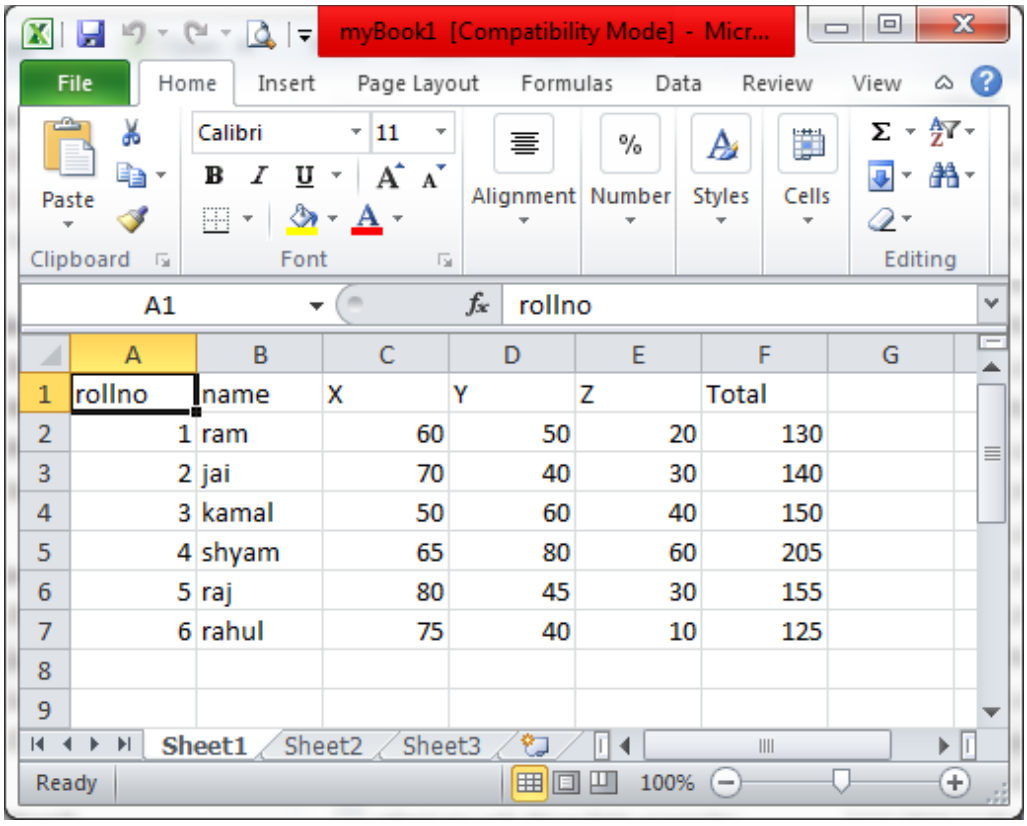
Practical No.05

Q.Write and test a program to update 10 student records into table into Excel file (using TestNG)

```
importjxl.Cell;
importjxl.Sheet;
importjxl.Workbook;
importjxl.read.biff.BiffException;

//Code to update 10 student records into table into Excel file
importorg.testng.annotations.BeforeClass;
importorg.testng.annotations.Test;
importjxl.*;
importjxl.read.*;
importjxl.write.*;
import java.io.*;

public class updatestudrecords {
        @BeforeClass
        public void f1()
        {}
        @Test
        public void testImportexport1() throws Exception {
        FileInputStream fi = new FileInputStream("D:\\selenium
        pracs\\myBook1.xls");
        Workbook w = Workbook.getWorkbook(fi);
        Sheet s = w.getSheet(0);
        String a[][] = new String[s.getRows()][s.getColumns()];
        FileOutputStreamfo = new FileOutputStream("D:\\selenium
        pracs\\myBook1res.xls");
        WritableWorkbookwwb = Workbook.createWorkbook(fo);
        WritableSheetws = wwb.createSheet("result1", 0);
        for (inti = 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);
        }
```

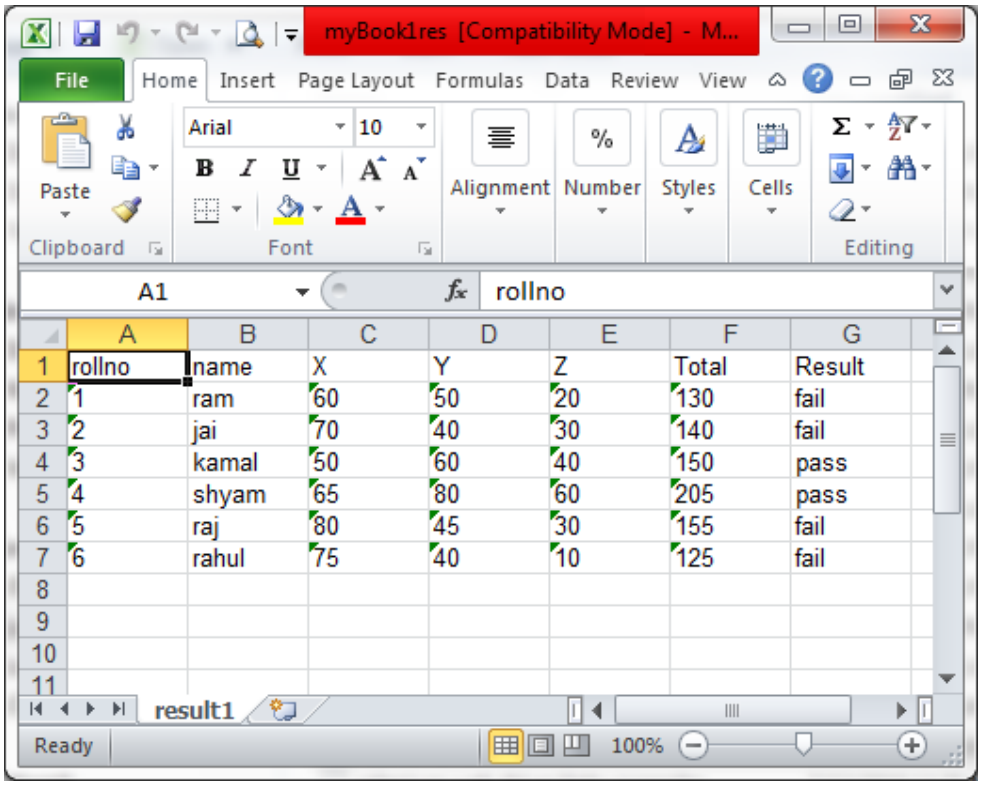
```
for (inti = 1; i<s.getRows(); i++) {
        for (int j = 2; j <s.getColumns(); j++)
        {
        a[i][j] = s.getCell(j, i).getContents();
        int x=Integer.parseInt(a[i][j]);
        if(x > 35)
        {
        Label l1 = new Label(6, i, "pass");
        ws.addCell(l1);
        }
        else
        {
        Label l1 = new Label(6, i, "fail");
        ws.addCell(l1);
        break;   }                }
        }
        wwb.write();
        wwb.close();              }        }
```

