

# **SOFTWARE TESTING LAB MANNUAL**

**FOR 6<sup>th</sup> SEM IS AND CS**

**(2011-2012)**

**BY  
MISS. SAVITHA R  
LECTURER  
INFORMATION SCIENCE DEPARTMENT  
GOVT. POLYTECHNIC  
GULBARGA**

FOR ANY FEEDBACK CONTACT TO  
EMAIL: [savitharamu@gmail.com](mailto:savitharamu@gmail.com)

## 1. Understand the Automation Testing Approach (Theory Concept)

### Introduction:

#### What is Automation?

- Automation is making a process automatic, eliminating the need for human intervention.
- Is a self-controlling or self-moving process.
  - Automation software offers automation wizard and commands of its own in addition to providing task recording and re-play capabilities.
  - Using these tools we can record the task
  - Then if needed use the editor to edit the task, add new commands or using GUI automation wizards

#### Benefits of Automation

**Fast:** test automation runs faster than human users

**Reliable:** it performs precisely the same operations at each time elimination human errors

**Repeatable:** It performs the same operation with different combination of data in a less time

**Programmable:** We can use sophisticated tests that brings out hidden information

**Reusable:** We can develop reusable components which can run on different versions of application under test

**Regression testing:** easy to conduct regression test

**Enabling 24\*7 testing:** test can be schedule and supports unattended recovery

**Robust Verification:** Support robust verification mechanism than any other testing tools

#### What Tests should be Automated?

- Tests executed for each software build
- tests which can be easily automated
- Business critical tests
- tests that are difficult to perform manually

#### Test artefacts (tools):

**Requirements:** the requirements that are being tested are stored into the repository. This feature enables us to map the requirements to test cases and test scripts.

**Test Plan:** Design the tests and build the test plan on it using the test management tool. The test plan will be maintained in a central repository. This will help to produce reusable tests cases in future. The information is sharable and preserved.

**Test case:** Contains description of test case and test steps.

**Test script:** Automated test script associated with each test case will be stored in the quality centre repository.

**Test sets:** this will be a set of selected test cases and associated test scripts. The tests sets are executed from the quality centre onto remote machines. The status of each execution is stored in the respective test case execution status. The status can be either pass/fail, it doesn't maintain exact details of execution.

**Test logs:** The test script execution logs are stored in global repository. It contains the status of each test case.

**Reusable components:** The repetitive functions that are invoked in most of the functions are stored, which reduces the coding redundancies.

**Object repository:** contains the declaration of the application objects.

Data sheets: contains the test data used for execution. it can be populated by the database or manually.

#### **Automation Activity:**

Record the application GUI

Write Script

Execute script

Review the script

Modify script if any changes.

Execute scripts on at least two machines

Store the script in central repository

Execute the script from central repository

Verify the execution status in central repository

Verify the test logs on common server

Prepare the test reports

## 2. Using Selenium IDE, Write a test suite containing minimum 4 test cases

- Download Selenium IDE 1.5 or above from <http://seleniumhq.org/download/> from firefox browser.
- For more information refer the link [http://seleniumhq.org/docs/book/Selenium\\_Documentation.pdf](http://seleniumhq.org/docs/book/Selenium_Documentation.pdf)
- The webpage used in this example is a 3<sup>rd</sup> semester IS/CS Web Design part2-java Script program 'Menu Driven Arithmetic Operations'

Test Suite
<a href="#">add</a>
<a href="#">subtract</a>
<a href="#">multiply</a>
<a href="#">divide</a>

### Test case 1

add		
open	file:///C:/Users/Savitha/Desktop/java%20script/Arithmetic%20operation.html	
assertTitle	Arithmetic Operation	
type	name=n1	10
type	name=n2	20
click	css=input[type="button"]	
storeValue	name=result	addressresult
echo	' Addition Result =\${addressresult}'	

### Test case 2

subtract		
open	file:///C:/Users/Savitha/Desktop/java%20script/Arithmetic%20operation.html	
assertTitle	Arithmetic Operation	
type	name=n1	50
type	name=n2	40
click	//input[@value='Subtract']	
storeValue	name=result	subtractresult
echo	' Subtraction Result : \${subtractresult} '	

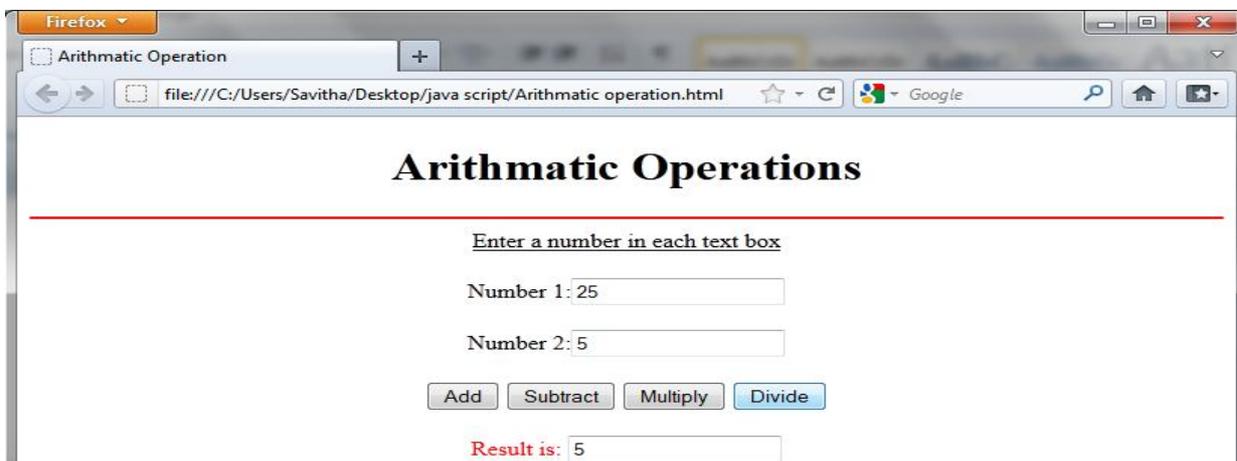
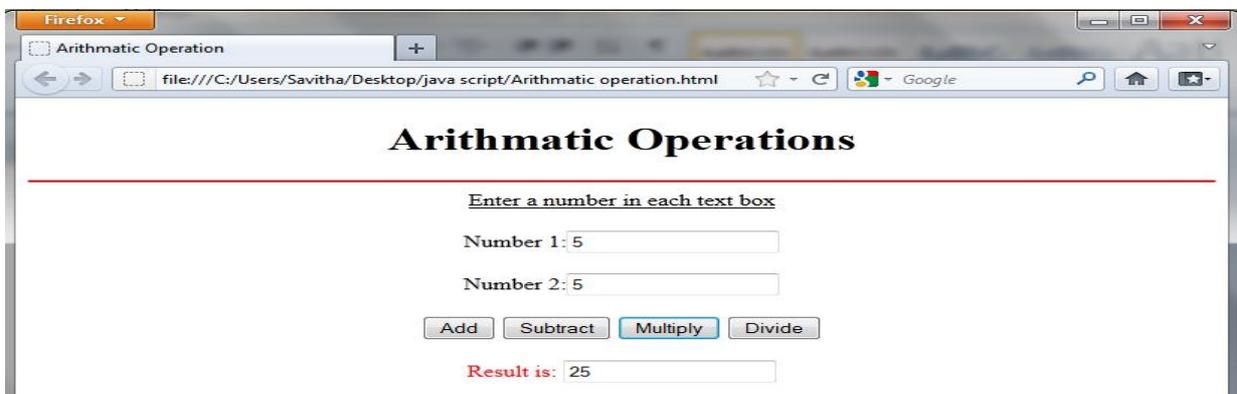
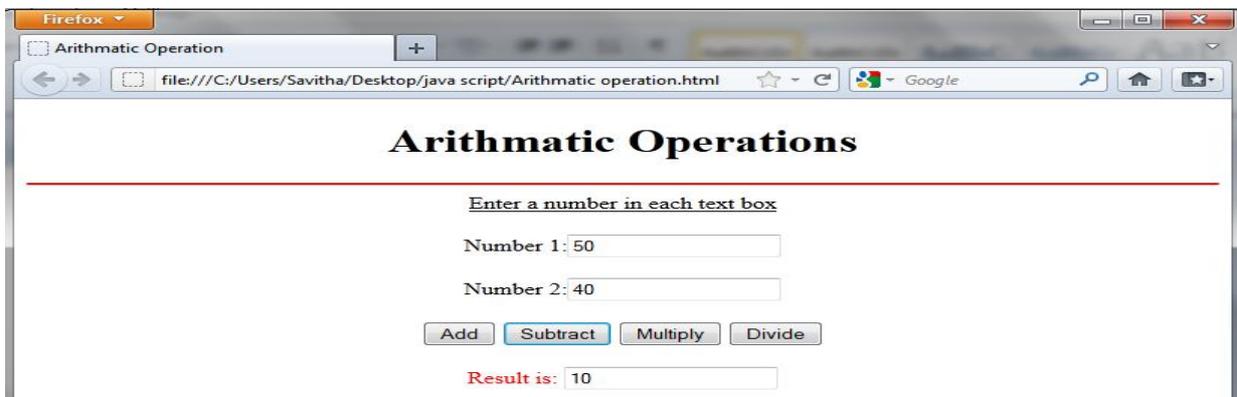
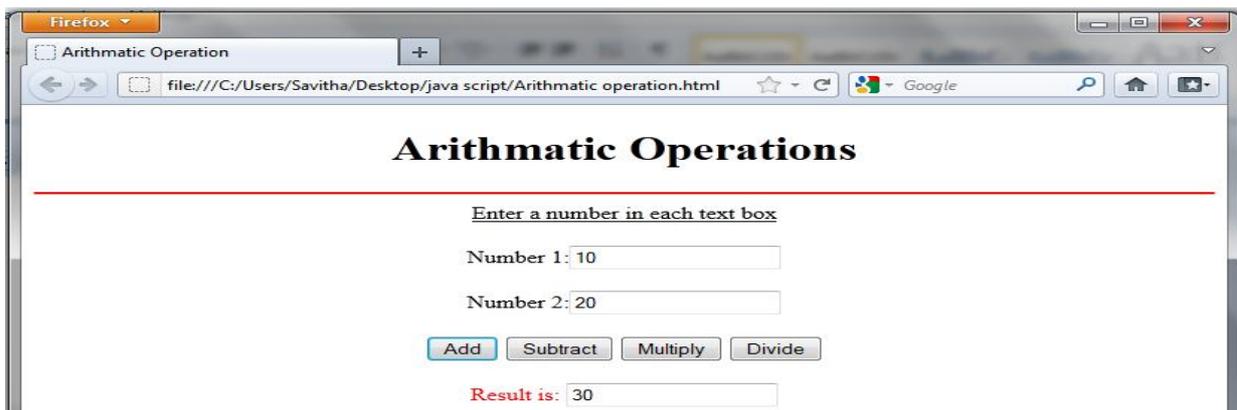
### Test case 3

