K.S.R COLLEGE OF ENGINEERING (AUTONOMOUS)

THOKKAVADI POST, TIRUCHENGODE-637215

NAMAKKAL DISTRICT, TAMILNADU

RECORD NOTE BOOK

REGISTERNO:

Certified that this is the bonafide record of work done by

Selvan / Selvi ____________________ of the ______ Semester

____branch during the

year____________in the ____________________laboratory.

Staff-In-Charge   Head of The Department

Submitted for the University practical

Examination on_____________________________________

Internal examiner   External Examiner

LIST OF EXPERIMENTS

Ex. No	Date	Experiment Name	Page No	Marks allocated
1		STUDY OF THE AUTOMATION TESTING APPROACH	3	
2		WRITE A TEST SUITE CONTAINING MINIMUM 4 TEST CASES USE	8	
3		CONDUCT A TEST SUITE FOR ANY TWO WEB SITES	11	
4		DEVELOP A WEB PAGE WHICH CALCULATES THE GCD OF 2 NUMBERS. WRITE A SCRIPT FOR TESTING IT.	15	
5		DEVELOP AND TEST A PROGRAM TO LOGIN A WEB	17	
6		WRITE AND TEST A PROGRAM TO UPDATE STUDENT RECORDS INTO EXCEL FILE IN TABLE FORMAT	24	
7		WRITE AND TEST A PROGRAM TO COUNT NUMBER OF ITEMS PRESENT ON A DESKTOP	31	
8		WRITE AND TEST A PROGRAM TO GET THE NUMBER OF LIST ITEMS SELECTED IN A LIST / COMBO BOX	33	

Ex. No: 1	STUDY OF THE AUTOMATION TESTING APPROACH
Date:	

Aim:

To study and analyse the Automation testing approach.

Automation Testing:

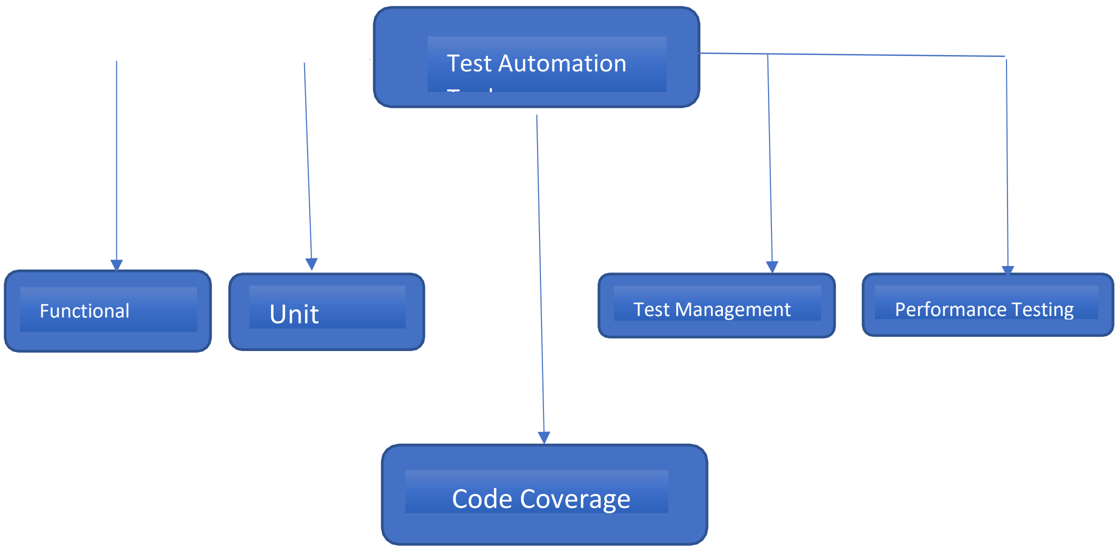
Automated testing is a process using software separate from the software under test to control the execution of tests and the comparison of actual outcomes with the expected outcomes. Automation tools are used to automate certain sections of manual testing but not all. Automated testing generally saves time, the tester can efficiently run a large number of tests in a short period and so important and repetitive tasks, as well as testing that would be difficult to do manually, can be automated. Besides saving time, automation testing saves money and effort, increases the quality of the testing tasks and also helps in improving software accuracy. Test Automation requires a skilled tester with knowledge of the automation tools and the software being tested to set up the test cases and perform the testing.

Advantage	**Disadvantage**
Improves accuracy and quick finding of bugs compared to manual testing	Choosing the right tool requires considerable effort, time, and evolution plan.
Saves time and effort by making testing more efficient	Requires knowledge of the testing tool.
Increases test coverage because multiple testing tools can be used at once allowing for parallel testing of different test scenarios	Cost of buying the testing tool and, in the case of playback methods, test maintenance is a bit expensive

Automation test script is repeatable	Proficiency is required to write the automation test scripts.

Automation Tools Categories:

Software testing automation tools can be divided into different categories as follows: Unit Testing Tools, Functional Testing Tools, Code Coverage Tools, Test Management Tools, and Performance Testing Tools.



TEST AUTOMATION FRAMEWORK

A test automation framework is a defined, extensible support structure within which the test automation suite is developed and implemented. Automation Framework comprises of a combination of tools and practices that are designed to help in executing software testing more efficiently. It includes the physical structures used for test creation and implementation as well as the logical interaction between components such as coding standards, test-data handling methods, object repositories, processes for storing test results, or information on how to access external resources. A framework "facilitates a standard way for modifying, adding, and deleting the

[test] scripts and functions," and provides "scalability and reliability withless effort.