Input:-



	A	B	C	D	E	F	G
1	rollno	name	X	Y	Z	Total	
2	1	ram	60	50	20	130	
3	2	jai	70	40	30	140	
4	3	kamal	50	60	40	150	
5	4	shyam	65	80	60	205	
6	5	raj	80	45	30	155	
7	6	rahul	75	40	10	125	
8							
9							

Output:-



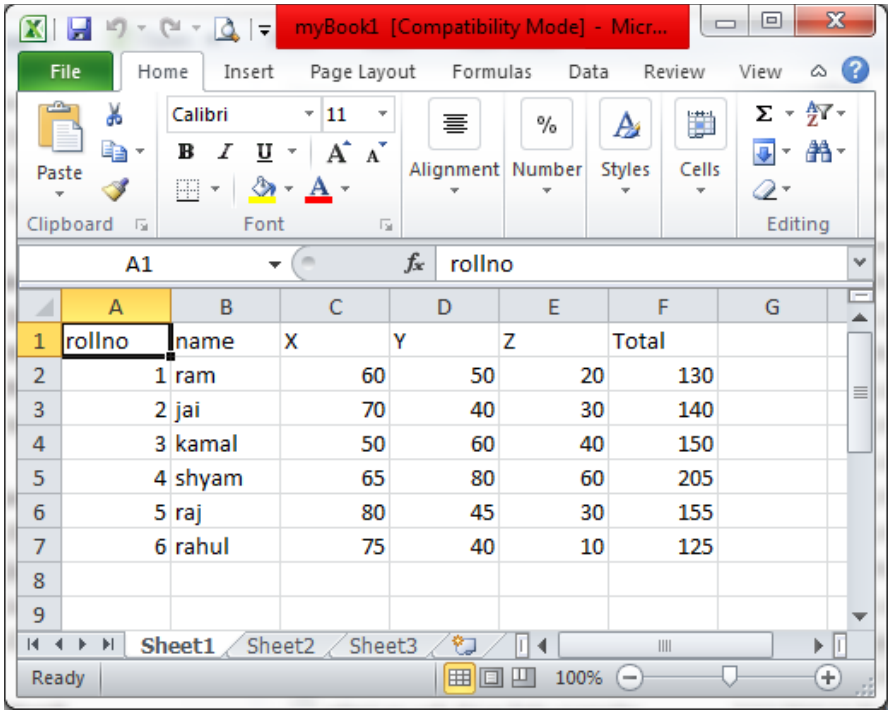
	A	B	C	D	E	F	G
1	rollno	name	X	Y	Z	Total	Result
2	1	ram	60	50	20	130	fail
3	2	jai	70	40	30	140	fail
4	3	kamal	50	60	40	150	pass
5	4	shyam	65	80	60	205	pass
6	5	raj	80	45	30	155	fail
7	6	rahul	75	40	10	125	fail
8							
9							
10							
11							

Practical No.06

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

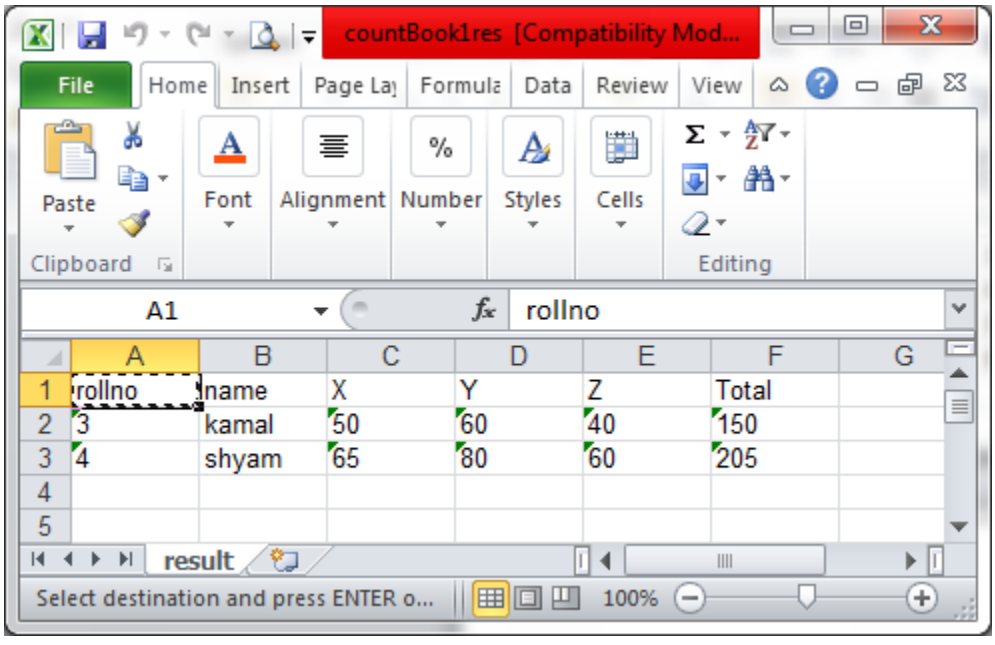
```
importjxl.*;
importjxl.read.*;
importjxl.write.*;
import java.io.*;
importorg.testng.annotations.Test;
public class countstuds {
	@Test
	public void testImportexport1() throws Exception {
	FileInputStream fi = new FileInputStream("D:\\selenium
	pracs\\myBook1.xls");
	Workbook w = Workbook.getWorkbook(fi);
	Sheet s = w.getSheet(0);
	String a[][] = new String[s.getRows()][s.getColumns()];
	FileOutputStreamfo = new FileOutputStream("D:\\selenium
	pracs\\countBook1res.xls");
	WritableWorkbookwwb = Workbook.createWorkbook(fo);
	WritableSheetws = wwb.createSheet("result", 0);
	int c=0;
	for (inti = 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:-



	A	B	C	D	E	F	G
1	rollno	name	X	Y	Z	Total	
2	1	ram	60	50	20	130	
3	2	jai	70	40	30	140	
4	3	kamal	50	60	40	150	
5	4	shyam	65	80	60	205	
6	5	raj	80	45	30	155	
7	6	rahul	75	40	10	125	
8							
9							

Output:-



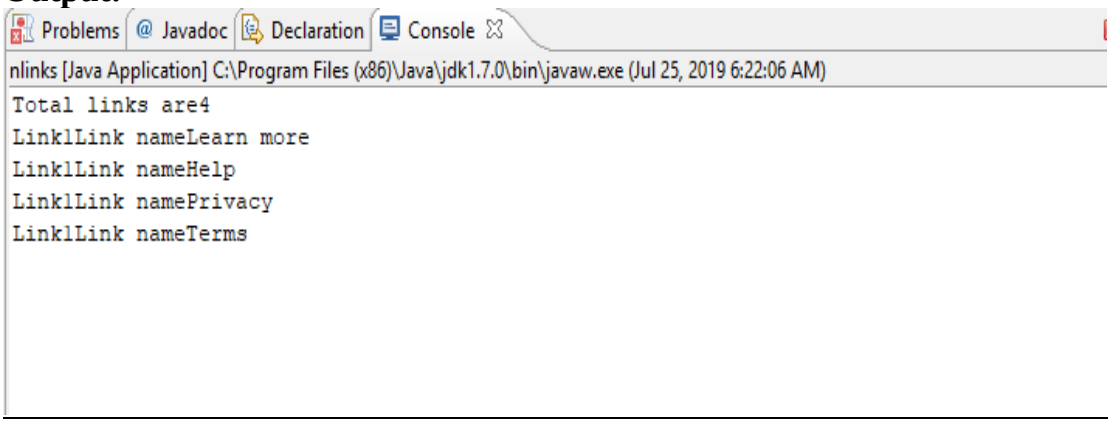
	A	B	C	D	E	F	G
1	rollno	name	X	Y	Z	Total	
2	3	kamal	50	60	40	150	
3	4	shyam	65	80	60	205	
4							
5							

Practical No.07

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

