Geetanjali Institute of Technical Studies

(Approved by AICTE, New Delhi and Affiliated to Rajasthan Technical University Kota (Raj.))

DABOK, UDAIPUR, RAJASTHAN 313022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

B. Tech - VIII SEMESTER

तमसो मा ज्योतिर्गमय्

Darkness to Light

ACADEMIC YEAR – 2022-23

SOTWARE TESTING AND VALIDATION LAB (8CS4-22)

Submitted To
Mr. Jitendra Sharma
Ms. Ayushi Ghill

Submitted By

VISION & MISSION OF INSTITUTE

INSTITUTE VISION

To achieve excellence in technical and management education through quality teaching, research and innovation.

INSTITUTE MISSION

To provide a conducive environment in order to produce socially responsible and productive professionals.

VISION & MISSION OF DEPARTMENT

VISION

To nurture the students to become employable graduates who can provide solutions to the societal issues through ICT.

MISSION

To nurture knowledge of students in theoretical and practical aspects in collaboration with industries.

To inculcate the students towards research and innovation to fulfill the need of industry & society.

To develop socially responsible professionals with values and ethics.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

The Programme Educational Objectives of the programme offered by the department are listed below:

- **PEO1:** ANALYTICAL SKILLS
    1. To facilitate the graduates with the ability to visualize, gather information, articulate, analyze, solve complex problems, and make decisions. These are essential to address the challenges of complex and computation intensive problems increasing their productivity
- **PEO2:** – TECHNICAL SKILLS

  2. To facilitate the graduates with the technical skills that prepare them for immediate employment and pursue certification providing a deeper understanding of the technology in advanced areas of computer science and related fields.
- **PEO3:** SOFT SKILLS

  To facilitate the graduates with the soft skills that include fulfilling the mission, setting goals, showing self-confidence by communicating effectively, having a positive attitude, get involved in team-work, being a leader, managing their career and their life.

COURSE OUTCOMES (COs)

CO1	The students will learn the different programming paradigms about Jabuti.
CO2	The students will understand the principles and aspects of JABUTI.
CO3	The students will be able to understand and apply the principles of jabuti and Selenium.
CO4	The students will be able to understand the concept of core functionality about the selenium and installation aspects as well.
CO5	The students will learn various streams and they will be able todevelop programming skills to solve basic real world problems using Selenium techniques.

INDEX

S. No.	LIST OF EXPERIMENT	DATE	SIGN
1.	Write a program that calculates the area and perimeter of the circle. And find the Coverage & Test Cases of that program using JaButi Tool., ROI and time during the execution of the program.		
2.	Write a program which read the first name and last name from console and matching with expected result by using JaBuTi.		
3.	Write a program that takes three double numbers from the java console representing , respectively, the three coefficients a,b, and c of a quadratic equation.		
4.	Write a program that reads commercial website URL from a url from file .you should expect that the URL starts with www and ends with .com. retrieve the name of the site and output it. For instance, if the user inputs www.yahoo.com, you should output yahoo. After that find the test cases and coverage using JaBu**ti.**		
5.	Write a program that reads two words representing passwords from the java console and outputs the number of character in the smaller of the two. For example, if the words are open and sesame, then the output should be 4, the length of the shorter word, open. And test this program using JaButi		
6.	Calculate the mutation score of programs given in 1(a) to 1 (f) using jumble Tool		
7.	Calculate the coverage analysis of programs given in 1 (a) To 1 (f) using Eclemma Free open source Tool**.**		
8.	Using Selenium IDE, Write a test suite containing minimum 4 test cases.		
9.	Conduct a test suite for any two websites.		
10.	Install Selenium server and demonstrate it using a script in Java/PHP		
11.	Write and test a program to login a specific web page using selenium.		
12.	Write and test a program to update 10 student records into table into Excel file.		

EXPERIMENT NO. 1

AIM: Write a program that calculates the area and perimeter of the circle. And find the Coverage & Test Cases of that program using JaButi Tool.

Introduction:

In this java program, we will **read radius of a circle and find their area and perimeter**, this program will be implementing using class and objects. Here value will be reading and printing through class methods.

In this example we will read radius of a circle and then **calculate area, perimeter of a circle**. We will create a class to find the area and perimeter.

In this program we will use Math.PI to use value of PI.

PROGRAM:

```
/*Java program to create class to calculate area and perimeter of circle.*/
importjava.util.*;

classAreaOfCircle
{
	private float radius=0.0f;
	private float area=0.0f;
	private float perimeter=0.0f;

	//function to read radius
	public void readRadius()
	{
		//Scanner class - to read value from keyboard
		Scanner sc=new Scanner(System.in);
		System.out.print("Enter radius:");
		radius=sc.nextFloat(); //to read float value from keyboard
	}

	//funtction to calculate area
	//return value - will return calculated area
	public float getArea()
	{
		area= Math.PI *radius*radius;
		return area;
	}

	//funtction to calculate perimeter
	//return value - will return calculated perimeter
	public float getPerimeter()
	{
		perimeter = 2* Math.PI *radius;
		return perimeter;
	}
}
```

```
public  class circle
{
        public static void main(String []s)
        {
                AreaOfCircle area=new AreaOfCircle();

                area.readRadius();
                System.out.println("Area of circle:" + area.getArea());
                System.out.println("Perimeter of circle:" + area.getPerimeter());
        }
}
```