Importance of Test Automation Framework

While testers can still script or record tests using an automation tool, however, integrating it with an organized framework typically provides additionalbenefits. A well-defined test automation framework will help the testing team in; achieving higher reusability of test components, developing scripts that are easily maintainable and, obtaining high-quality test automation scripts. Utilizing a framework for automated testing will also increase test speed and efficiency, improvetest accuracy, and reduce test maintenance costs as well as lower risks. Test automation frameworks provide support foundation to a variety of automated software tests including: Unit testing, Functional testing, Performance testing.

Types of Test Automation Framework

- Linear Automation Framework.
- Modular Based Framework
- Library Architecture Framework
- Data-Driven Framework
- Keyword-Driven Framework
- Hybrid Testing Framework

AUTOMATION TOOLS

An automation tool is a software itself with the help of which the actual software in focus can be tested, in other words, the automation tool help and serves as a means in doing software testing. The rapid and unparalleled change in technology affects how organizations develop, validate, deliver, and operate software products. Meeting the demands of today's software quality standard requires testing the software, and the success of the testing project is largely determined by testing technique and automation tool used. There are various testing automation tools each having its strengths and weaknesses and serving for a different purpose.

Junit

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and it allows the developer to write an oracle for each test case, and to automatically execute test sets

Selenium

Selenium is a framework for testing web applications that is compatible with various browsers and platforms like Windows, Mac, and Linux. Selenium helps the testers to write tests in various programming languages like Java, PHP, C#, Python, Groovy, Ruby, and Perl

Unified Functional Testing (UFT)

UTF, formerly Quick Test Professional (QTP), is a test automation tool for functional and regression testing, it's probably the most popular commercial tool for functional test automation. UFT offers a comprehensive set of features that can cover most of functional automated testing needs on desktop, mobile and Web platforms

Katalon Studio

Katalon Studio is an automated testing platform that offers a comprehensive set of features to implement full automated testing solutions for Web, API, and Desktop and Mobile applications. Built on top of the open-source Selenium and Appium frameworks, Katalon Studio allows teams to get started with test automation quickly by reducing the effort and expertise required for learning and integrating theseframeworks for automated testing needs.

Result

Hence, the study of Automation software testing approach is successfully completed.

Ex. No: 2	WRITE A TEST SUITE CONTAINING MINIMUM 4 TEST CASES USE
Date:	

Aim:

To write a test suite containing minimum 4 test cases.

Test Cases:

Step 1) A simple test case to explain the scenario would be

Test Case #	Test Case Description
1	Check response when valid email and password is entered

Step 2) In order to execute the test case, you would need Test Data. Adding it below

Test Case #	Test Case Description	Test Data
1	Check response when valid email and password is entered	Email: guru99@email.com Password: lNf9^Oti7^2h

Identifying test data can be time-consuming and may sometimes require creatingtest data afresh. The reason it needs to be documented.

Step 3) In order to execute a test case, a tester needs to perform a specific set ofactions on the AUT. This is documented as below:

Test Case #	Test Case Description	Test Steps	Test Data
1	Check response when valid email and password is entered	1) Enter Email Address 2) Enter Password 3) Click Sign in	Email: guru99@email.com Password: lNf9^Oti7^2h

Many times, the Test Steps are not simple as above, hence they need documentation. Also, the author of the test case may leave the organization or goon a vacation or is sick and off duty or is very busy with other critical tasks. Arecently hire may be asked to execute the test case. Documented steps will helphim and also facilitate reviews by other stakeholders.

Step 4) The goal of test cases in software testing is to check behavior of the AUT foran expected result. This needs to be documented as below

Test Case #	Test Case Description	Test Data	Expected Result
1	Check response when valid email and password is entered	Email: guru99@email.com Password: lNf9^Oti7^2h	Login should besuccessful

During test execution time, the tester will check expected results against actualresults and assign a pass or fail status

Test Case #	Test Case Description	Test Data	Expected Result	ActualResult	Pass/Fail
1	Check response when valid email and password is entered	Email: guru99@email.com Password: lNf9^Oti7^2h	Login should be successful	Login was successful	Pass

Step 5) That apart your test case -may have a field like, Pre - Condition which specifies things that must in place before the test can run. For our test case, a

pre-condition would be to have a browser installed to have access

to the site under test. A test case may also include post-Conditions which

specifies anything thatapplies after the test case completes. For our test

case, a postcondition would be time & date of login is stored in the database

Result:

Thus, the test cases are verified Successfully.

Ex. No: 3	CONDUCT A TEST SUITE FOR ANY TWO WEB SITES
Date:	

Aim:

To conduct a test suite for any 2 websites.

Description:

Now we have 2 test cases:

- GoogleSignUpForm: which tests which fields are present in the Sign-Upform.
- GoogleSignUpErrors: which tries to submit the form without requiredfields.

To create a test suite in Selenium IDE:

1. Open Selenium IDE
2. Go to File>New Test Suite
3. To add test cases: Go to File>Add test case
4. Navigate to the location of your test case
5. Click on Add
6. Repeat steps 3-5 for more test cases

After you added test cases to your test suite, to export it as HTML, simply:

7. Go to File>Export as HTML
8. Enter a file name
9. Click on Save

In the HTML file, the test suite is a table in which each entry is a test case link:

After you generated your HTML suite file, you can run the suite using –htmlSuitecommand.

-htmlSuite requires you to specify:

- browserString (e.g. "*firefox")
- startURL (e.g. "http://www.google.com")

- suiteFile (e.g. "c:\absolute\path\to\my\HTMLSuite.html")
- resultFile (e.g.

"c:\absolute\path\to\my\results.html") Follow these

steps to run the suite:

10. Create a HTML file that will keep your test results
11. Open a command line
12. cd to the selenium-server.jar location.