```
importorg.openqa.selenium.By;
importorg.openqa.selenium.WebDriver;
importorg.openqa.selenium.WebElement;
importorg.openqa.selenium.firefox.FirefoxDriver;
importorg.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.setCapability("marionette",true);
driver= new FirefoxDriver(capabilities);
driver.get("http://gmail.com/”);


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

Output:-



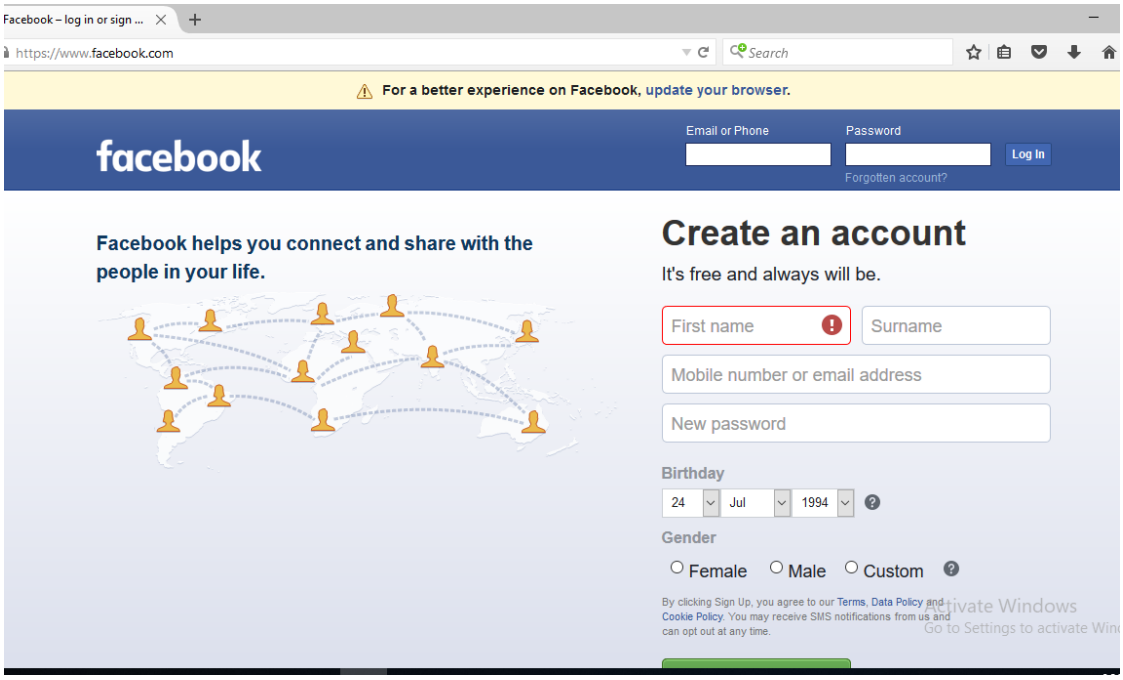
Practical No.08

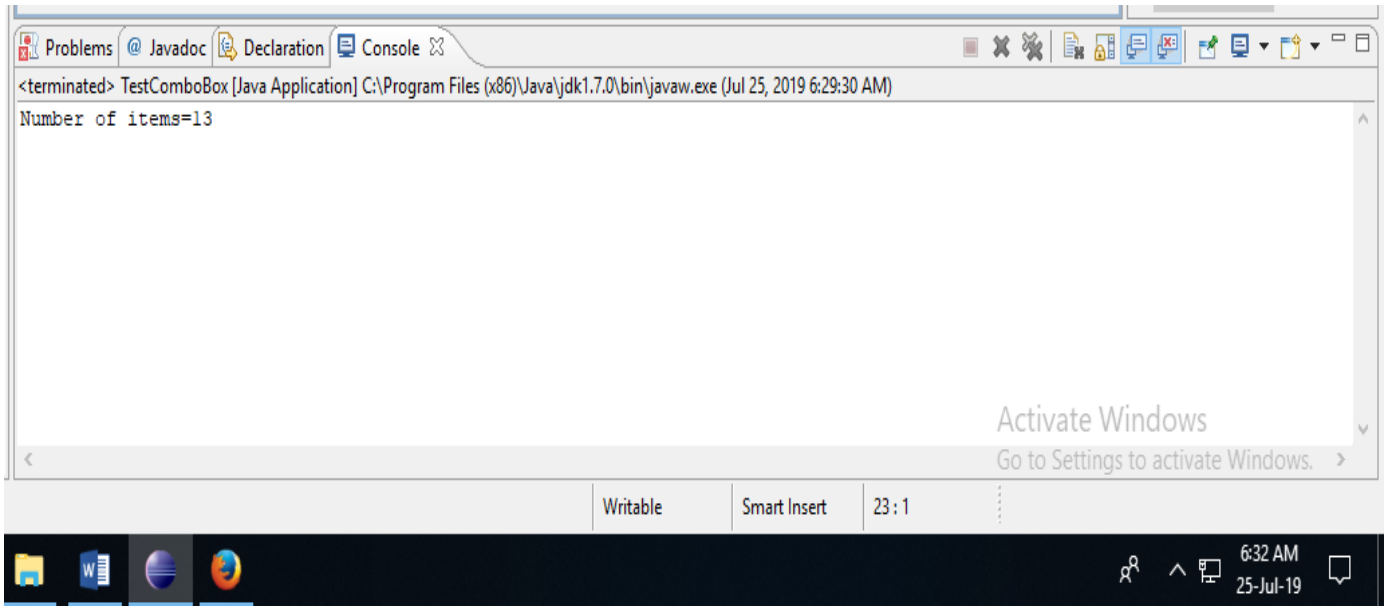
Q.Write and test a program to get the number of items in a list / combo box.

```
importjava.util.*;
importorg.openqa.selenium.By;
importorg.openqa.selenium.WebDriver;
importorg.openqa.selenium.WebElement;
importorg.openqa.selenium.firefox.FirefoxDriver;
importorg.openqa.selenium.support.ui.Select;
publicclassTestComboBox {

                static String driverPath="F:\\st\\geckodriver\\geckodriver-v0.24.0-win64";
        publicstatic WebDriver driver;
        publicstaticvoid main(String[] args)
        {
// TODO auto generated method stub
        System.setProperty("webdriver.gecto.driver",driverPath);
        WebDriver driver=newFirefoxDriver();
        driver.get("https://www.facebook.com/");