CODE OUTPUT:

```
Compile: javac circle.java
Run: java circle
```

```
Output:

Enter radius:15.50

Area of circle:754.385

Perimeter of circle:97.340004
```

EXPERIMENT NO. 2

AIM: Write a program which read the first name and last name from console and matching with expected result by using JaBuTi

PROGRAM:

```
import java.util.Scanner;

public class FullNameString {

 public static void main(String[] args) {

  Scanner input = new Scanner(System.in);

  System.out.print("Enter first name :: ");
  String firstName = input.nextLine();

  System.out.print("Enter middle name :: ");
  String middleName = input.nextLine();

  System.out.print("Enter surname :: ");
  String lastName = input.nextLine();

  // Here StringBuffer is used to combine 3 strings into single
  StringBuffer fullName = new StringBuffer();
  fullName.append(firstName);
  fullName.append(" ");   // For space between names
  fullName.append(middleName);
  fullName.append(" ");   // For space between names
  fullName.append(lastName);
  System.out.println("Hello, " + fullName);
 }
}
```

CODE OUTPUT:

```
C:\>javac FullNameString.java
C:\>java FullNameString
Enter first name :: Rahul
Enter middle name :: S.
Enter surname :: Tamkhane
Hello, Rahul S. Tamkhane
          }
```

EXPERIMENT NO. 3

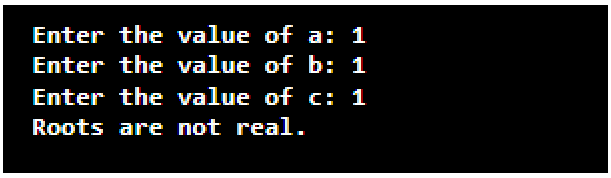
AIM: Write a program that takes three double numbers from the java console representing , respectively, the three coefficients a,b, and c of a quadratic equation

PROGRAM:

QuadraticEquationExample1.java

```
import java.util.Scanner;
public class QuadraticEquationExample1
{
public static void main(String[] Strings)
{
Scanner input = new Scanner(System.in);
System.out.print("Enter the value of a: ");
double a = input.nextDouble();
System.out.print("Enter the value of b: ");
double b = input.nextDouble();
System.out.print("Enter the value of c: ");
double c = input.nextDouble();
double d= b * b - 4.0 * a * c;
if (d> 0.0)
{
double r1 = (-b + Math.pow(d, 0.5)) / (2.0 * a);
double r2 = (-b - Math.pow(d, 0.5)) / (2.0 * a);
System.out.println("The roots are " + r1 + " and " + r2);
}
else if (d == 0.0)
{
double r1 = -b / (2.0 * a);
System.out.println("The root is " + r1);
}
else
{
System.out.println("Roots are not real.");
}
}
}
```

OUTPUT



```
Enter the value of a: 1
Enter the value of b: 1
Enter the value of c: 1
Roots are not real.
```

EXPERIMENT NO. 4

AIM: Write a program that reads commercial website URL from a url from file .you should expect that the URL starts with www and ends with .com. retrieve the name of the site and output it. For instance, if the user inputs www.yahoo.com, you should output yahoo. After that find the test cases and coverage using JaButi**.**

PROGRAM:

```
// Java program t demonstrate working of URL
importjava.net.MalformedURLException;
import java.net.URL;

public class URLclass1
{
    public static void main(String[] args)
                            throwsMalformedURLException
    {

            // creates a URL with string representation.
            URL url1 =
            new URL("https://www.google.co.in/?gfe_rd=cr&ei=ptYq" +
                            "WK26I4fT8gfth6CACg#q=geeks+for+geeks+java");

            // creates a URL with a protocol,hostname,and path
            URL url2 = new URL("http", "www.geeksforgeeks.org",
                                            "/jvm-works-jvm-architecture/");

            URL url3 = new URL("https://www.google.co.in/search?"+
                                            "q=gnu&rlz=1C1CHZL_enIN71" +
                                            "4IN715&oq=gnu&aqs=chrome..69i57j6" +
                                            "9i60l5.653j0j7&sourceid=chrome&ie=UTF" +
                                            "-8#q=geeks+for+geeks+java");

            // print the string representation of the URL.
            System.out.println(url1.toString());
            System.out.println(url2.toString());
            System.out.println();
            System.out.println("Different components of the URL3-");

            // retrieve the protocol for the URL
            System.out.println("Protocol:- " + url3.getProtocol());

            // retrieve the hostname of the url
            System.out.println("Hostname:- " + url3.getHost());

            // retrieve the default port
            System.out.println("Default port:- " +
                                                                        url3.getDefaultPort());

            // retrieve the query part of URL
            System.out.println("Query:- " + url3.getQuery());
```

```
                // retrieve the path of URL
                System.out.println("Path:- " + url3.getPath());

                // retrieve the file name
                System.out.println("File:- " + url3.getFile());

                // retrieve the reference
                System.out.println("Reference:- " + url3.getRef());
        }
    }
```