13. For example: cd C:\selenium-remote-control-1.0.3\selenium-server-1.0.3

14. Run the command with the required arguments. For example:

15. java -jar selenium-server.jar -port 4546 -htmlSuite *firefox "http://www.google.com" "c:\SeleniumTest\TestSuite.html" "C:\test.html"

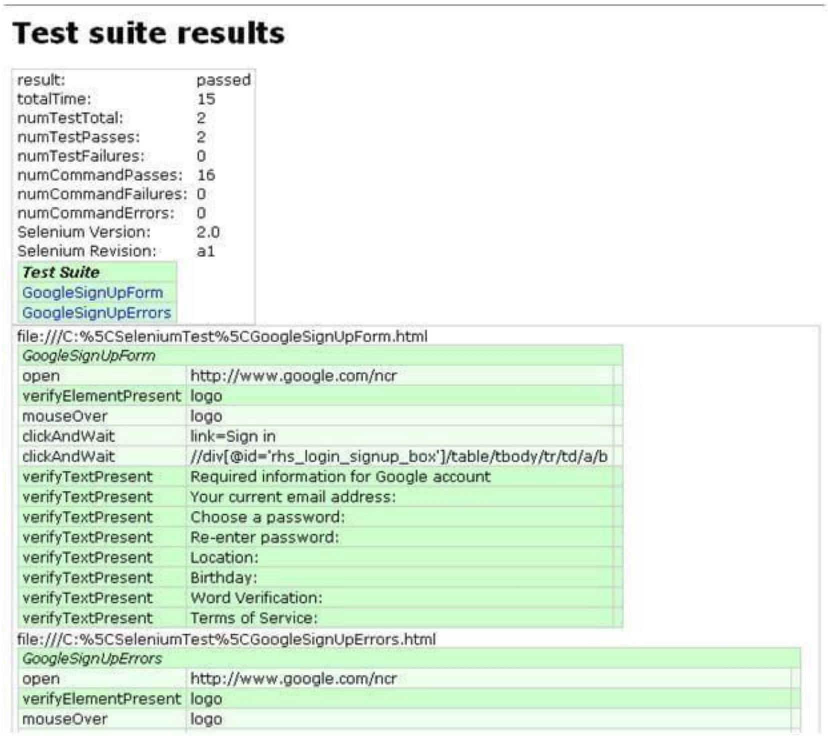
You will see how Test runner is invoked and starts running the test in Firefox.

After the test script is done, you can check the test results in the HTML file youprovided as argument.

Program :

```
<?xml                version="1.0"
encoding="UTF-8"?>

        <!DOCTYPE    html    PUBLIC    "-//W3C//DTD
 XHTML                    1.0                    Strict//EN"
 http://www.w3.org/TR/xhtml1/DTD/xhtml1-
 strict.dtd>

        <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

        <head>

            <meta content="text/html; charset=UTF-8" http-
            equiv="content-type" />

        <title>Test Suite</title>

        </head>

        <body>

        <table            id="suiteTable"            cellpadding="1"
    cellspacing="1" border="1"class="selenium">

                <tbody>

                <tr>

                        <td><b>Test Suite</b></td>

               </tr>

               <tr>    <td><href="file:///C:\SeleniumTestSuite\GoogleSignUp

                       Form.html>GoogleSignU

                       </a</td>

                      PForm
        </tr>

        <tr>

            <td><a
        href="file:///C:\SeleniumTestSuite\GoogleSignUpErrors.ht
        ml">GoogleSignUpErrors

            </a></td>

        </tr></tbody>

    </table></body>

    </html>
```

Output:



Test suite results

result:	passed
totalTime:	15
numTestTotal:	2
numTestPasses:	2
numTestFailures:	0
numCommandPasses:	16
numCommandFailures:	0
numCommandErrors:	0
Selenium Version:	2.0
Selenium Revision:	a1

Test Suite
GoogleSignUpForm
GoogleSignUpErrors

file:///C:%5CSeleniumTest%5CGoogleSignUpForm.html

GoogleSignUpForm	
open	http://www.google.com/ncr
verifyElementPresent	logo
mouseOver	logo
clickAndWait	link=Sign in
clickAndWait	//div[@id='rhs_login_signup_box']/table/tbody/tr/td/a/b
verifyTextPresent	Required information for Google account
verifyTextPresent	Your current email address:
verifyTextPresent	Choose a password:
verifyTextPresent	Re-enter password:
verifyTextPresent	Location:
verifyTextPresent	Birthday:
verifyTextPresent	Word Verification:
verifyTextPresent	Terms of Service:

file:///C:%5CSeleniumTest%5CGoogleSignUpErrors.html

GoogleSignUpErrors	
open	http://www.google.com/ncr
verifyElementPresent	logo
mouseOver	logo

Result:

Thus, a test suite is done for two websites successfully.

Ex. No: 4	DEVELOP A WEB PAGE WHICH CALCULATES THE GCD OF 2 NUMBERS. WRITE A SCRIPT FOR TESTING IT.
Date:	

Aim:

To Design a web page for calculating GCD numbers and write test for Scripting.

HTML code:

```
<!DOCTYPE html>
 <html>
 <head>
 <meta charset="utf-8">
 <title>GCD of two numbers</title>
</head>
 <body>
</body>
</html>
```

JavaScript Code:

```
function gcd_two_numbers(x, y)
{
if ((typeof x !== 'number') || (typeof y !== 'number'))return
 false;
```

```
     x = Math.abs(x);
   y  =  Math.abs(y);
   while(y) {
     var t = y;y
     = x % y; x =
     t;
     }
     return x;
   }
     console.log(gcd_two_numbers(12, 13));
   console.log(gcd_two_numbers(9, 3));
```

OUTPUT:

Test Data :

```
console.log(gcd_two_numbers(12, 13));
console.log(gcd_two_numbers(9, 3));
```

Output:

```
1
3
```

Result:

Thus Design a web page for calculating GCD numbers and write test for Scripting is completed successfully.