        Select se=newSelect(driver.findElement(By.xpath("//Select[@id='month']")));
        List <WebElement>mylist=se.getOptions();
        mylist.size();
        System.out.println("Number of items="+mylist.size());
        }
}
```

OUTPUT:

Number of Items=13





Practical No.09

Q. Write and test a program to count the number of check boxes on the page checked and unchecked count.

****First create a html file using Notepad****
Practchk.html

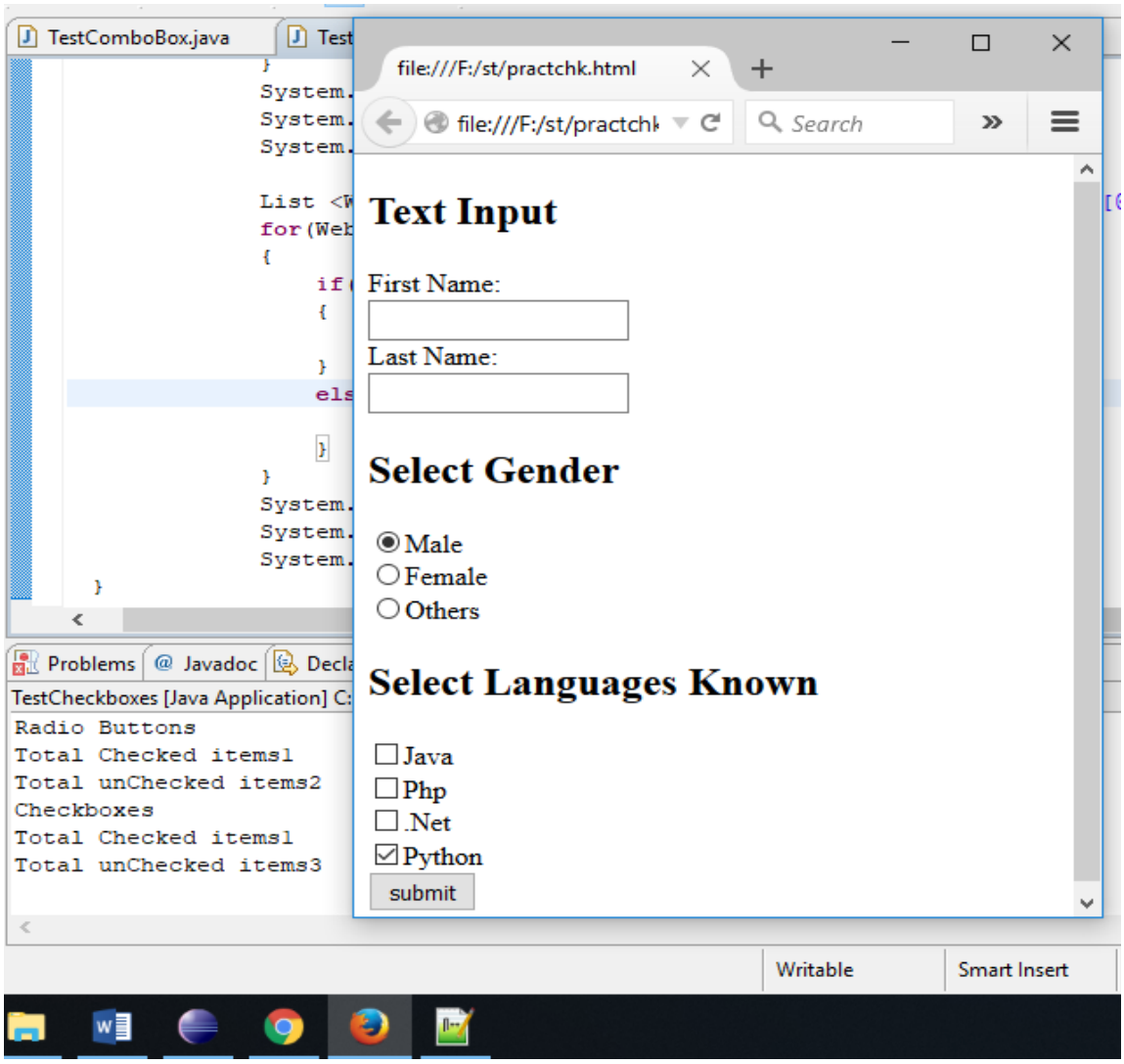
```
<!DOCTYPE html>
<html>
<body>
<form>
<h2>Text Input</h2>
        First Name:</br>
<input type="text" name="Firstname">
</br>
        Last Name:</br>
<input type="text" name="lastname">
</br>
<h2>Select Gender</h2>
<input type="radio" name="gender" value="male" checked>Male</br>
<input type="radio" name="gender" value="female">Female</br>
<input type="radio" name="gender" value="others">Others</br>

<h2>Select Languages Known</h2>
<input type="checkbox" name="lang" value="Java">Java</br>
<input type="checkbox" name="lang" value="Php">Php</br>
<input type="checkbox" name="lang" value="ASP.net">.Net</br>
<input type="checkbox" name="lang" value="Python" checked="checked">Python</br>
<input type="submit"  value="submit"></br>
</form>
</body>
</html>
```