multiply		
open	file:///C:/Users/Savitha/Desktop/java%20script/Arithmetic%20operation.html	
assertTitle	Arithmetic Operation	
type	name=n1	5
type	name=n2	5
click	//input[@value='Multiply']	
storeValue	name=result	multiplyresult
echo	'Multiplication Result : \${multiplyresult} '	

### Test case 4

divide		
open	file:///C:/Users/Savitha/Desktop/java%20script/Arithmetic%20operation.html	
assertTitle	Arithmetic Operation	
type	name=n1	25
type	name=n2	5
click	//input[@value='Divide']	
storeValue	name=result	divresult
echo	'Division Result :\${divresult}'	

OUTPUT:



The screenshot shows the Selenium IDE 1.6.0 interface. The main window displays a test case named 'divide' with the following commands in a table:

Command	Target	Value
open	file:///C:/Users/Savitha/...	
assertTitle	Arithmetic Operation	
type	name=n1	25
type	name=n2	5
click	//input[@value='Divide']	
storeValue	name=result	divresult
echo	'Division Result :\${divres...	

Below the table, there are input fields for 'Command', 'Target', and 'Value', along with a 'Find' button. The left sidebar shows a list of test cases: 'add', 'subtract', 'multiply', and 'divide', with 'divide' selected. The bottom status bar shows 'Runs: 4' and 'Failures: 0'.

The Log window at the bottom contains the following execution log:

```
[info] Executing: |open | file:///C:/Users/Savitha/Desktop/java%20script/Arithmetic%20operation.html | |
[info] Executing: |assertTitle | Arithmetic Operation | |
[info] Executing: |type | name=n1 | 10 |
[info] Executing: |type | name=n2 | 20 |
[info] Executing: |click | css=input[type="button"] | |
[info] Executing: |storeValue | name=result | addressult |
[info] Executing: |echo | ' Addition Result =${addressult}' | |
[info] echo: ' Addition Result =30'
[info] Changed test case
[info] Executing: |open | file:///C:/Users/Savitha/Desktop/java%20script/Arithmetic%20operation.html | |
[info] Executing: |assertTitle | Arithmetic Operation | |
[info] Executing: |type | name=n1 | 50 |
[info] Executing: |type | name=n2 | 40 |
[info] Executing: |click | //input[@value='Subtract'] | |
[info] Executing: |storeValue | name=result | subtractresult |
[info] Executing: |echo | ' Subtraction Result : ${subtractresult}' | |
[info] echo: ' Subtraction Result : 10 '
[info] Changed test case
[info] Executing: |open | file:///C:/Users/Savitha/Desktop/java%20script/Arithmetic%20operation.html | |
[info] Executing: |assertTitle | Arithmetic Operation | |
[info] Executing: |type | name=n1 | 5 |
[info] Executing: |type | name=n2 | 5 |
[info] Executing: |click | //input[@value='Multiply'] | |
[info] Executing: |storeValue | name=result | multiplyresult |
[info] Executing: |echo | 'Multiplication Result : ${multiplyresult}' | |
[info] echo: 'Multiplication Result : 25 '
[info] Changed test case
[info] Executing: |open | file:///C:/Users/Savitha/Desktop/java%20script/Arithmetic%20operation.html | |
[info] Executing: |assertTitle | Arithmetic Operation | |
[info] Executing: |type | name=n1 | 25 |
[info] Executing: |type | name=n2 | 5 |
[info] Executing: |click | //input[@value='Divide'] | |
[info] Executing: |storeValue | name=result | divresult |
[info] Executing: |echo | 'Division Result :${divresult}' | |
[info] echo: 'Division Result :5'
```

## 3. Conduct a test suite for any two web sites

Test Suite
<a href="#">redbus</a>
<a href="#">search</a>

redbus		
open	http://www.redbus.in/?gclid=CLTz5Paalq4CFcN56wodgwIAIA	
assertTitle	ONLINE BUS TICKETS BOOKING SERVICE » ONLINE BUS TICKET RESERVATION	
click	id=DDLSource	
select	id=DDLSource	label=Chennai
select	id=DDLDestination	label=Bangalore
click	css=#tDestinations > option[value="210"]	
click	css=img.ui-datepicker-trigger	
click	link=19	
click	css=img.ui-datepicker-trigger	
clickAndWait	id=searchBtn	
assertTitle	Book Ticket - Search Buses	
assertTitle	Book Ticket - Search Buses	
storeTable	id=list5.0.6	a
echo	\${a}	

search		
open	/	
type	id=lst-ib	java
click	css=div.gsq_a > table > tbody > tr > td	
click	name=btnG	
storeText	id=resultStats	result
echo	\${result}	