OUTPUT:

```
https://www.google.co.in/?gfe_rd=cr&ei=ptYqWK26I4fT8gfth6CACg#q=geeks+for+geeks+java
https://www.geeksforgeeks.org/jvm-works-jvm-architecture/

Different components of the URL3-
Protocol:- https
Hostname:- www.google.co.in
Default port:- 443
Query:-
q=gnu&rlz=1C1CHZL_enIN714IN715&oq=gnu&aqs=chrome..69i57j69i60l5.653j0j7&sourceid=
chrome&ie=UTF-8
Path:- /search
File:-
/search?q=gnu&rlz=1C1CHZL_enIN714IN715&oq=gnu&aqs=chrome..69i57j69i60l5.653j0j7&s
ourceid=chrome&ie=UTF-8
```

EXPERIMENT NO. 5

Aim: Write a program that reads two words representing passwords from the java console and outputs the number of character in the smaller of the two. For example, if the words are open and sesame, then the output should be 4, the length of the shorter word, open. And test this program using JaButi.

PROGRAM:

```
 import java.util.Scanner;
class Main
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);

        String strs[] = sc.nextLine().split(" ");
        int max_Length = 0;
        int indexL = 0;
        int max_Frequency = 0;
        int indexF = 0;
        System.out.println("Input a text in a line:");
        for (int i = 0; i < strs.length; i++)
        {
            if (max_Length < strs[i].length())
            {
                indexL = i;
                max_Length = strs[i].length();
            }
            int ctr = 0;
            for (int j = i; j < strs.length; j++)
            {
                if (strs[i].equals(strs[j]))
                {
                    ctr++;
                }
            }
            if (max_Frequency < ctr)
            {
                indexF = i;
                max_Frequency = ctr;
            }
        }
                        System.out.println("Most frequent text and the word which has the maximum number
of letters:");
        System.out.println(strs[indexF] + " " + strs[indexL]);
    }
}
```

OUTPUT:

```
Thank you for your comment and your participation.
Input a text in a line:
Most frequent text and the word which has the maximum number of letters:
your participation.
```

EXPERIMENT 6

Aim:Calculate the mutation score of programs given in 1(a) to 1 (f) using jumble Tool

PROGRAM: int index = 0;

```
while(true)

{

index++;

if (index == 10)

break;

}
```

Mutated Code in JAVA

```
int index = 0;

while (true)

{

index++;

if (index >= 10)
break;
}
```

Explanation: The mutant code above will **pass the Jumble test** because change in == to >= does not affect the output of the code. Execution will stop when index == 10 and since we are increasing value by 1 and index starts from 0 so the output will remain same.

Example with Jumble

The code written below has been tested with Jumble plugin in Eclipse. The code detects the first occurrence of a duplicate and returns the value to the calling function. The program is flawed in many ways which you can try figuring out.

```
// Java program to illustrate mutation Testing
// The code detects the first occurrence of a
// duplicate and returns the value to the calling
// function
packagetestPackage;

            importjava.util.Arrays;
            importjava.util.List;
            publicclassSampProg
            {
               protectedintrepeatedNumber(finalList a)
               {
                  intlen = a.size(),i,dup = -1;
                  int[] arr = newint[len];
                  for(i=0; i<len; i++)
                  {
                     arr[i] = a.get(i);
                  }
```

```
        Arrays.sort(arr);
        try
        {
            for(i=1; i<len; i++)
            {
                if(arr[i] == arr[i-1])
                {
                    dup = arr[i];
                    break;
                }
            }
        }
        catch(Exception e)
        {
            System.out.println(e.getMessage());
        }
        returndup;
    }
}
```

After writing the program we create test cases using JUnit in Java. The output obtained on performing Jumble Analysis is given below:

```
Mutating testPackage.SampProg

Tests: testPackage.SampProgTest

Mutation points = 11, unit test time limit 2.94s

M FAIL: (testPackage.SampProg.java:8): -1 -> 1

M FAIL: (testPackage.SampProg.java:10): 0 -> 1

.M FAIL: (testPackage.SampProg.java:10): negated conditional

M FAIL: (testPackage.SampProg.java:16): 1 -> 0

M FAIL: (testPackage.SampProg.java:18): 1 -> 0

M FAIL: (testPackage.SampProg.java:18): - -> +

M FAIL: (testPackage.SampProg.java:18): negated conditional

M FAIL: (testPackage.SampProg.java:16): += -> -=

M FAIL: (testPackage.SampProg.java:16): negated conditional

.Jumbling took 7.595s

Score: 18%
```

EXPERIMENT 7

AIM: Calculate the coverage analysis of programs given in 1 (a) to 1 (f) using Eclemma Free open source Tool.