Eclipsecode:

```
import java.util.*;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class TestCheckboxes {

static String driverPath="F:\\st\\geckodriver\\geckodriver-v0.24.0-win64";
public static WebDriver driver;
public static void main(String[] args)
{
        // TODO auto generated method stub
System.setProperty("webdriver.gecto.driver",driverPath);
WebDriver driver=new FirefoxDriver();
driver.get("file:///F:/st/practchk.html");
int radiochk=0,checkboxchk=0;
int radiounchk=0,checkboxunchk=0;
List<WebElement>els=driver.findElements(By.xpath("//input[@type='radio']"));
for(WebElementel:els)
{
        if(el.isSelected())
            {
                radiochk++;
            }
        else{
                radiounchk++;
            }
}
System.out.println("Radio Buttons");
System.out.println("Total Checked items"+ radiochk);
System.out.println("Total unChecked items"+ radiounchk);

List<WebElement>ebox=driver.findElements(By.xpath("//input[@type='checkbox']"
));
for(WebElementel:ebox)
{
        if(el.isSelected())
            {
                checkboxchk++;
            }
        else{
                checkboxunchk++;
            }
}
```

```
        System.out.println("Checkboxes");
        System.out.println("Total Checked items"+ checkboxchk);
        System.out.println("Total unChecked items"+ checkboxunchk);
        }
}
```

Output:



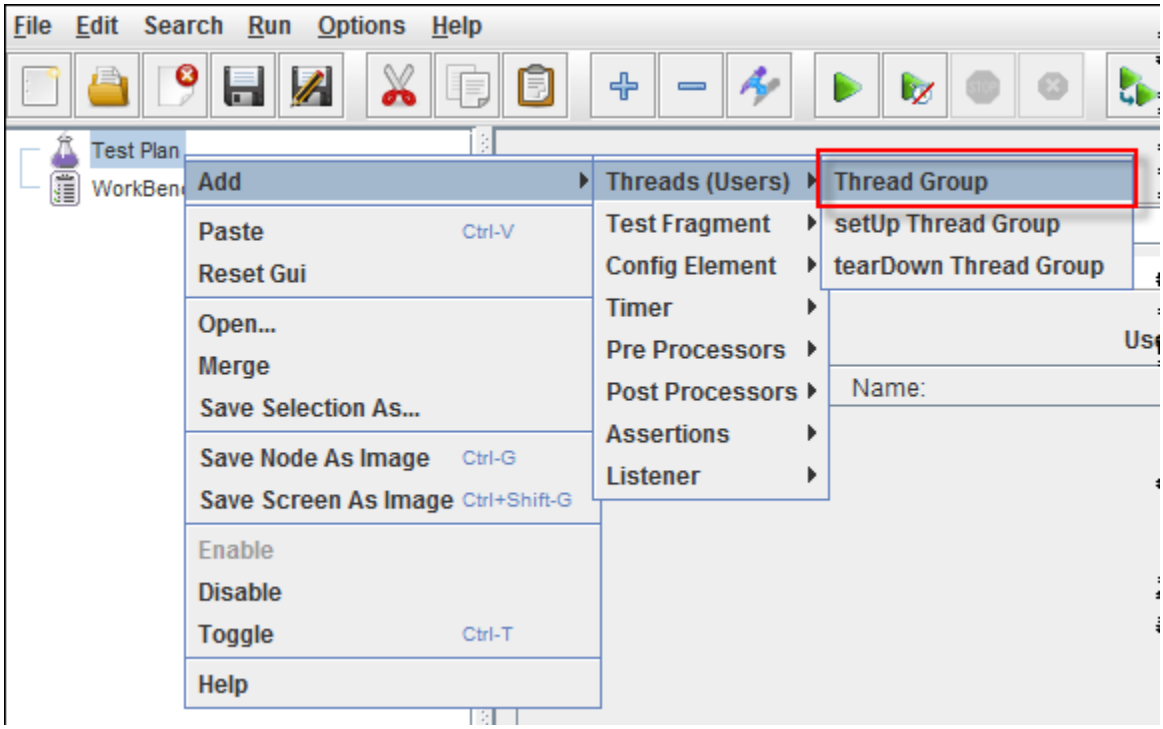
Practical No.10

Aim:- Load Testing using JMeter

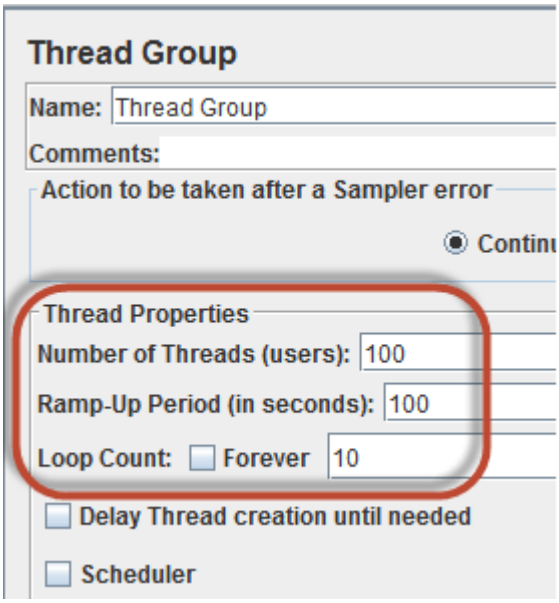
Step 1) Add Thread Group

1. Start **JMeter**
2. Select **Test Plan** on the tree
3. Add **Thread Group**

Right click on the "Test Plan" and add a new thread group: **Add** -> **Threads (Users)** -> **Thread Group**



In the Thread Group control panel, enter Thread Properties as follows:



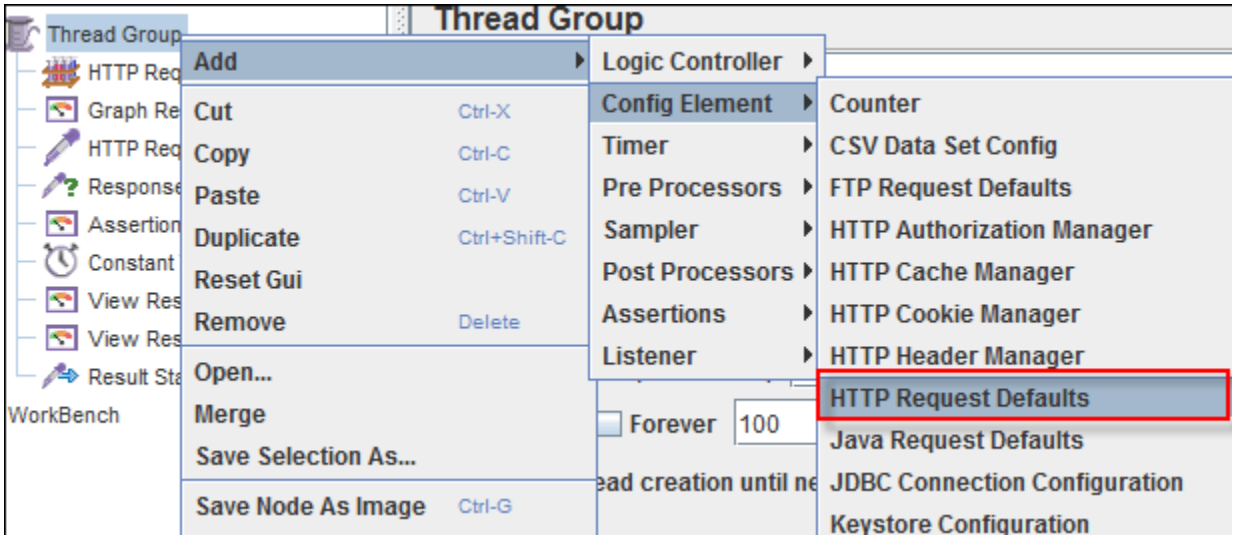
- **Number of Threads**: 100 (Number of users connects to the target website: 100)
- **Loop Count**: 10 (Number of time to execute testing)
- **Ramp-Up Period**: 100

Step 2) Adding JMeter elements

Now we determine what JMeter elements in this test. The elements are

- **HTTP request Default**

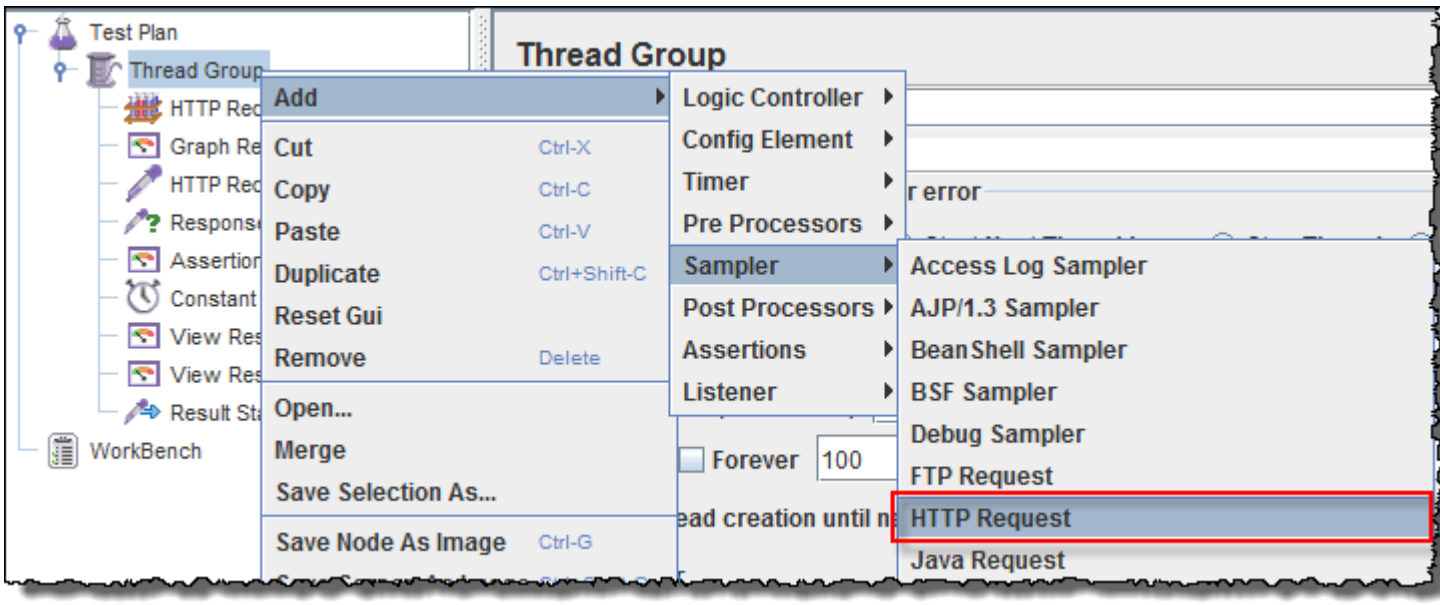
This element can be added by right-clicking on the Thread Group and selecting: **Add** -> **Config Element** -> **HTTP Request Defaults.**



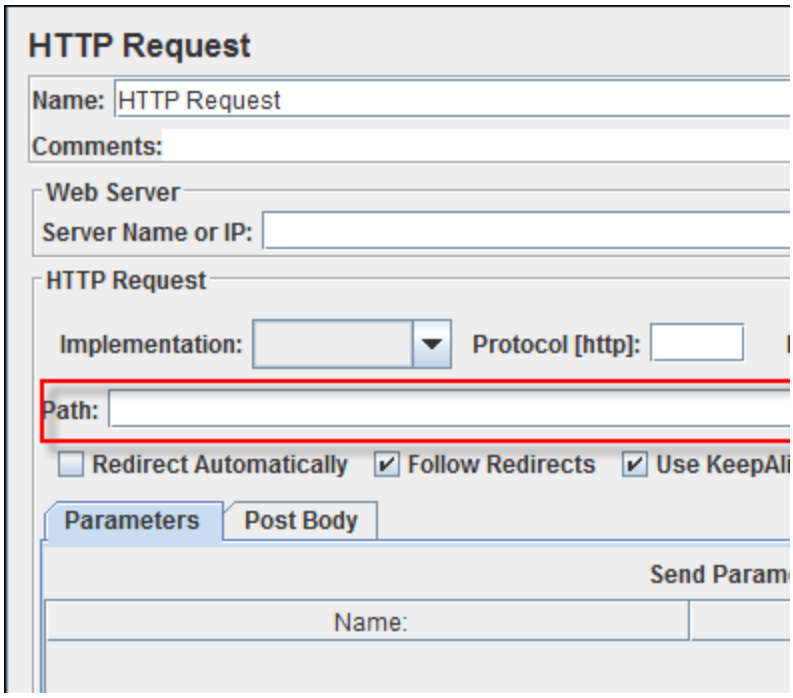
In the HTTP Request Defaults control panel, enter the Website name under test (http://www.google.com)

- **HTTP Request**

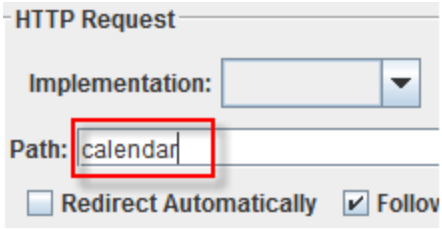
Right-click on Thread Group and select: **Add** -> **Sampler** -> **HTTP Request**.