OUTPUT:

redbus.html - Selenium IDE 1.6.0

Base URL: file:///C:/Users/Savitha/Desktop/java%20script/gcd.html

Command	Target	Value
select	id=DDLSource	label=Chennai
select	id=DDLDestination	label=Bangalore
click	css=#tDestinations > op...	
click	css=img.ui-datepicker-t...	
click	link=19	
click	css=img.ui-datepicker-t...	
clickAndWait	id=searchBtn	
assertTitle	Book Ticket - Search Bus...	
assertTitle	Book Ticket - Search Bus...	
storeTable	id=list5.0.6	a
echo	\${a}	

Runs: 1  
Failures: 0

```

Log Reference UI-Element Rollup Info Clear
[info] Executing: |select | id=DDLDestination | label=Bangalore |
[info] Executing: |click | css=#tDestinations > option[value="210"] | |
[info] Executing: |click | css=img.ui-datepicker-trigger | |
[info] Executing: |click | link=19 | |
[info] Executing: |click | css=img.ui-datepicker-trigger | |
[info] Executing: |clickAndWait | id=searchBtn | |
[info] Executing: |assertTitle | Book Ticket - Search Buses | |
[info] Executing: |assertTitle | Book Ticket - Search Buses | |
[info] Executing: |storeTable | id=list5.0.6 | a |
[info] Executing: |echo | ${a} | |
[info] echo: 555
    
```

ONLINE BUS TICKETS BOOKING SERVICE...

www.redbus.in/?gclid=CLT5PaaIq4CFcN56wodgwIAA

redBus.in

From: Chennai To: Bangalore

Date of Journey: 19-Feb-2012

Search buses

Call for bus tickets (7:30 am to 11 pm)

(044) 394 12345

redBus Advantages

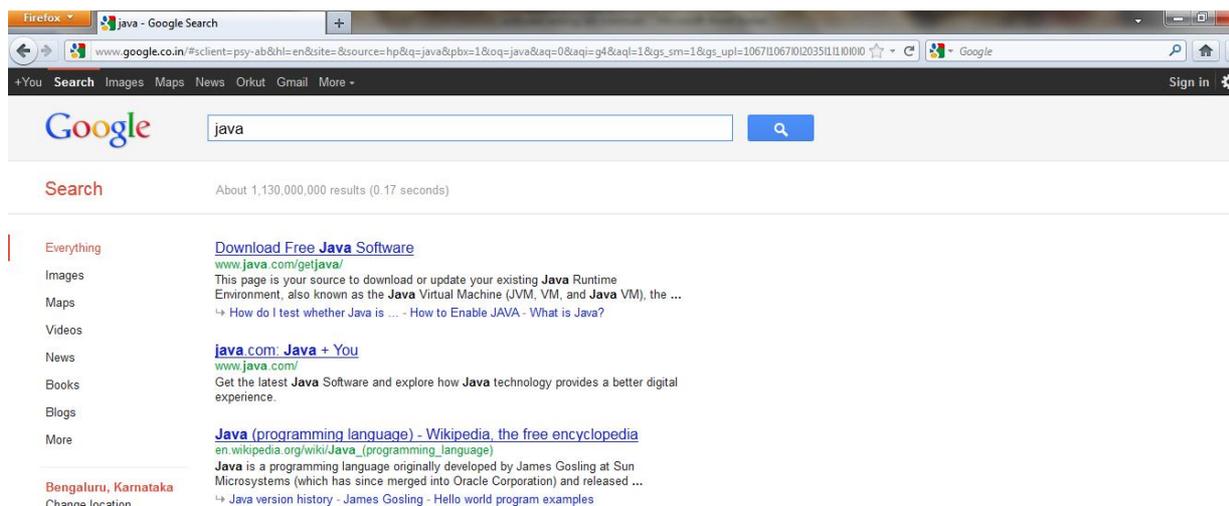
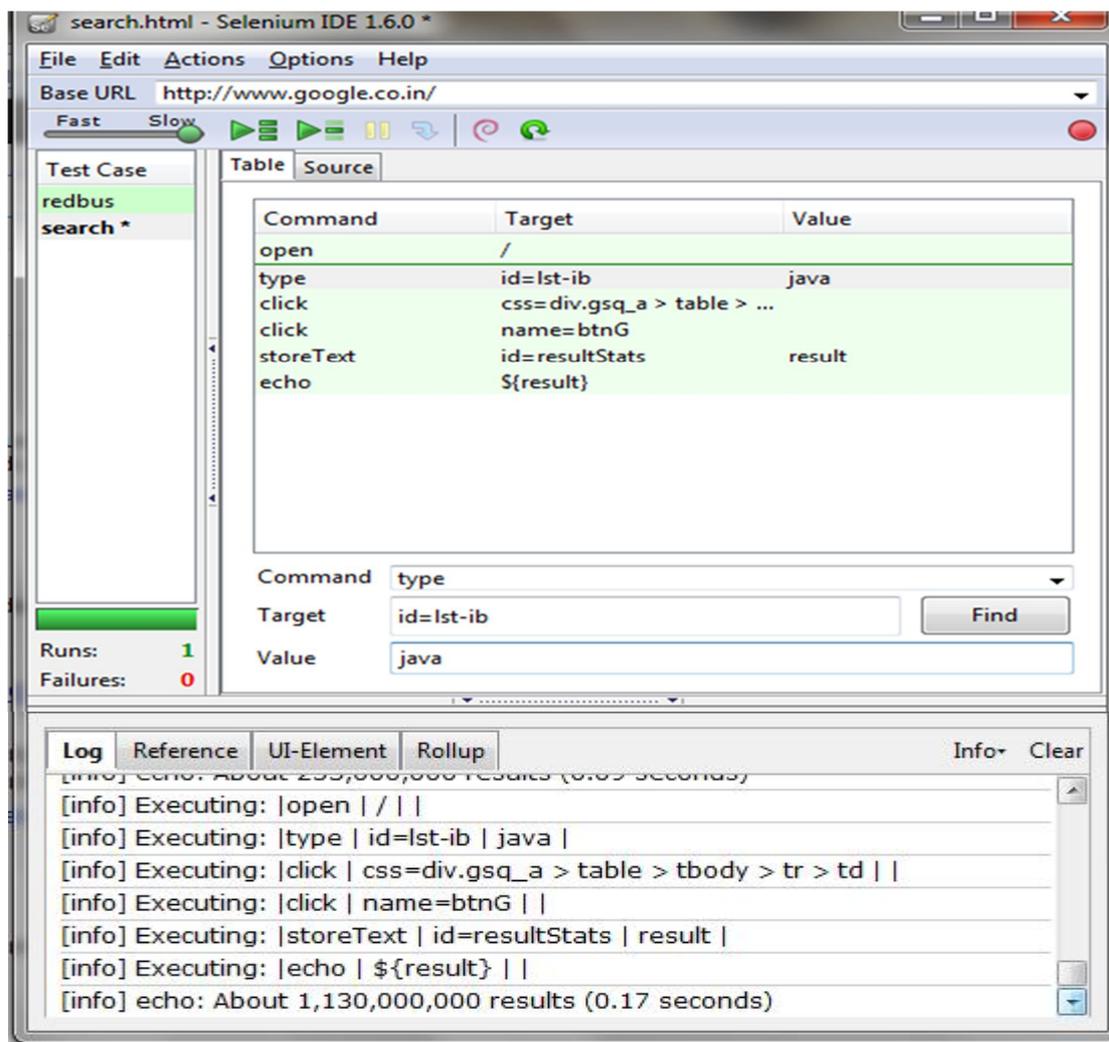
- Cash on delivery service of bus tickets in major cities
- Book online using credit, debit cards and net banking
- Book bus tickets online or over IVR through our call centers

Book Ticket - Search Buses

www.redbus.in/Booking/SelectBus1.aspx?fromCityId=122&fromCityName=Chennai&toCityId=122&toCityName=Bangalore&dej=19-Feb-2012&busType=An