PROGRAM:

EclEmma is a free Java code coverage tool for Eclipse, available under the Eclipse Public License. It brings code coverage analysis directly into the Eclipse workbench:

- **Fast develop/test cycle:** Launches from within the workbench like JUnit test runs can directly be analyzed for code coverage.
- **Rich coverage analysis:** Coverage results are immediately summarized and highlighted in the Java source code editors.
- **Non-invasive:** EclEmma does not require modifying your projects or performing any other setup.

Since version 2.0 EclEmma is based on the JaCoCo code coverage library. The Eclipse integration has its focus on supporting the individual developer in an highly interactive way. For automated builds please refer to JaCoCo documentation for integrations with other tools.

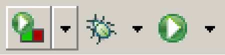
Originally EclEmma was inspired by and technically based on the great EMMA library developed by VladRoubtsov.

The update site for EclEmma is **http://update.eclemma.org/**. EclEmma is also available via the Eclipse Marketplace Client, simply search for "EclEmma".

Launching

EclEmma adds a so called *launch mode* to the Eclipse workbench. It is called *Coverage* mode and works exactly like the existing *Run* and *Debug* modes. The *Coverage* launch mode can be activated from the *Run* menu or the workbench's toolbar:

EclEmma adds a so called *launch mode* to the Eclipse workbench. It is called *Coverage* mode and Works exactly like the existing *Run* and *Debug* modes. The *Coverage* launch mode can be activated from the *Run* menu or the workbench's toolbar:



Simply launch your applications or unit tests in the *Coverage* mode to collect coverage information.

Currently the following launch types are supported:

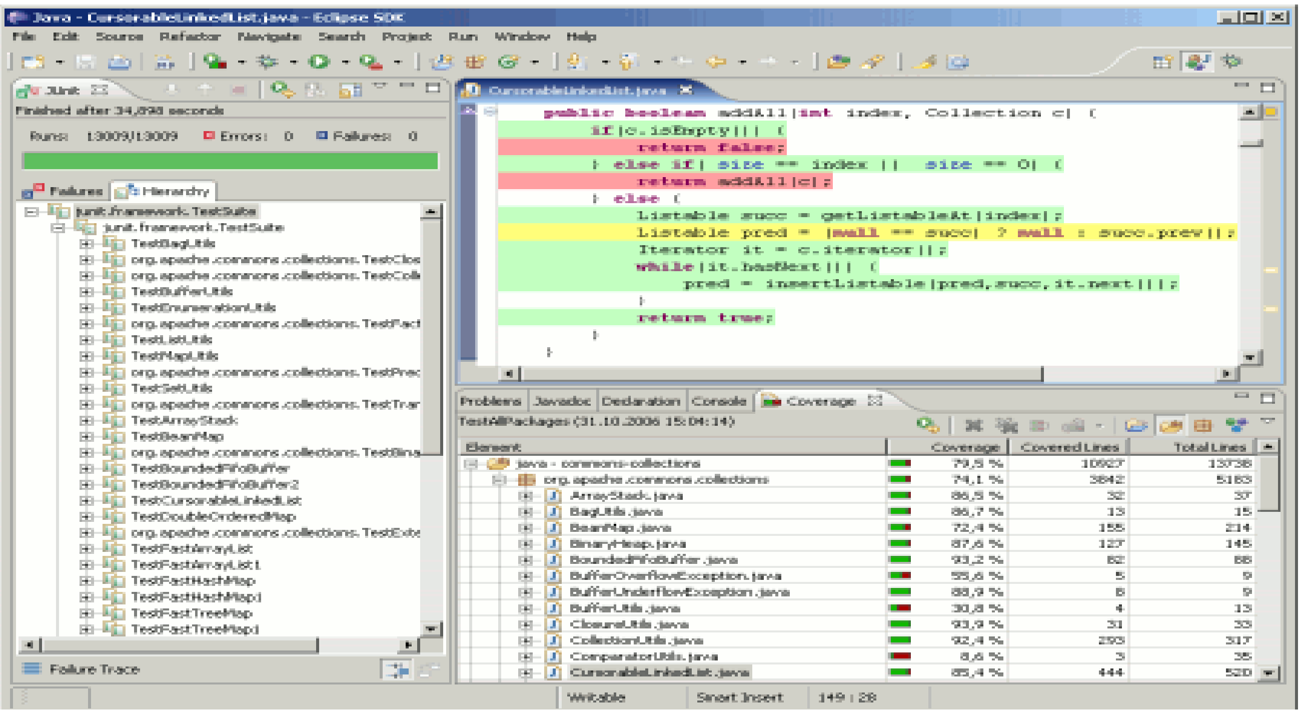
- Local Java application
- Eclipse/RCP application
- Equinox OSGi framework
- JUnit test
- TestNG test
- JUnit plug-in test
- JUnit RAP test
- SWTBot test
- Scala application

Analysis

On request or after your target application has terminated code coverage information is automatically available in the

Eclipse workbench:

- **Coverage overview:** The *Coverage* view lists coverage summaries for your Java projects,
- allowing
- drill-down to method level.
- **Source highlighting:** The result of a coverage session is also directly visible in the Java source editors.
- A customizable color code highlights fully, partly and not covered lines.
- This works for your own source code as well as for source attached to instrumented external libraries.
- Additional features support analysis for your test coverage:
- Different counters: Select whether instructions, branches, lines, methods, types
- orcyclomatic complexity should be summarized.
- Multiple coverage sessions: Switching between coverage data from multiple sessions is possible.
- Merge Sessions: If multiple different test runs should be considered for analysis coverage
- sessions can easily be merged.
- Import/Export
- While EclEmma is primarily designed for test runs and analysis within the Eclipse workbench, it provides some import/export features.
- Execution data import: A wizard allows to import JaCoCo *.exec execution data files from external launches.
- Coverage report export: Coverage data can be exported in HTML, XML or CSV format or asJaCoCo execution data files (*.exec).