Ex. No: 5	DEVELOP A WEB PAGE WHICH CALCULATES THE GCD OF 2 NUMBERS. WRITE A SCRIPT FOR TESTING IT.
Date:	

Aim:

To develop and test a program to login a web.

Description:

Before we perform automation testing for login validation using Selenium and Java, there are some basic steps that need to be followed for whichever test case you intend to write. If you follow them, you will never have incomplete test casesin your automation suite:

1. Create a Selenium WebDriver instance.
2. Configure your browser if required (for example, maximize browser, disablebrowser notifications, etc.).
3. Navigate to the required URL (webpage).
4. Locate the HTML element.
5. Perform the action on the located HTML element.
6. Verify and validate the action (concluded step).
7. Take screenshots and generate the report using a framework for the testcases.

1. Create A Selenium WebDriver Instance

```
Webdriver driver=new ChromeDriver();
```

In order to launch the website in the desired browser, you need to set the system properties to the path of the driver for the required browser. In this Selenium Java tutorial, we will use Chromedriver for demonstrating Selenium login example with Java. The syntax for the same will be: System.setProperty("webdriver.chrome.driver", "File path for the Exe");

2. Configure Your Browser If Required

Based on the needs, we can configure the browser. For example, in this SeleniumJava tutorial regarding Selenium login with Java, a browser, by default, will be inminimized mode. However, we can set up the browser in the maximize mode.
Below is the syntax.
driver.manage().window().maximiz
e();

3. Navigate To the Required URL

Open the browser with the desired URL. All you have to do is write the belowsyntax and you have your URL open in the desired instantiated browser. driver.get("https://www.linkedin.com/login");

4. Locate The HTML Element

This is the heart of writing a Selenium script. For this to function, you need to havea clear understanding of the different locators used to find the HTML element. Youcan refer my below articles that talks about the different locators available in Selenium and how to locate the element with different examples:

a. ID locator in Selenium WebDriver

b. Name Locator in Selenium WebDriver

c. TagName Locator in Selenium WebDriver

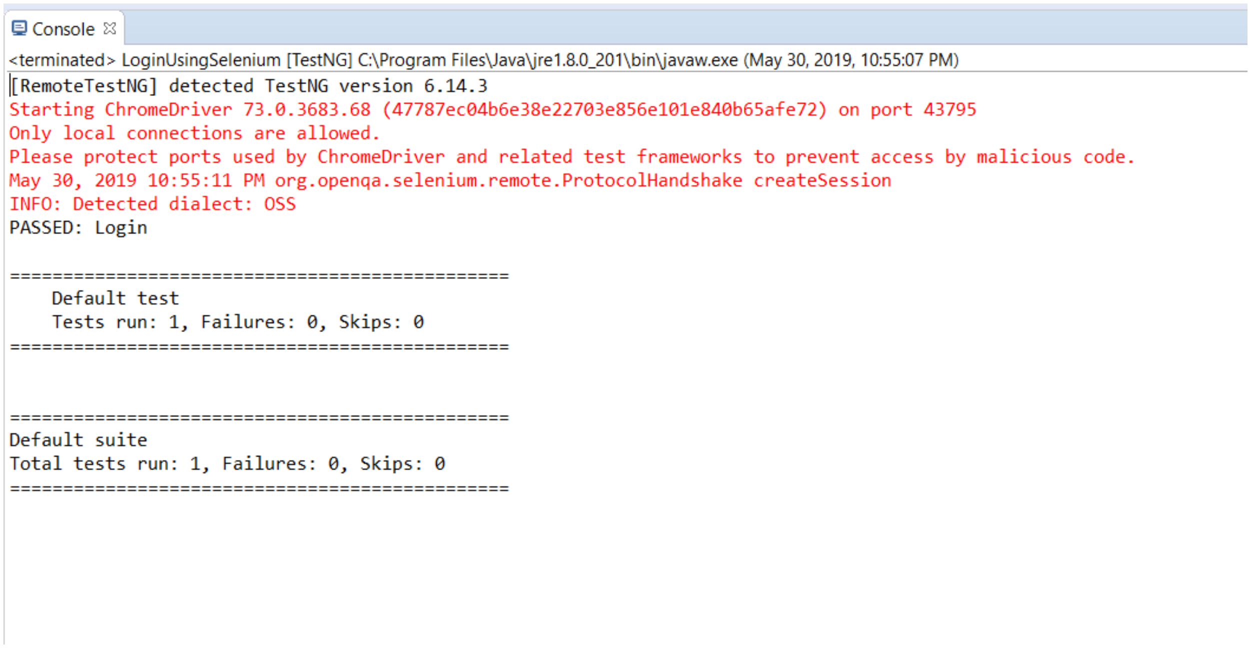
d. CSS Selector in Selenium WebDriver

e. XPath in Selenium WebDriver

Program:

```
import

java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

importorg.openqa.selenium.WebDriv

er;                                       import

org.openqa.selenium.WebElement;

import

org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import

org.testng.annotations.Test;

public                              class
```

```
LoginUsingSelenium {

  @Test public void login() { // TODO Auto-generated method

  stubSystem.setProperty("webdriver.chrome.driver", "path of driver");

 WebDriver driver=new ChromeDriver(); driver.manage().

 window().maximize();driver.get("https://www.linkedin.com/login");

 WebElement    username=driver.findElement(By.id("username"));

 WebElement  password=driver.findElement(By.id("password"));

 WebElement

 login=driver.findElement(By.xpath("//button[text()='Sign            in']"));

 username.sendKeys("example@gmail.com");

 password.sendKeys("password");               login.click();                String

 actualUrl="https://www.linkedin.com/feed/";

 String                                    expectedUrl=

 driver.getCurrentUrl();

 Assert.assertEquals(expectedUrl,actualUrl

 ); }                                                          }
```



```
<terminated> LoginUsingSelenium [TestNG] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (May 30, 2019, 10:55:07 PM)
[RemoteTestNG] detected TestNG version 6.14.3
Starting ChromeDriver 73.0.3683.68 (47787ec04b6e38e22703e856e101e840b65afe72) on port 43795
Only local connections are allowed.
Please protect ports used by ChromeDriver and related test frameworks to prevent access by malicious code.
May 30, 2019 10:55:11 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
PASSED: Login

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Failures: 0, Skips: 0
===============================================
```

Import

java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import

org.openqa.selenium.WebDriver;

import

org.openqa.selenium.WebElement;

import

org.openqa.selenium.chrome.ChromeDriver;

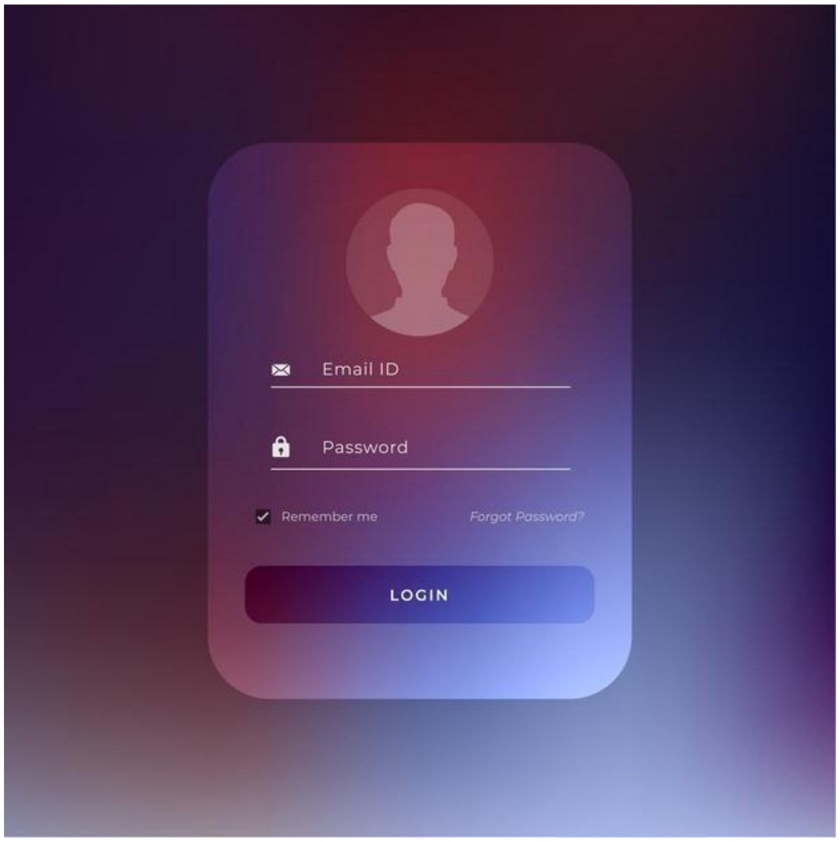
import org.testng.Assert;

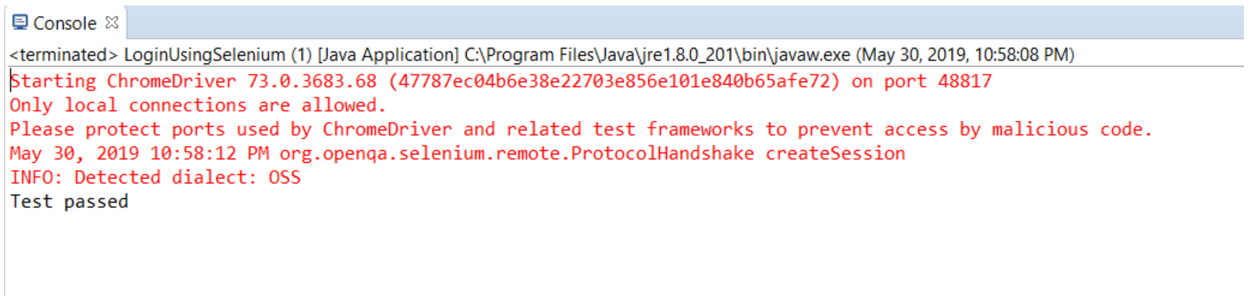
public class LoginUsingSelenium {

   public static void main(String[] args) { // TODO Auto-generated

```
method stubSystem.setProperty("webdriver.chrome.driver", "
path of driver ");
WebDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.linkedin.com/login");
WebElement username=driver.findElement(By.id("username"));
WebElement password=driver.findElement(By.id("password"));
WebElement
login=driver.findElement(By.xpath("//button[text()='Sign in']"));
username.sendKeys("example@gmail.com");
password.sendKeys("password"); login.click(); String
actualUrl="https://www.linkedin.com/feed/";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed"); } else { System.out.println("Test failed"); } }
}
```

Output:





```
Console
<terminated> LoginUsingSelenium (1) [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (May 30, 2019, 10:58:08 PM)
Starting ChromeDriver 73.0.3683.68 (47787ec04b6e38e22703e856e101e840b65afe72) on port 48817
Only local connections are allowed.
Please protect ports used by ChromeDriver and related test frameworks to prevent access by malicious code.
May 30, 2019 10:58:12 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
Test passed
```

Result:

The given program to Write and test a program to Update student records intoExcel file in table format is successfully executed.

