @AfterClass
 public void tearDown()throws Exception {
selenium.stop();
seleniumserver.stop();
 }
 }
```

OUTPUT



Dropdownvalues.xls - Microsoft Excel

```
   A
1  Options in drop down
2  ..Sem..
3        1
4        2
5        3
6        4
7        5
8        6
9        7
```

EXPT NO :-10 Write and test a program to count number of check boxes on the page checked and unchecked count.

```
package testscripts;
import com.thoughtworks.selenium.*;
import org.openqa.selenium.server.*;
import org.testng.annotations.*;

public class exp8 {
public Selenium selenium;
public SeleniumServer seleniumserver;

@BeforeClass
public void setUp() throws Exception {
RemoteControlConfiguration rc = new RemoteControlConfiguration();
seleniumserver = new SeleniumServer(rc);
selenium = new DefaultSelenium("localhost", 4444, "*firefox","http://");
seleniumserver.start();
selenium.start();
}
@Test
public void testDefaultTNG() throws Exception {
selenium.open("http://browsershots.org/");
selenium.windowMaximize();
//Count of check boxes
Number c  =selenium.getXpathCount("//input[@type['checkbox']]");
System.out.println("Count of check boxes " +c);
//Number of check boxes checked
Number  d = selenium.getXpathCount("//input[@type='checkbox' and
@checked]");
System.out.println("Count of Checked check boxes " +d);
//Number of check boxes unchecked
Number e = selenium.getXpathCount("//input[@type='checkbox' and
not(@checked)]");
System.out.println("Count of Checked check boxes " +e);
}
@AfterClass
public void tearDown() throws Exception {
selenium.stop();
seleniumserver.stop();

}
}
```

OUTPUT

```
Count of check boxes 127
Count of Checked check boxes 91
Count of Checked check boxes 36
```