Java - CursorableLinkedList.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
JUnit
Failures
Hierarchy
Failure Trace
CursorableLinkedList.java
Problems Javadoc Declaration Console Coverage
Writable
Smart Insert
149 : 28

Finished after 34,898 seconds

Runs: 13009/13009 Errors: 0 Failures: 0

```
junit.framework.TestSuite
  junit.framework.TestSuite
    TestBagUtils
    org.apache.commons.collections.TestClos
    org.apache.commons.collections.TestColl
    TestBufferUtils
    TestEnumerationUtils
    org.apache.commons.collections.TestFact
    TestListUtils
    TestMapUtils
    org.apache.commons.collections.TestPred
    TestSetUtils
    org.apache.commons.collections.TestTrar
    TestArrayStack
    TestBeanMap
    org.apache.commons.collections.TestBina
    TestBoundedFifoBuffer
    TestBoundedFifoBuffer2
    TestCursorableLinkedList
    TestDoubleOrderedMap
    org.apache.commons.collections.TestExte
    TestFastArrayList
    TestFastArrayList1
    TestFastHashMap
    TestFastHashMap1
    TestFastTreeMap
    TestFastTreeMap1
```

```java
    public boolean addAll(int index, Collection c) {
        if(c.isEmpty()) {
            return false;
        } else if( size == index ||  size == 0) {
            return addAll(c);
        } else {
            Listable succ = getListableAt(index);
            Listable pred = (null == succ) ? null : succ.prev();
            Iterator it = c.iterator();
            while(it.hasNext()) {
                pred = insertListable(pred,succ,it.next());
            }
            return true;
        }
    }
```

TestAllPackages (31.10.2006 15:04:14)

Element	Coverage	Covered Lines	Total Lines
java - commons-collections	79,5 %	10927	13738
org.apache.commons.collections	74,1 %	3842	5183
ArrayStack.java	86,5 %	32	37
BagUtils.java	86,7 %	13	15
BeanMap.java	72,4 %	155	214
BinaryHeap.java	87,6 %	127	145
BoundedFifoBuffer.java	93,2 %	82	88
BufferOverflowException.java	55,6 %	5	9
BufferUnderflowException.java	88,9 %	8	9
BufferUtils.java	30,8 %	4	13
ClosureUtils.java	93,9 %	31	33
CollectionUtils.java	92,4 %	293	317
ComparatorUtils.java	8,6 %	3	35
CursorableLinkedList.java	85,4 %	444	520

EXPERIMENT 8

AIM: UsingSeleniumIDE,Write a testsuite containing minimum4test cases.

TC‟S #1: Manual Steps:

- Open (Example : Typewww.google.com)
- Type“energyefficient”intheGoogleSearchInputBox
- Click outside on an emptyspot
- Click SearchButton
- Verify the Text Present as “energyefficient”
- AsserttheTitleas“energyefficient-GoogleSearch”
- Savethetestcasewith.HTMLExtension.

TC‟S#2:

- Open (Example : Typewww.google.com)
- Type“SeleniumRC”intheGoogleSearchInputBox
- Click outside on an emptyspot
- Click SearchButton
- Verify the Text Present as “SeleniumRC”
- AsserttheTitleas“SeleniumRC-GoogleSearch”
- Savethetestcasewith.HTMLExtension.

Stepsforcreatingtestsuite:

1. CreatemoreTc‟ssaveeachTestCasewith<.html>extension.

2. OpenFirefox

3. Open Tools □ SeleniumIDE

4. File□Open□newTestSuite

5. File□Open□AddTestcases

6. Add more testcases

7. Save Suite with <.Html>extensions.

8. Run the testsuite.

EXPERIMENT 9

AIM:Conductatestsuitefornaytwowebsites.

TC‟S #1: ManualSteps:

- Open (Example : Typewww.google.com)
- Type“energyefficient”intheGoogleSearchInputBox
- Click outside on an emptyspot
- Click SearchButton
- Verify the Text Present as “energyefficient”
- AsserttheTitleas“energyefficient-GoogleSearch”
- Savethetestcasewith.HTMLExtension.

Steps for creating test suite:

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

2. OpenFirefox

3. Open Tools □ SeleniumIDE

4. File□Open□newTestSuite

5. File□Open□AddTestcases

6. Add more testcases

7. Save Suite with <.Html>extensions.

8. Run the testsuite

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:Verifythebrowserstitlehasthevalue"energyefficient-Yahoo!Search Results"
8. Save the test case with .HTML Extension.