Chennai-Bangalore | Sun, Feb 19

Travels	Bus type	Amenities	Dep. + Arr. Hrs.	Seats	Ratings	mTkt	Fare
Shama sardar Travels			06:45 AM	12 seats	★★★★☆		₹ 555
Volvo A/C Semi Sleeper (2+2)			02:15 PM	22 ratings			
canc. policy · address		photos · video	7.30 hrs		details		view seats



#### 4. Install Selenium server and demonstrate it using a script in Java/PHP

- Download the following :  
selenium-server-standalone-2.15.0  
selenium-java-client-driver  
from <http://seleniumhq.org/download/>  
save these two jar files under jdk1.6.0\bin (You can use jdk6 and above versions of java)
- Download 'Eclipse IDE for java developers'  
from <http://www.eclipse.org/downloads/>. This is not an installable file. so just copy the file under C: drive. whenever you need to write test code in eclipse just double click on the eclipse application inside the eclipse folder

Java script to develop a web page which calculates the GCD of 2 numbers. Save this file on desktop and save as GCD.html

```
<html>
<head>
<script type="text/javascript">
function gcd()
{
    var x,y;
    x=parseInt(document.myform.n1.value);
    y=parseInt(document.myform.n2.value);
    while(x!=y)
    {
        if(x>y)
            x=x-y;
        else
            y=y-x;
    }
    document.myform.result.value=x;
}
</script>
</head>
<body>
<h1 align="center"> Program to calculate gcd of two numbers </h1>
<hr color="red">
<center>
Enter two numbers :
<form name="myform">
Number 1 : <input type="text" name="n1" value=""> <br> <br>
Number 2 : <input type="text" name="n2" value=""> <br> <br>
<input type="button" value="Get GCD" onClick="gcd()"> <br> <br>
GCD is : <input type="text" name="result" value="">
</form>
</body>
</html>
```

**java test script for testing the gcd webpage. (write this code in eclipse environment).**

```
import com.thoughtworks.selenium.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
```

```

@SuppressWarnings("deprecation")
public class GcdTest extends SeleneseTestCase {
    @Before
    public void setUp() throws Exception {
        selenium = new DefaultSelenium("localhost", 4444, "*chrome",
"file:///C:/Users/Savitha/Desktop/GCD.html");
        selenium.start();
    }

    @Test
    public void testGcd() throws Exception {
        selenium.setSpeed("2000");
        selenium.open("file:///C:/Users/Savitha/Desktop/GCD.html");
        selenium.windowMaximize();
        selenium.type("name=n1", "5");
        selenium.type("name=n2", "6");
        selenium.click("css=input[type=\"button\"]");
        String result = selenium.getValue("name=result");
        System.out.println("the gcd is : " +result);
    }

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

```

Write the Above code in eclipse by following the below steps

- Open eclipse application-> give the folder name where you want to store your files. the default will be ' workspace '
- Then go to file->new->java project.
- You will get **Create java project** window , Name the project to **st** -> click **finish**.  
Double click on the new project **st** on the package explorer -> you will get **src** folder and JRE System Library
- right click on the **src** folder->go to new->click on the **class** -> you will get **new java class** window ->name the class as **GcdTest** -> click finish.

Note: Make sure that the path of your GCD.html file must be correct , in this example it is

```
file:///C:/Users/Savitha/Desktop/GCD.html
```

After writing this code if u get red lines under the selenium then include the selenium server and selenium java driver jar file. the steps are

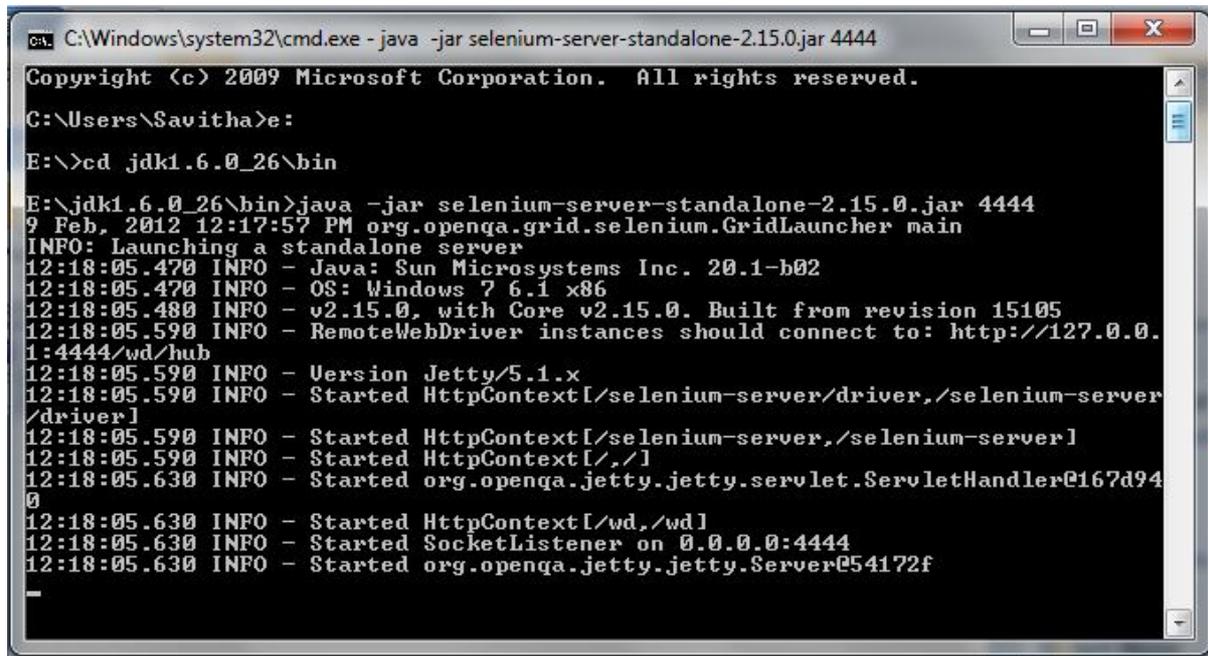
- Right click on the **st** folder-> click on **properties**-> click on **Java Build path**-> then select the tab **libraries**-> click on **Add External JARs**-> browse for **selenium-server-standalone-2.15.0**-> click **open**-> again click on **Add External JARs** -> browse for **selenium-java-client-driver**->click **open**-> click **ok**.

To run the above program

- First open the command prompt-> go to the jdk1.6.0\_26\bin folder -> run/write the command

```
Java -jar selenium-server-standalone-2.15.0.jar 4444
```

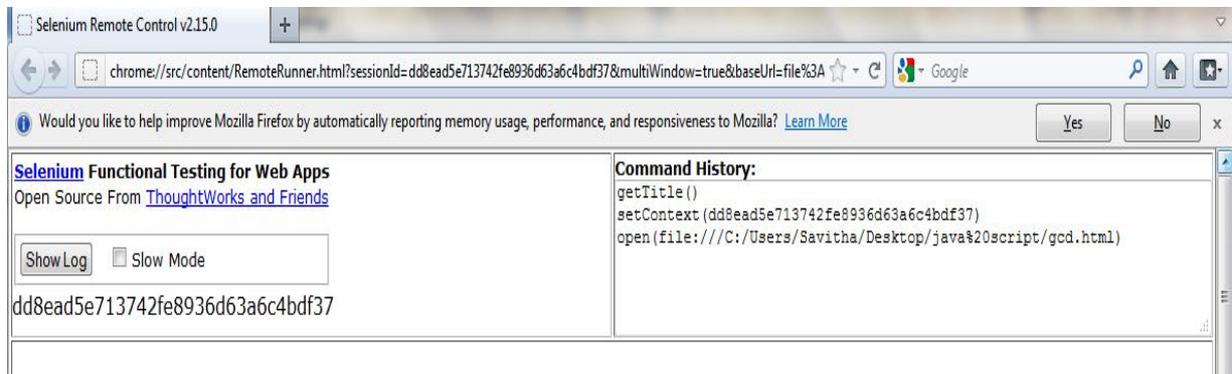
(In the above command 4444 is the local port number)