Ex. No: 6	WRITE AND TEST A PROGRAM TO UPDATE STUDENT RECORDS INTO EXCEL FILE IN TABLE FORMAT
Date:	

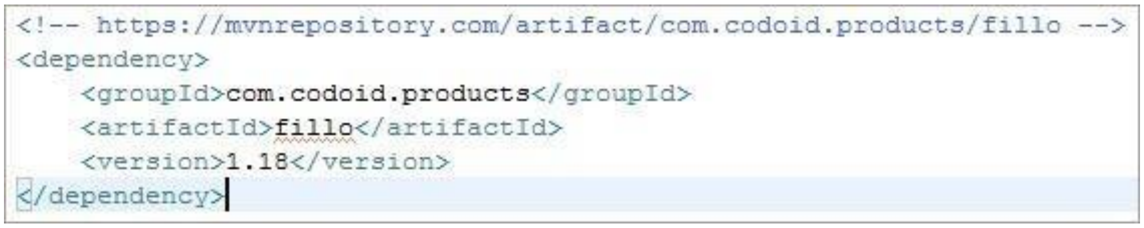
Aim:

Write and test a program to Update student records into Excel file in table format.

Description:

Reading or writing data is one of the most commonly used operations, either fetching values from database tables or fetching values from an excel sheet andusing them for performing analytics.

Program:



```
<!-- https://mvnrepository.com/artifact/com.codoid.products/fillo -->
<dependency>
    <groupId>com.codoid.products</groupId>
    <artifactId>fillo</artifactId>
    <version>1.18</version>
</dependency>
```

f. It is an excel API for Java language.

g. It supports .xls and .xlsx files.

h. It supports SELECT, UPDATE, and INSERT queries.

i. Use with or without the WHERE clause and LIKE clause.

SELECT Operation: SELECT statement performs the same function, as it does infetching the values from a table and display to the end-user, same way here the SELECT statement returns data from an excel sheet.

Syntax :

SELECT * From Sheet Name

UPDATE Operation: UPDATE statement modifies the existing records in theexcel sheet.

Syntax:

UPDATE sheet1 Set Column Name= 'Value'

INSERT Operation: INSERT statement inserts a new record in an excel sheet.

Syntax:

INSERT INTO Sheet Name (ColumnName1, ColumnName2) VALUES ('Val1','Val2')

Perform the same operations with the WHERE and LIKE operators:

j. "SELECT * from Sheet Name where ID=1 and name='Jesus'"
k. "SELECT * from Sheet Name where column1=value1 and column2=value2and column3=value3"
l. "UPDATE Sheet Name Set Country='UK' where ID=10 and name='Jesus'"
m. "SELECT * from Sheet Name where Name like 'Jes%'"

Execution Steps To Be Followed For SELECT/INSERT/UPDATE Operation:#1) //Create an Object of Fillo Class.

Fillo fillo = new Fillo();

#2) //Create an Object for Connection class and use the getConnection() method defined inside Fillo class, to establish the connection between the excel sheet andFillo API's.

Connection connection = fillo.getConnection("excelPath");

#3) // Select all the values present in a sheet. Those present inside the excel andstore its output in a string variable.

String strSelectQuerry = "Select * from&nbsp; SheetName";

#4) // execute the Select query and store the result in a Recordset class present inthe Fillo API.

Recordset recordset =connection.executeQuery(strSelectQuerry);

#5) //use while loop to iterate through all the columns and rows available in thesheet present inside the excel file.

while(recordset.next()){
// through getfield() method&nbsp; retrieve the data present in a particular

columnSystem.out.println(recordset.getField("Column1"));&nbsp; }

#6) //Use an update query to update the details in the excel file.

String strUpdateQuerry = "Update Data Set SiteTitle =

'SoftwareTestingHelp.com' ";connection.executeUpdate (strUpdateQuerry);

#7) //Use Insert query to insert data in the excel sheet.

String strInsertQuerry = "INSERT INTO Data

(SiteTitle,SiteTopic) Values('Bharat','NewDelhi')

connection.executeUpdate (strInsertQuerry);

#8) // close the recordset to avoid a memory leak.

recordset. Close();

#9) // close the connection to avoid

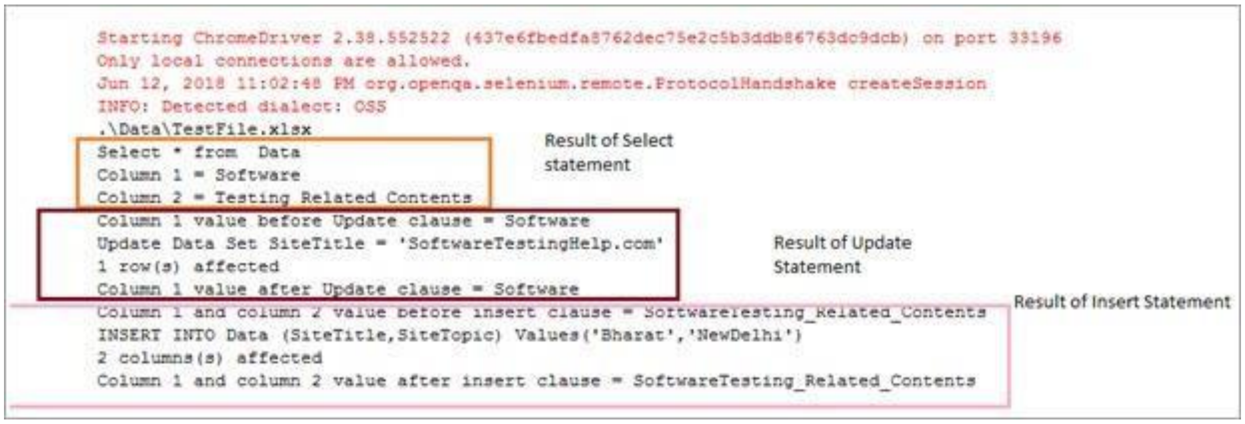
memory leak.connection. Close();