EXPERIMENT 10

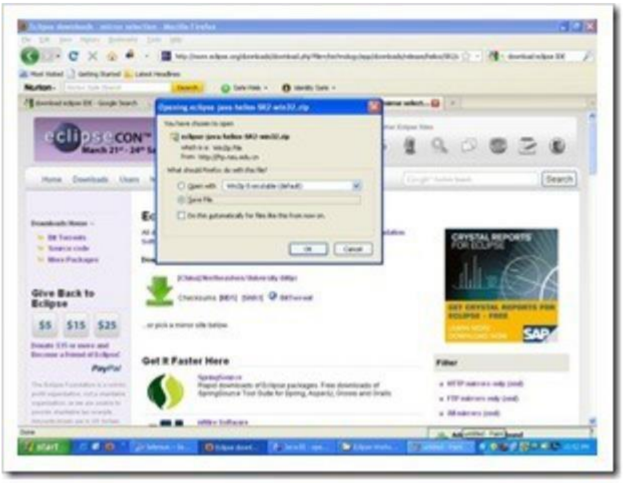
Aim:Install Selenium server and demonstrate it using a script in Java/PHP

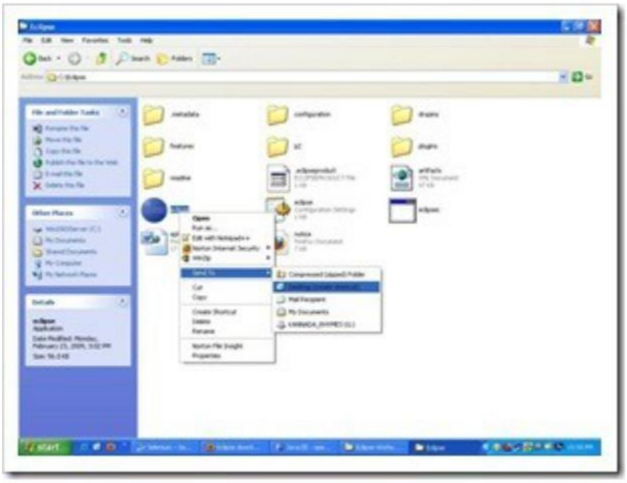
PROGRAM:

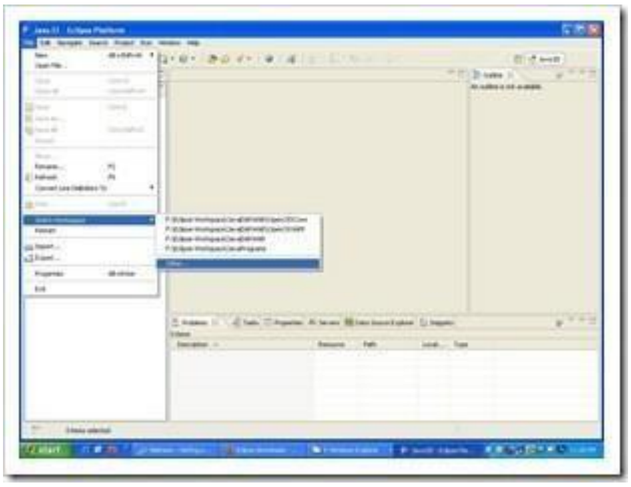
InstallationofSeleniumRCandEclipse

DownloadEclipse

1. Go to URL –http://www.eclipse.org/downloads/
2. SelectEclipseIDEforJavaDevelopers(ClickonWindows32bitplatform)
3. ClickonOKbuttonandsavetoalocaldrive(i.e.C:orD:,etc)
4. Unzipthedownloadedzipfileandrenamethatto**Eclipse**
5. Create one more folder “Eclipse-Workspace” (i.e. C:Eclipse-Workspace)inthesamedrivewhereEclipseisunzippedandrenamed.
6. Create Eclipse desktop shortcut (go to C:Eclipse folder –> right click Eclipse.exeandthenclickon“desktopcreateshortcut”)asdemonstrated in the belowpictures.





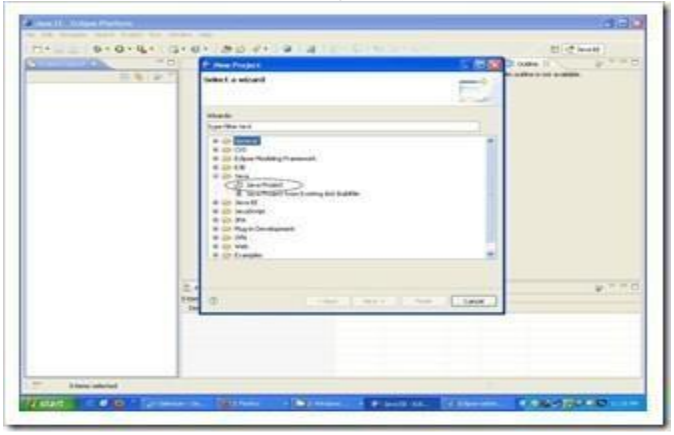


Wehavefinishedsettinguptheeclipse.Now,weneedtodownloadSeleniumRC server/clientdriverandconfigurethattoEclipse