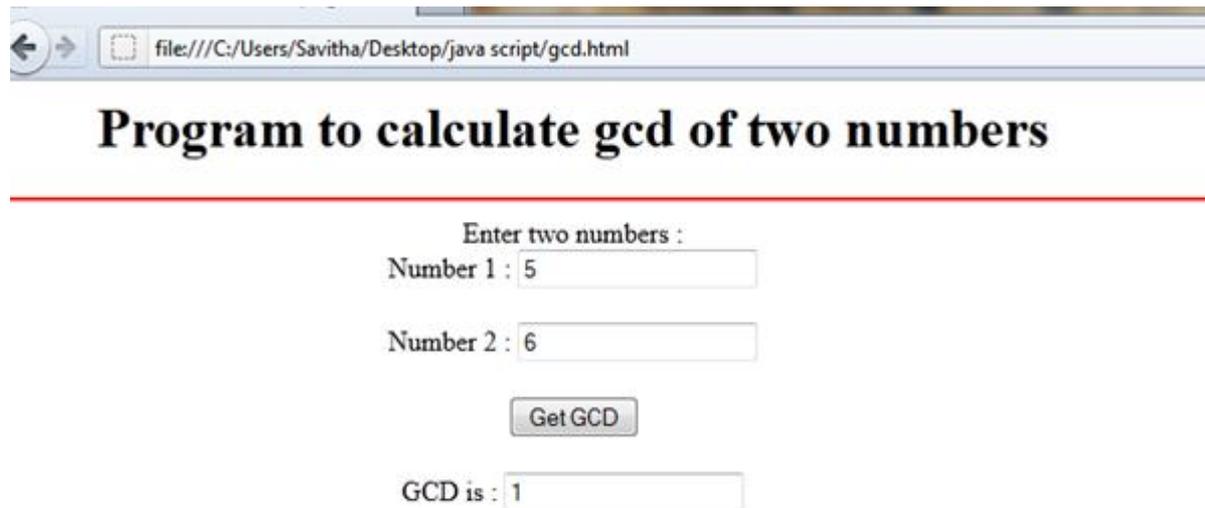


```
C:\Windows\system32\cmd.exe - java -jar selenium-server-standalone-2.15.0.jar 4444
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Savitha>e:
E:\>cd jdk1.6.0_26\bin
E:\jdk1.6.0_26\bin>java -jar selenium-server-standalone-2.15.0.jar 4444
9 Feb, 2012 12:17:57 PM org.openqa.grid.selenium.GridLauncher main
INFO: Launching a standalone server
12:18:05.470 INFO - Java: Sun Microsystems Inc. 20.1-b02
12:18:05.470 INFO - OS: Windows 7 6.1 x86
12:18:05.480 INFO - v2.15.0, with Core v2.15.0. Built from revision 15105
12:18:05.590 INFO - RemoteWebDriver instances should connect to: http://127.0.0.
1:4444/wd/hub
12:18:05.590 INFO - Version Jetty/5.1.x
12:18:05.590 INFO - Started HttpContext[/selenium-server/driver,/selenium-server
/driver/]
12:18:05.590 INFO - Started HttpContext[/selenium-server,/selenium-server/]
12:18:05.590 INFO - Started HttpContext[/,/]
12:18:05.630 INFO - Started org.openqa.jetty.jetty.servlet.ServletHandler@167d94
0
12:18:05.630 INFO - Started HttpContext[/wd,/wd/]
12:18:05.630 INFO - Started SocketListener on 0.0.0.0:4444
12:18:05.630 INFO - Started org.openqa.jetty.jetty.Server@54172f
```

Now your Selenium RC Server has started to run.

Next step is to run your testing program in eclipse. For that click on the green run button in eclipse.





file:///C:/Users/Savitha/Desktop/java script/gcd.html

## Program to calculate gcd of two numbers

---

Enter two numbers :

Number 1 :

Number 2 :

GCD is :

**OUTPUT:**

You can see the output in the eclipse console as follows



```
<terminated> Gcd [JUnit] C:\Program Files\Java\jre6\bin\javaw.exe (11-Feb-2012 8:10:02 PM)
the gcd is : 1
```

## 5. Write and test a program to login a specific web page

(Write this java test code in eclipse. before running this code makes sure your selenium RC server is running)