Selenium code:

```
package

softwareTestingHelp.Com;

import

org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import

org.openqa.selenium.chrome.ChromeDriver;

import

com.codoid.products.exception.FilloException;

import com.codoid.products.fillo.Connection;

import com.codoid.products.fillo.Fillo;

import

com.codoid.products.fillo.Recordset;

public class ReadWriteExcel {

   static WebDriver driver;
   //demo site -https://wordpress.com/start/about?ref=create-blog-lp
   //download jar file -
https://mvnrepository.com/artifact/com.codoid.products/fillo

   publicstaticvoid main(String args[]) throwsInterruptedException,
FilloException{

   //Calling          up          the          GoogleChrome          driver
   System.setProperty("webdriver.chrome.driver", "D:\\Srinivas\\New
 folder\\exe\\chromedriver.exe
   ");          driver          =
   newChromeDriver();

   //Opening    the    demo    site    -    wordpress.com
   driver.get("https://wordpress.com/start/about?ref=create-
   blog-lp");
   //Locating the Test data excel file
   String    excelPath    =    ".\\Data\\TestFile.xlsx";
   System.out.println(excelPath);
   //Create  an  Object  of  Fillo
   ClassFillo fillo = newFillo();
   //Create an Object for Connection class and use getConnection()
   //method defined inside Fillo class, to establish connection
betweenexcelsheet and Fillo API's.
   Connection connection = fillo.getConnection(excelPath);
   //Select all the values present in a sheet, which is present
inside theexcel and store its output in a String variable
   String     strSelectQuerry     =     "Select     *     from     Data";
   System.out.println(strSelectQuerry);
   //Execute  the  Select  query  and  store  the  result  in  a
Recordset classpresent in Fillo API.
   Recordset recordset =null;
   recordset = connection.executeQuery(strSelectQuerry);
   //use  while  loop  to  iterate  through  all  columns  and  rows
available insheet present inside excel file
   while(recordset.next()){
   System.out.println("Column          1          =          "
   +recordset.getField("SiteTitle"));       String       siteTitle       =
   recordset.getField("SiteTitle");

driver.findElement(By.xpath("//input[@name='siteTitle']")).clear();
driver.findElement(By.xpath("//input[@name='siteTitle']")).sendKeys(siteTitle);
   System.out.println("Column              2              =              "
   +recordset.getField("SiteTopic"));      String      siteTopic      =
   recordset.getField("SiteTopic");

   driver.findElement(By.xpath("//input[@name='siteTopic']")).clear();

driver.findElement(By.xpath("//input[@name='siteTopic']")).sendKeys(siteTopic);
   connection.close();
         }
   //Use update query to update the details in excel file
   Connection                    connection1                    =
   fillo.getConnection(excelPath);
   System.out.println("Column 1 value before Update clause = "
+recordset.getField("SiteTitle"));
   String  strUpdateQuerry  =  "Update  Data  Set
 SiteTitle ='SoftwareTestingHelp.com' ";
   System.out.println(strUpdateQuerry);
   connection1.executeUpdate(strUpdateQuerry);
   System.out.println("Column 1 value after Update clause = "
+recordset.getField("SiteTitle"));
   //Use Insert query to update the data in excel sheet
   Connection                    connection2                    =
   fillo.getConnection(excelPath);
   System.out.println("Column 1 and column 2 value before insert
clause =" +recordset.getField("SiteTitle")
   +recordset.getField("siteTopic"));
 String     strInsertQuerry     =     "INSERT     INTO     Data     (SiteTitle,SiteTopic)
Values('Bharat','NewDelhi')";
   System.out.println(strInsertQuerry);

   connection2.executeUpdate(strInsertQuerry);

   System.out.println("Column 1 and column 2 value after insert

   clause = "+recordset.getField("SiteTitle")
```

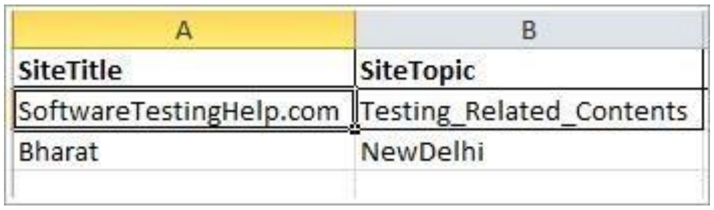
```
    +recordset.getField("siteTopic"));
   }
}
```

Output:



```
Starting ChromeDriver 2.38.552522 (437e6fbedfa8762dec75e2c5b3ddb86763dc9dcb) on port 33196
Only local connections are allowed.
Jun 12, 2018 11:02:48 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
.\Data\TestFile.xlsx
Select * from  Data
Column 1 = Software
Column 2 = Testing Related Contents
Column 1 value before Update clause = Software
Update Data Set SiteTitle = 'SoftwareTestingHelp.com'
1 row(s) affected
Column 1 value after Update clause = Software
Column 1 and column 2 value before insert clause = SoftwareTesting_Related_Contents
INSERT INTO Data (SiteTitle,SiteTopic) Values('Bharat','NewDelhi')
2 columns(s) affected
Column 1 and column 2 value after insert clause = SoftwareTesting_Related_Contents
```

Result of Select statement
Result of Update Statement
Result of Insert Statement



A	B
SiteTitle	SiteTopic
SoftwareTestingHelp.com	Testing_Related_Contents
Bharat	NewDelhi

Result:

The given program to Write and test a program to Update student records intoExcel file in table format is successfully executed.