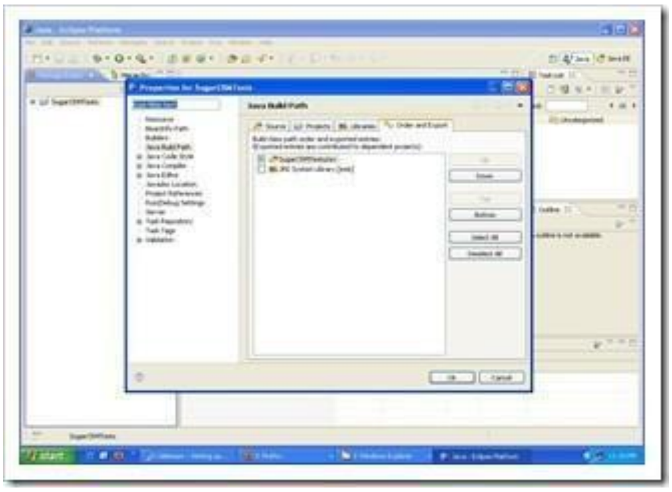
1. Download Selenium server:http://seleniumhq.org/download/
2. Download Selenium Client driver for Java (from Selenium ClientDrivers section)
3. Create“Selenium”folderinC:driveandcopytheSelenium-server.jar aswellasunziptheSeleniumClientdriver(C:Selenium)

Downloadingandunzippingthefilesintoafolderisdone.Weneedtoconfigure theappropriateSeleniumClientdriverJarfiletotheEclipse.

1. GotoEclipse–>ClickFile–>New–>Project(fromvariousoptionsneedto select just“project”)
   InSelectWizard–>ClickJava–>“JavaProject”(demonstratedinthe belowfigure)



1. Give the project name (e.g.SugarCRMTests)
2. Click Finish – ClickYes
3. Nowwearedonewithcreationofprojectandneedtoconfigure the Selenium Client driver to thisProject
4. RightClick“SugarCRMTests”project



5. Click“JavaBuildPath”
6. Click Librariestab
7. Click “Add External JARs”button
8. Select “Selenium Client Drivers” unzipped in C:Selenium folder (SeleniumServerJARfileshouldnotbeadded)
9. ClickOK

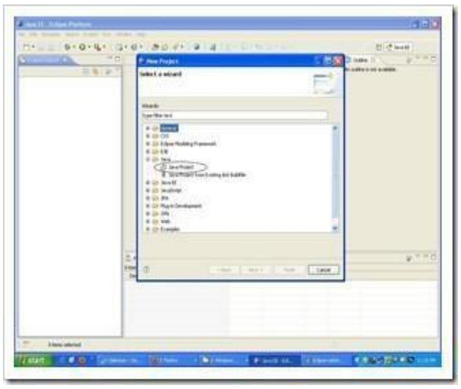
Referencedlibraries–>containsboththeSeleniumClientdriverjarfiles as shown in the belowpicture
Wehavefinishedsettinguptheeclipse.Now,weneedtodownloadSeleniumRC server/clientdriverandconfigurethattoEclipse

4. Download Selenium server:http://seleniumhq.org/download/
5. Download Selenium Client driver for Java (from Selenium ClientDrivers section)
6. Create“Selenium”folderinC:driveandcopytheSelenium-server.jar aswellasunziptheSeleniumClientdriver(C:Selenium)

Downloadingandunzippingthefilesintoafolderisdone.Weneedtoconfigure theappropriateSeleniumClientdriverJarfiletotheEclipse.

2. GotoEclipse–>ClickFile–>New–>Project(fromvariousoptionsneedto select just“project”)
3. InSelectWizard–>ClickJava–>“JavaProject”(demonstratedinthe belowfigure)

OUTPUT:



EXPERIMENT 11

AIM: Write and test a program to login a specific web page.

PROGRAM :

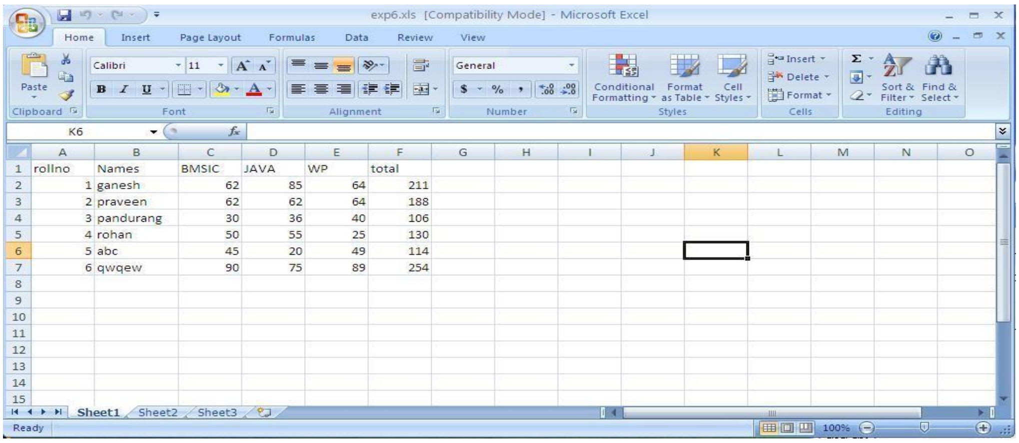
```
importcom.thoughtworks.selenium.*;
import org.junit.After;
importorg.junit.Before;
import org.junit.Test;
importjava.util.regex.Pattern;

public class exp5 extends SeleneseTestCase
        { @Before
        public void setUp() throws Exception {
                selenium=newDefaultSelenium("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();
        }
                                                    }
```