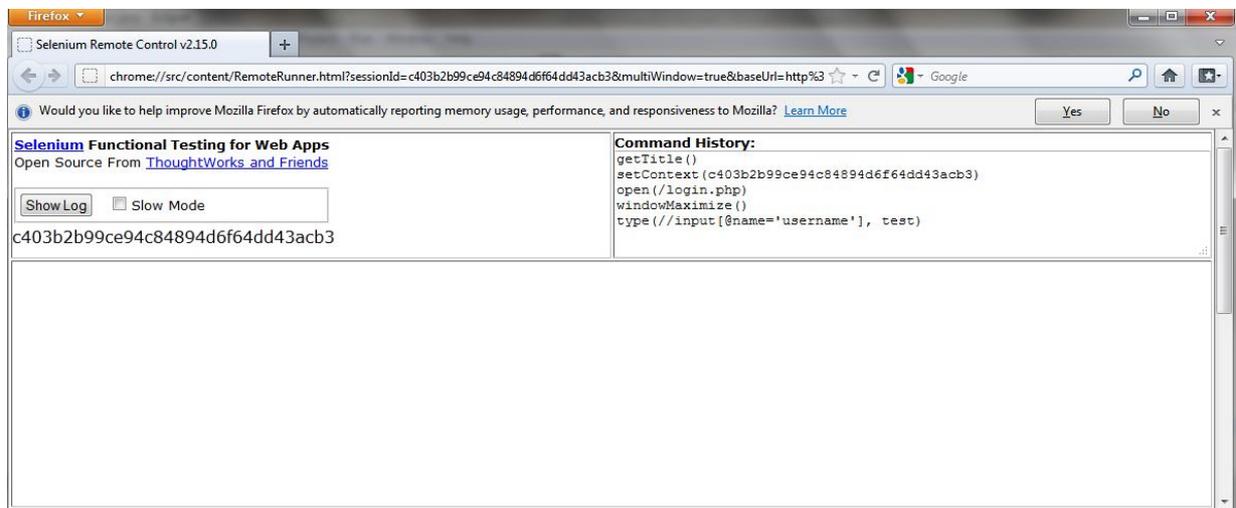
```
import com.thoughtworks.selenium.*;
@SuppressWarnings("deprecation")
public class login extends SeleneseTestCase {
    public void setUp() throws Exception {
        setUp("http://php.thedemosite.co.uk/", "*firefox");
    }

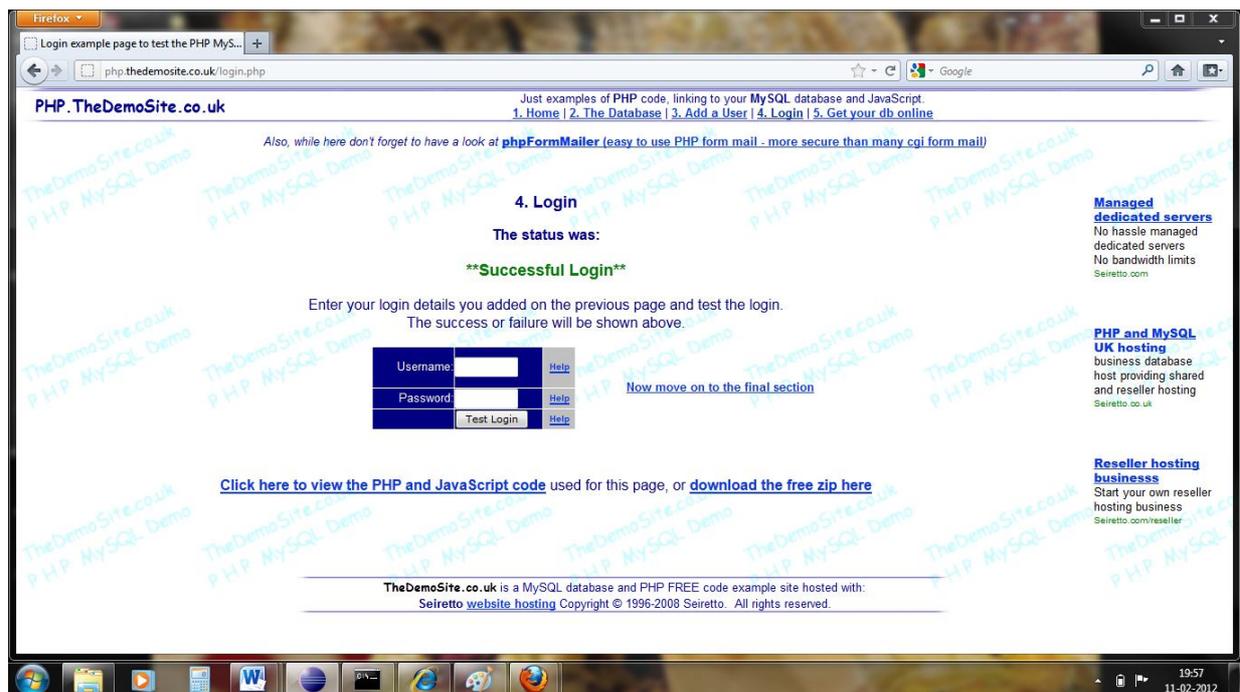
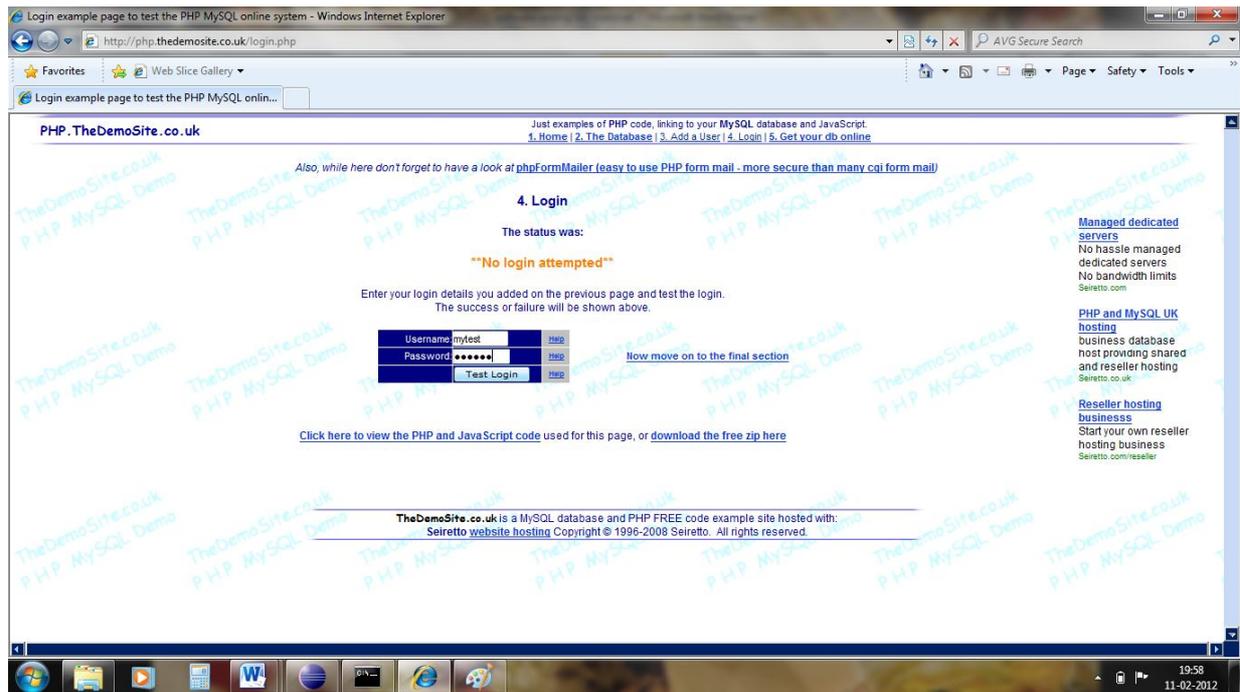
    public void testloginlogout(){

        selenium.setSpeed("2000");
        selenium.open("/login.php");
        selenium.windowMaximize();
        selenium.type("//input[@name='username']", "mytest");
        selenium.type("//input[@name='password']", "mytest");
        selenium.click("//input[@type='button']");
        selenium.waitForPageToLoad("50000");

    }
}
```

### OUTPUT:





**6. Write and test a program to update 10 student records into table into Excel file**

- Write this java test code in eclipse. Since we are not using any Selenium object here no need of selenium RC Server to run at the command prompt for this program.
- This program will create an Excel file Student.xls under the desktop.
- Since we are dealing with the excel, we need to add the external jar file jxl-2.6 in to the java project.( download jxl-2.6 and Follow the same steps as we used to add selenium server and selenium java driver to our java project refer page no 10 of this manual.)

```
import java.io.File;
import java.io.IOException;
import java.util.Locale;

import jxl.CellView;
import jxl.Workbook;
import jxl.WorkbookSettings;
import jxl.format.UnderlineStyle;
import jxl.write.Formula;
import jxl.write.Label;
import jxl.write.Number;
import jxl.write.WritableCellFormat;
import jxl.write.WritableFont;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import jxl.write.WriteException;
import jxl.write.biff.RowsExceededException;
```

```
public class student_excel_write {

    private WritableCellFormat timesBoldUnderline;
    private WritableCellFormat times;
    private String inputFile;

    public void setOutputFile(String inputFile) {
        this.inputFile = inputFile;
    }

    public void write() throws IOException, WriteException {
        File file = new File(inputFile);
        WorkbookSettings wbSettings = new WorkbookSettings();

        wbSettings.setLocale(new Locale("en", "EN"));

        WritableWorkbook workbook = Workbook.createWorkbook(file, wbSettings);
        workbook.createSheet("Report", 0);
        WritableSheet excelSheet = workbook.getSheet(0);
        createLabel(excelSheet);
        createContent(excelSheet);

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

```

private void createLabel(WritableSheet sheet)
    throws WriteException {
    // Lets create a times font
    WritableFont times10pt = new WritableFont(WritableFont.TIMES, 10);
    // Define the cell format
    times = new WritableCellFormat(times10pt);
    // Lets automatically wrap the cells
    times.setWrap(true);

    // Create a bold font with underlines

    WritableFont times10ptBoldUnderline = new WritableFont(
        WritableFont.TIMES, 10, WritableFont.BOLD, false,
        UnderlineStyle.SINGLE);
    timesBoldUnderline = new WritableCellFormat(times10ptBoldUnderline);
    // Lets automatically wrap the cells
    timesBoldUnderline.setWrap(true);

    CellView cv = new CellView();
    cv.setFormat(times);
    cv.setFormat(timesBoldUnderline);
    cv.setAutosize(true);