Ex. No: 7	WRITE AND TEST A PROGRAM TO COUNT NUMBER OF ITEMS PRESENT ON A DESKTOP
Date:	

Aim:

To Write a program to count number of items in the desktop.

Java Code:

```
private const uint GET_ITEM_COUNT = 0x1000 +
    4;[DllImport("user32.DLL")]

    private static extern int SendMessage(IntPtr hWnd, uint Msg, int
wParam, int lParam);

    [DllImport("user32.DLL")]

    private static extern IntPtr FindWindow(string lpszClass, string
lpszWindow);

    [DllImport("user32.DLL")]

    private static extern IntPtr FindWindowEx(IntPtr hwndParent,
IntPtr hwndChildAfter,string lpszClass, string lpszWindow);

    public static int GetDesktopCount()

    {

        //Get the handle of the desktop listview

        IntPtr vHandle = FindWindow("Progman", "Program
        Manager");vHandle = FindWindowEx(vHandle, IntPtr.Zero,
"SHELLDLL_DefView", null);

        vHandle = FindWindowEx(vHandle, IntPtr.Zero, "SysListView32",
"FolderView");

        //Get total count of the icons on the desktop
```

```
int vItemCount = SendMessage(vHandle, GET_ITEM_COUNT, 0,

0)return vItemCount;
```

OUTPUT:

10 depends on desktop

Result:

Thus Write a program to count number of items in the desktopis completed successfully.

Ex. No: 8	WRITE AND TEST A PROGRAM TO GET THE NUMBER OF LIST ITEMS SELECTED IN A LIST / COMBO BOX
Date:	

Aim:

To create a simple program to get the number of list items selected in a list / combo box

Algorithm:

Step 1: JComboBox() : creates a new empty JComboBox .

Step 2:JComboBox(ComboBoxModel M) : creates a new JComboBox with items from specifiedComboBoxModel

Step 3:JComboBox(E [ ] i) : creates a new JComboBox with items from specified array.

Step 4:JComboBox(Vector items) : creates a new JComboBox with items from the specified vector

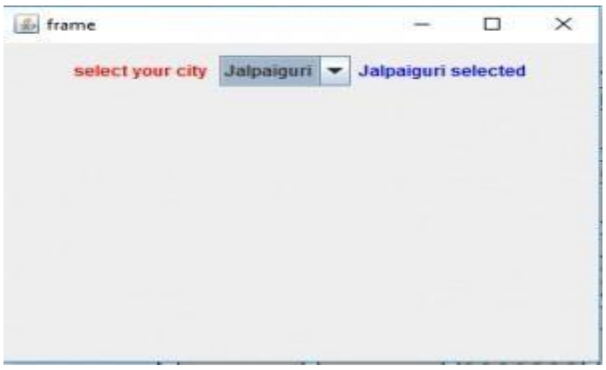
Program:

```
import
java.awt.event.*;
import
java.awt.*;
import
javax.swing.*;
class solve extends JFrame implements ItemListener {
  static JFrame
  f;        static
  JLabel l, l1;
  static JComboBox c1;
  public static void main(String[] args)
```

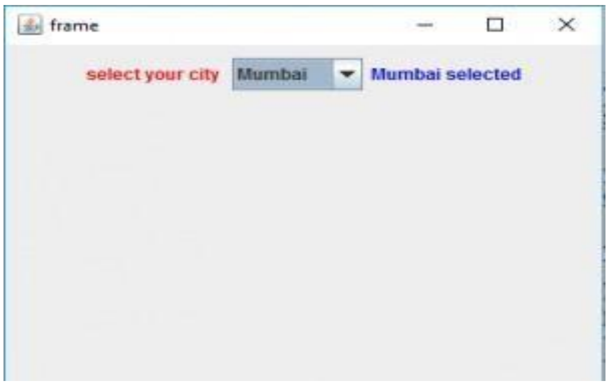
```
{
   f          =          new
    JFrame("frame");
    solve s = new

   solve();
   f.setLayout(n
   ew
   FlowLayout()
   );
   String s1[] = { "Jalpaiguri", "Mumbai", "Noida", "Kolkata", "New Delhi" };
   c1          =          new
   JComboBox(s1);
   c1.addItemListener
   (s);
   l = new JLabel("select your city ");
   l1 = new JLabel("Jalpaiguri selected");
   l.setForeground(Color
   .red);
   l1.setForeground(Col
   or.blue);
   JPanel   p   =   new
   JPanel();p.add(l);
    p.add(c1);
    p.add(l1);
    f.add(p);
   f.setSize(400,
   300); f.show();
}
public void itemStateChanged(ItemEvent e)
{
```

```
            if   (e.getSource()   ==   c1)   {
               l1.setText(c1.getSelectedIt
               em() + " selected");

            }

         }

      }
```

Output :



select your city Jalpaiguri Jalpaiguri selected



select your city Mumbai Mumbai selected

Result:

Thus create a simple program to get the number of list items selected in a list / combo box was createdsuccessfully.