EXPERIMENT 12

AIM:Write and test a program to update 10 student records into table into Excel file .

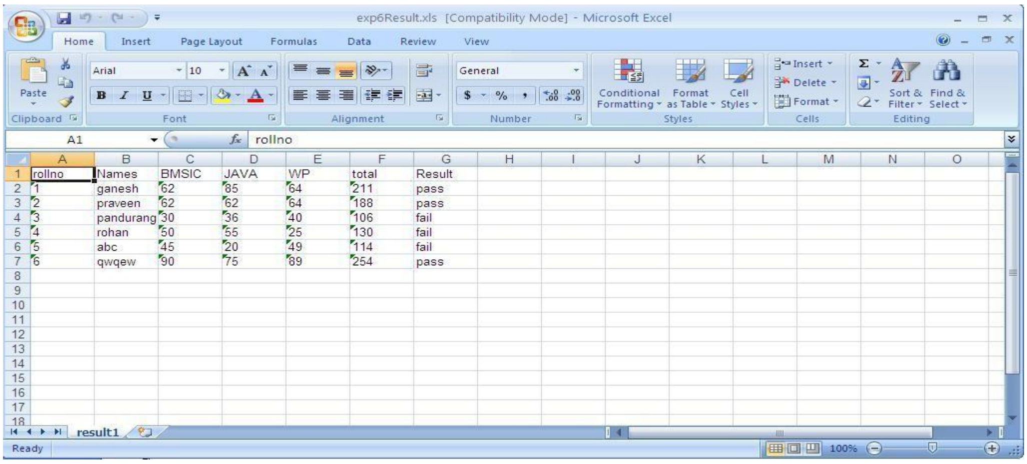
```
importjava.io.FileInputStream;
import java.io.FileOutputStream;
import jxl.Sheet;
importjxl.Workbook;
import jxl.write.Label;
importjxl.write.WritableSheet; import
jxl.write.WritableWorkbook; import
org.testng.annotations.*; public class
newj {

@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()];
FileOutputStreamfo=newFileOutputStream("D://exp6Result.xls");
WritableWorkbookwwb = Workbook.createWorkbook(fo);
WritableSheetws=wwb.createSheet("result1",0);for(inti=0;i<s.getRows();i++)
for(intj = 0; j <s.getColumns(); j++)
{

        a[i][j] = s.getCell(j, i).getContents();
                        Labell2=newLabel(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(intj=2;j<s.getColumns();j++)
        {
                a[i][j] = s.getCell(j, i).getContents();
                intx=Integer.parseInt(a[i][j]);
        if(x > 35)
        {
        Labell1=newLabel(6,i,"pass");
        ws.addCell(l1);
        }
        else
        {
                Labell1=newLabel(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



A1 rollno

	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

RUBRICS EVALUATION

Performance Criteria	Scale 1 (0-25%)	Scale 2 (26-50%)	Scale 3 (51-75%)	Scale 4 (76-100%)	Score (Numerical)
Understandability Ability to analyse Problem and Identify solution	Unable to understand the problem.	Able to understand the problem partially and unable to identify the solution	Able to understand the problem completely but unable to identify the solution	Able to understand the problem completelyand able to providealternative solution too.	
Logic Ability to specify Conditions & control flow that are appropriate for theproblem domain.	Program logic is incorrect	Program logic is on the right track but has several errors	Program logic is mostly correct, but may containan occasional boundary error or redundant orcontradictory condition.	Program logic is correct, with no known boundary errors, and no redundant or contradictory conditions.	
Debugging Ability to execute /debug	Unable to execute program	Unable to debug several errors.	Able to execute program with several warnings.	Able to execute program completely	
Correctness Ability to code formulae and algorithms that reliably produce correct answers or appropriate results.	Program does not produce correct answers or appropriate results for most inputs.	Program approaches correct answers or appropriate results for most inputs, but can contain miscalculations in some cases.	Program produces correct answers or appropriate results for most inputs.	Program produces correct answers or appropriate results for all inputs tested.	
Completeness Ability to demonstrate and deliver on time.	Unable to explain the code and the code was overdue.	Unable to explain the code and the code submission was late.	Able to explain code and the program was delivered within the due date.	Able to explain code and the program was delivered on time.	
				TOTAL	