    // Write a few headers
    addCaption(sheet, 0, 0, "Student Name");
    addCaption(sheet, 1, 0, "Subject 1");
    addCaption(sheet, 2, 0, "subject 2");
    addCaption(sheet, 3, 0, "subject 3");
}

private void createContent(WritableSheet sheet) throws WriteException,
    RowsExceededException {
    // Write a few number
    for (int i = 1; i < 10; i++) {
        // First column
        addLabel(sheet, 0, i, "Student " + i);
        // Second column
        addNumber(sheet, 1, i, ((i*i)+10));
        addNumber(sheet, 2, i, ((i*i)+4));
        addNumber(sheet, 3, i, ((i*i)+3));
    }
}

private void addCaption(WritableSheet sheet, int column, int row, String s)
    throws RowsExceededException, WriteException {
    Label label;
    label = new Label(column, row, s, timesBoldUnderline);
    sheet.addCell(label);
}

private void addNumber(WritableSheet sheet, int column, int row,
    Integer integer) throws WriteException, RowsExceededException {
    Number number;

```

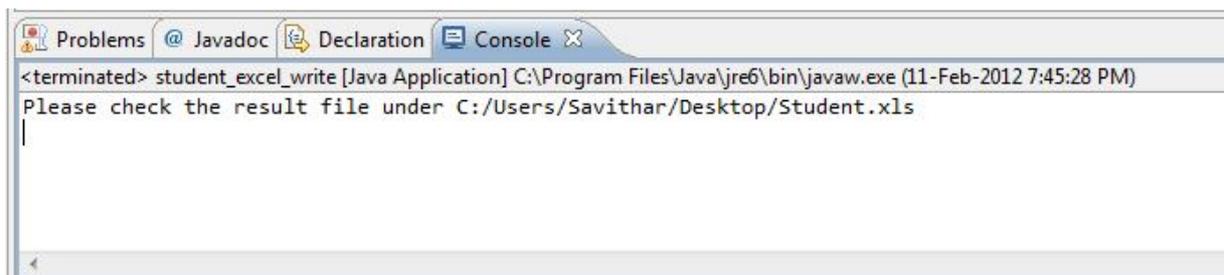
```

        number = new Number(column, row, integer, times);
        sheet.addCell(number);
    }

    private void addLabel(WritableSheet sheet, int column, int row, String s)
        throws WriteException, RowsExceededException {
        Label label;
        label = new Label(column, row, s, times);
        sheet.addCell(label);
    }

    public static void main(String[] args) throws WriteException, IOException {
        student_excel_write test = new student_excel_write();
        test.setOutputFile("C:/Users/Savitha/Desktop/Student.xls");
        test.write();
        System.out.println("Please check the result file under
C:/Users/Savithar/Desktop/Student.xls ");
    }
}

```

**OUTPUT:**

	A	B	C	D
1	<u>Student Name</u>	<u>Subject 1</u>	<u>subject 2</u>	<u>subject 3</u>
2	Student 1	11	5	4
3	Student 2	14	8	7
4	Student 3	19	13	12
5	Student 4	26	20	19
6	Student 5	35	29	28
7	Student 6	46	40	39
8	Student 7	59	53	52
9	Student 8	74	68	67
10	Student 9	91	85	84
11				

### 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 )

- Write this java test code in eclipse. Since we are not using any Selenium object here no need of selenium RC Server to run at the command prompt for this program.
- This program reads an excel file so Create an Excel file Student.xls under the desktop as follows.
- Since we are dealing with the excel, we need to add the external jar file jxl-2.6 in to the java project.( download jxl-2.6 and Follow the same steps as we used to add selenium server and selenium java driver to our java project refer page no 10 of this manual.)

	A	B	C	D	E
	<u>Student</u>				
1	<u>Name</u>	<u>Subject 1</u>	<u>subject 2</u>	<u>subject 3</u>	
2	Student 1	35	67	60	
3	Student 2	36	46	57	
4	Student 3	59	48	58	
5	Student 4	80	80	60	
6	Student 5	35	29	28	
7	Student 6	46	40	39	
8	Student 7	59	53	52	
9	Student 8	74	68	67	
10	Student 9	91	85	84	
11					
12					
13					
14					

```

import java.io.File;
import java.io.IOException;

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

public class student_excel_read {

    private String inputFile;

    public void setInputFile(String inputFile) {
        this.inputFile = inputFile;
    }

    public void read() throws IOException {
        File inputWorkbook = new File(inputFile);
        Workbook w;

```

```
boolean flag=false;
int count=0;
try {
    w = Workbook.getWorkbook(inputWorkbook);
    // Get the first sheet
    Sheet sheet = w.getSheet(0);
    // Loop over first 10 column and lines

    for (int j = 0; j < sheet.getRows(); j++) {
        for (int i = 0; i < sheet.getColumns(); i++) {
            Cell cell = sheet.getCell(i, j);

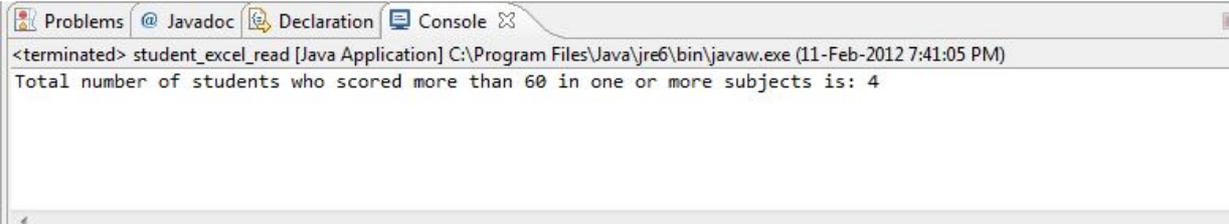
            if (cell.getType() == CellType.NUMBER) {
                if(Integer.parseInt(cell.getContents())>60){
                    flag = true;
                    if(flag == true){
                        count++;
                        flag=false;
                    }
                    break;
                }
            }
        }
    }

    System.out.println("Total number of students who scored more than 60 in one
or more subjects is: " +count);
} catch (BiffException e) {
    e.printStackTrace();
}

}

public static void main(String[] args) throws IOException {
    student_excel_read test = new student_excel_read();
    test.setInputFile("C:/Users/Savitha/Desktop/Student.xls");
    test.read();
}
}
```

### OUTPUT:



The screenshot shows a Java IDE console window with the following output:

```
<terminated> student_excel_read [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (11-Feb-2012 7:41:05 PM)
Total number of students who scored more than 60 in one or more subjects is: 4
```

## 8. Write and test a program to provide total number of objects present / available on the page

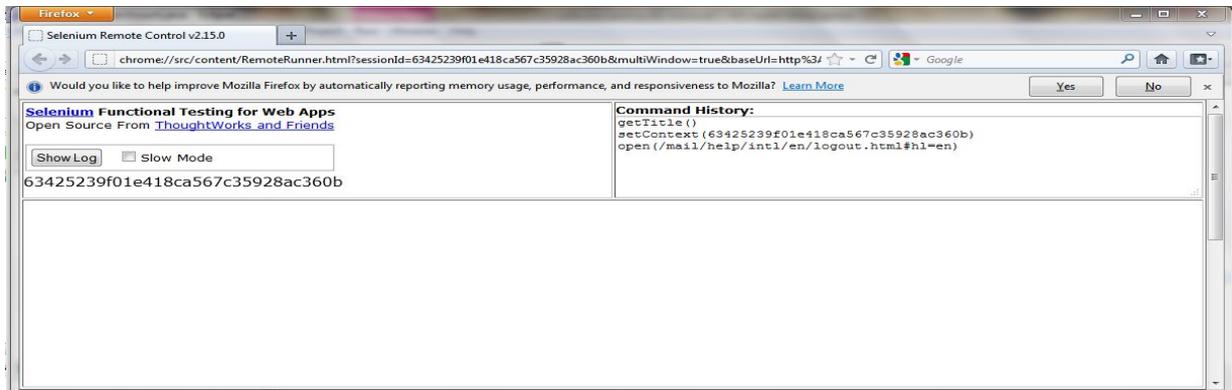
- Write this java test code in eclipse. before running this code makes sure your selenium RC server is running

```
import com.thoughtworks.selenium.*;
@SuppressWarnings("deprecation")
public class objectcount extends SeleniumTestCase {
    public void setUp() throws Exception {
        setUp("http://www.google.com/", "*firefox");
    }