In HTTP Request Control Panel, the Path field indicates which **URL request** you want to send to Google server.



For example, if you enter "*calendar*" in Path field. JMeter will create the URL request http://www.google.com/calendar to Google server



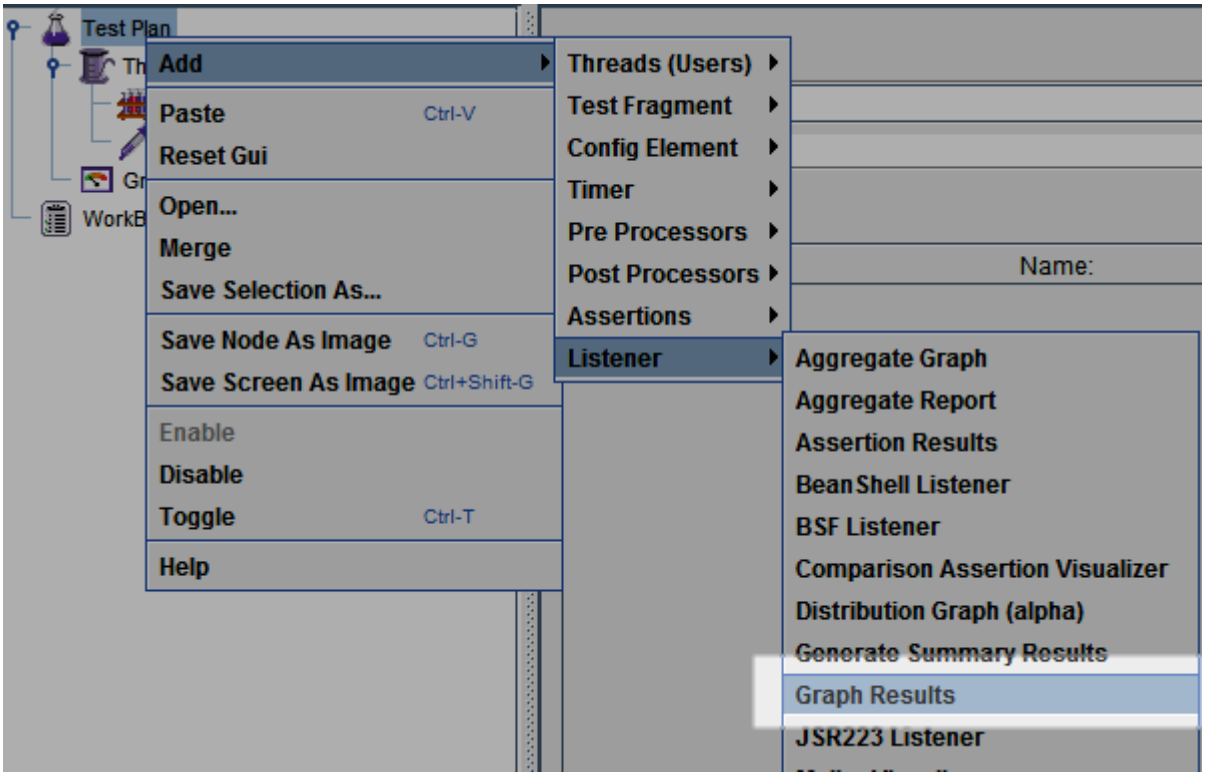
If you keep the Path field blank JMeter will create the URL request http://www.google.com to Google server.

In this test, you keep the Path field blank to make JMeter create the URL request http://www.google.com to Google server.

Step 3) Adding Graph result

JMeter can show the test result in Graph format.

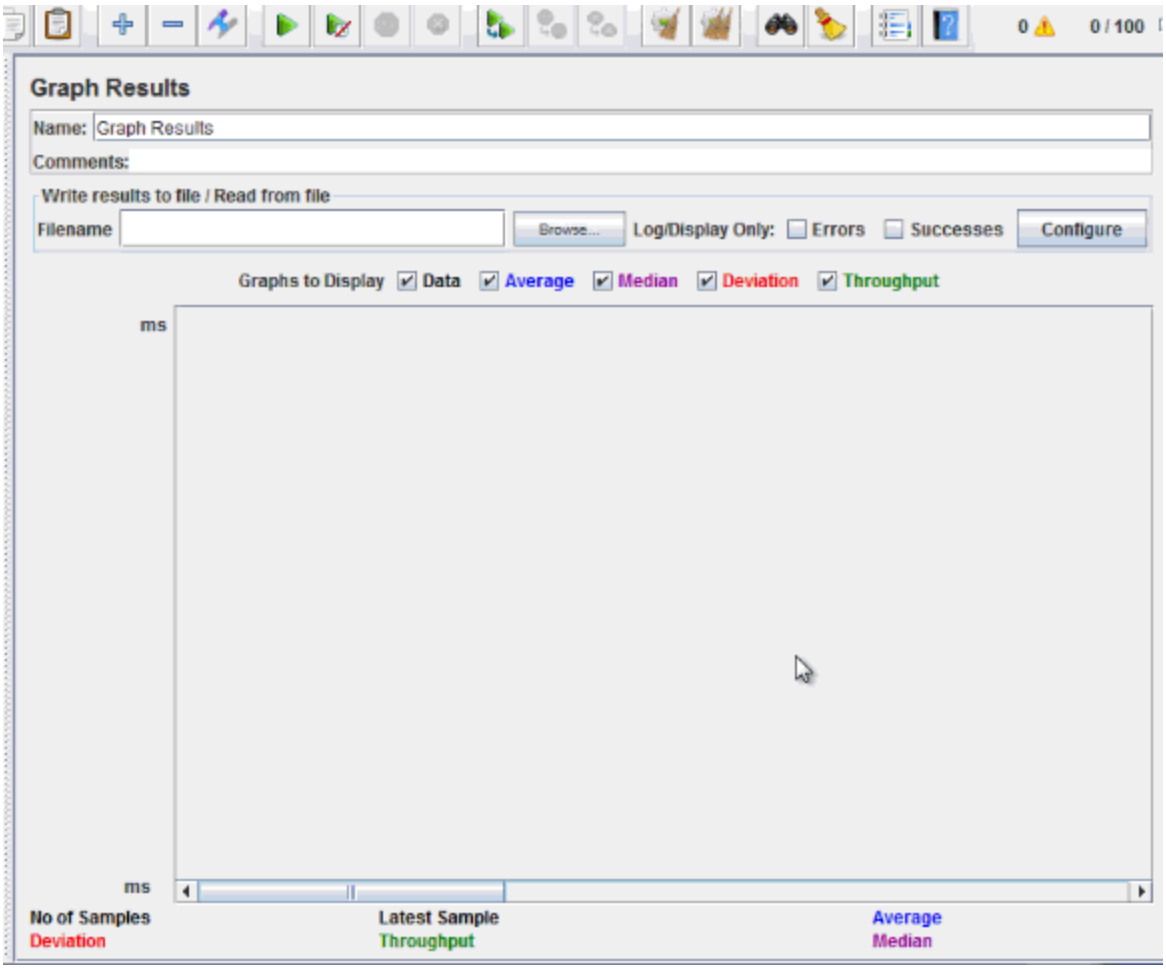
Right click Test Plan, **Add** -> **Listener** -> **Graph Results**