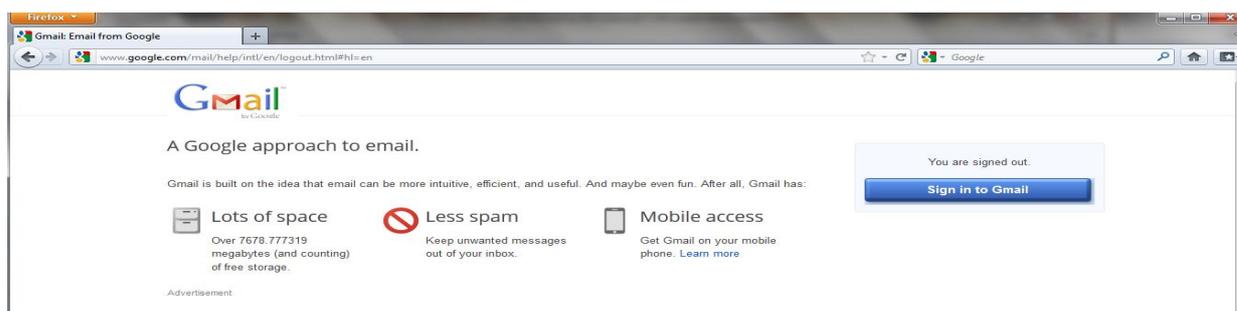
    public void testloginlogout(){

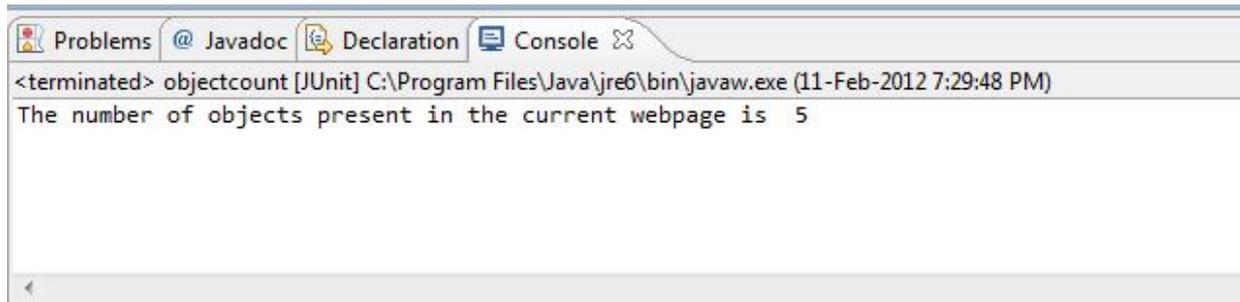
        selenium.setSpeed("1000");

        selenium.open("/mail/help/intl/en/logout.html#hl=en");
        selenium.waitForPageToLoad("30000");
        selenium.windowMaximize();
        int num = selenium.getXpathCount("//p").intValue();
        System.out.println("The number of option elements present are " +num);
    }
}
```



## OUTPUT:





The screenshot shows a console window with a tab labeled 'Console'. The text in the console is as follows:

```
<terminated> objectcount [JUnit] C:\Program Files\Java\jre6\bin\javaw.exe (11-Feb-2012 7:29:48 PM)  
The number of objects present in the current webpage is 5
```

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

- Write this java test code in eclipse. before running this code makes sure your selenium RC server is running)
- Write Combocount.html as shown below and save under desktop

```
<html>
<body>
<select>
  <option>Volvo</option>
  <option>Express</option>
  <option>Mercedes</option>
<option>RajaHamsa</option>
</select>
</body>
</html>
```

- Write the below code in eclipse
- Make sure that the path of your combocount.html file must be correct for your system , in this example it is

file:///C:/Users/Savitha/Desktop/combocount.html

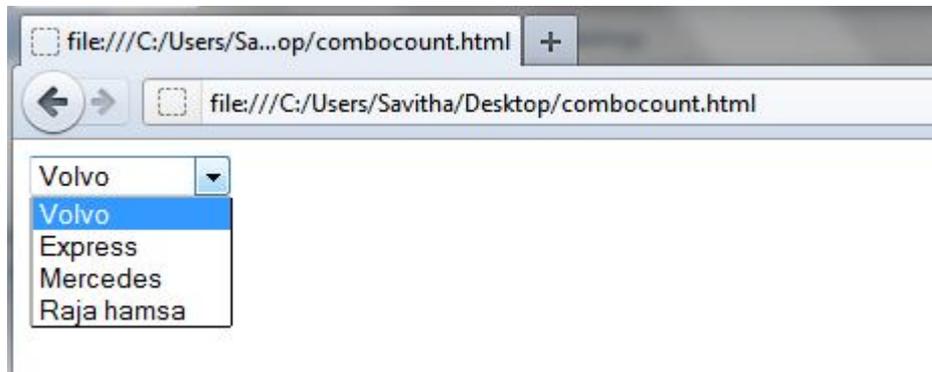
```
import com.thoughtworks.selenium.*;
public class Comboitemcount extends SeleneseTestCase {
    public void setUp() throws Exception {
        selenium = new DefaultSelenium("localhost", 4444, "*chrome",
"file:///C:/Users/Savitha/Desktop/combocount.html");
        selenium.start();}

    public void testloginlogout(){

        selenium.setSpeed("1000");

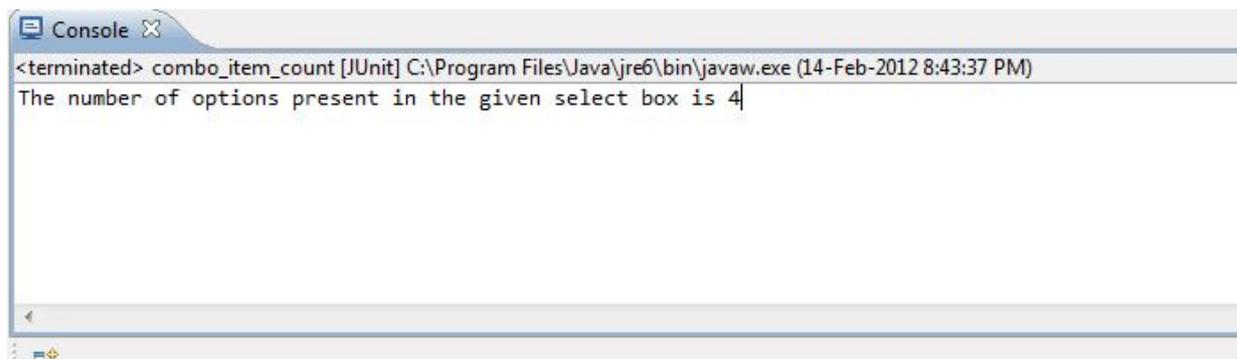
        selenium.open("file:///C:/Users/Savitha/Desktop/combocount.html");
        selenium.waitForPageToLoad("30000");
        selenium.windowMaximize();
        String[] selectelements = new String[1000];
        selectelements= selenium.getSelectOptions("//select");
        System.out.println("The number of options present in the given select box is "
+selectelements.length);

    }
}
```



### OUTPUT:

You can see the output in the eclipse console as follows



**10. Write and test a program to count number of items present on a desktop**

- Write this java test code in eclipse. Since we are not using any Selenium object here no need of selenium RC Server to run at the command prompt for this program.
- Write a visual basic script with the following code and save as count.vbs under the desktop.

```
Set fso = createobject("Scripting.FileSystemObject")
DesktopPath = CreateObject("WScript.Shell").SpecialFolders("Desktop")
'Files count
msgbox "The number of files on the desktop is : " &fso.GetFolder(DesktopPath).Files.Count
'Folders count
msgbox "The number of folders on the desktop is : " &fso.GetFolder(DesktopPath).SubFolders.Count
```

- In eclipse write the following code.

```
import java.io.*;

public class desktop_item_count {

public static void main(String[] args) {

try {

Runtime.getRuntime().exec("wscript C:/Users/Savitha/Desktop/count.vbs");

} catch (IOException e) {

System.exit(0);

}

}

}
```

Note: Don't forget to change the desktop file path for your system

**OUTPUT**