Step 4) Run Test and get the test result

Press **the Run** button (Ctrl + R) on the Toolbar to start the software testing process. You will see the test result display on Graph in the real time.

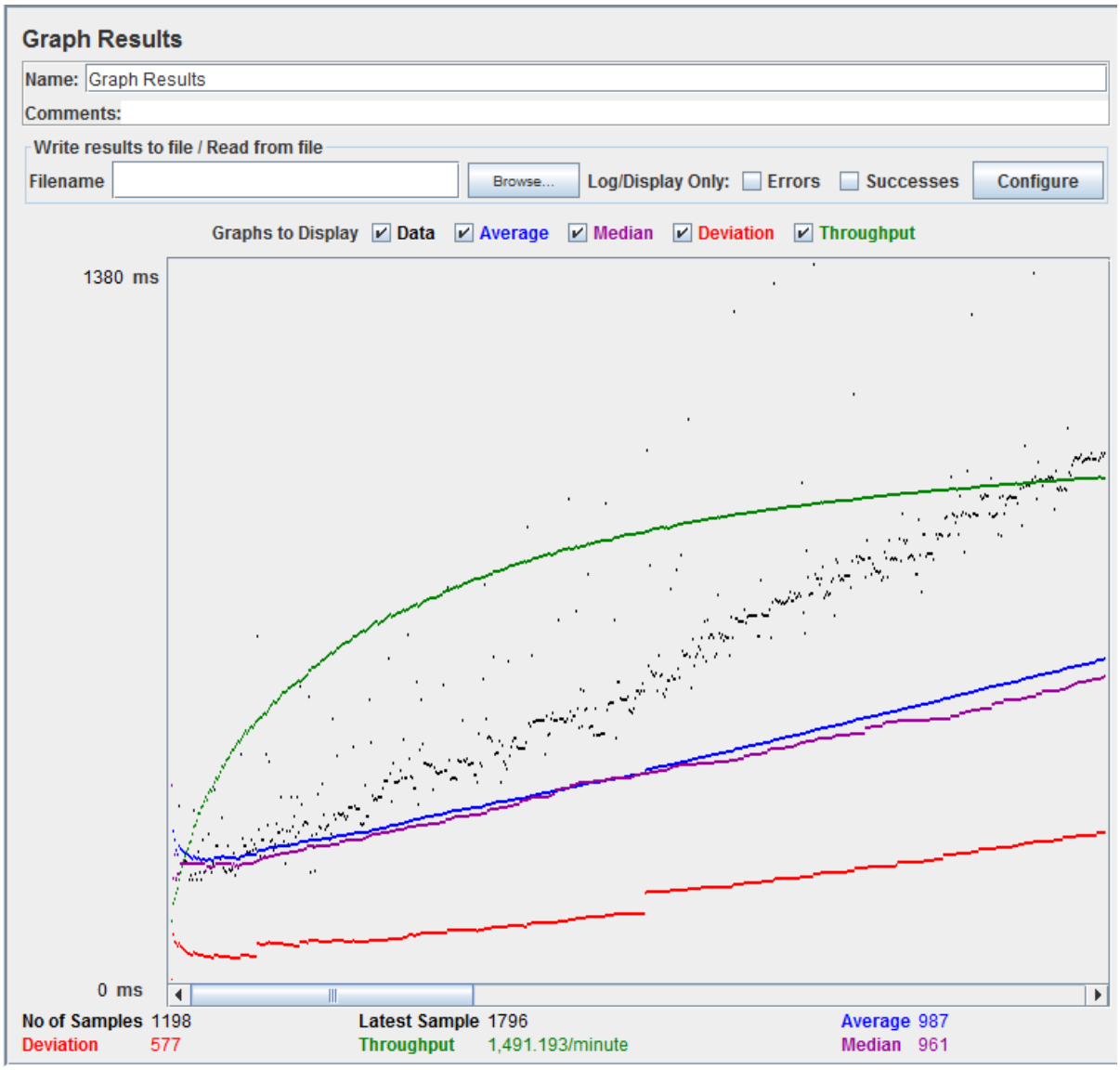
The picture below presents a graph of a test plan, where we simulated 100 users who accessed on website www.google.com.



At the bottom of the picture, there are the following statistics, represented in colors:

- Black: The total number of current samples sent.
- Blue: The current average of all samples sent.
- Red: The current standard deviation.
- Green: Throughput rate that represents the number of requests per minute the server handled

Let analyze the performance of Google server in below figure.



To analyze the performance of the web server under test, you should focus on 2 parameters

- Throughput
- Deviation

The **Throughput** is the most important parameter. It represents the ability of the server to handle a heavy load. The **higher** the Throughput is, the **better** is the server performance.

In this test, the throughput of Google server is 1,491.193/minute. It means Google server can handle 1,491.193 requests per minute. This value is quite high so we can conclude that Google server has good performance

The **deviation** is shown in red - it indicates the deviation from the average.
The **smaller** the **better**.