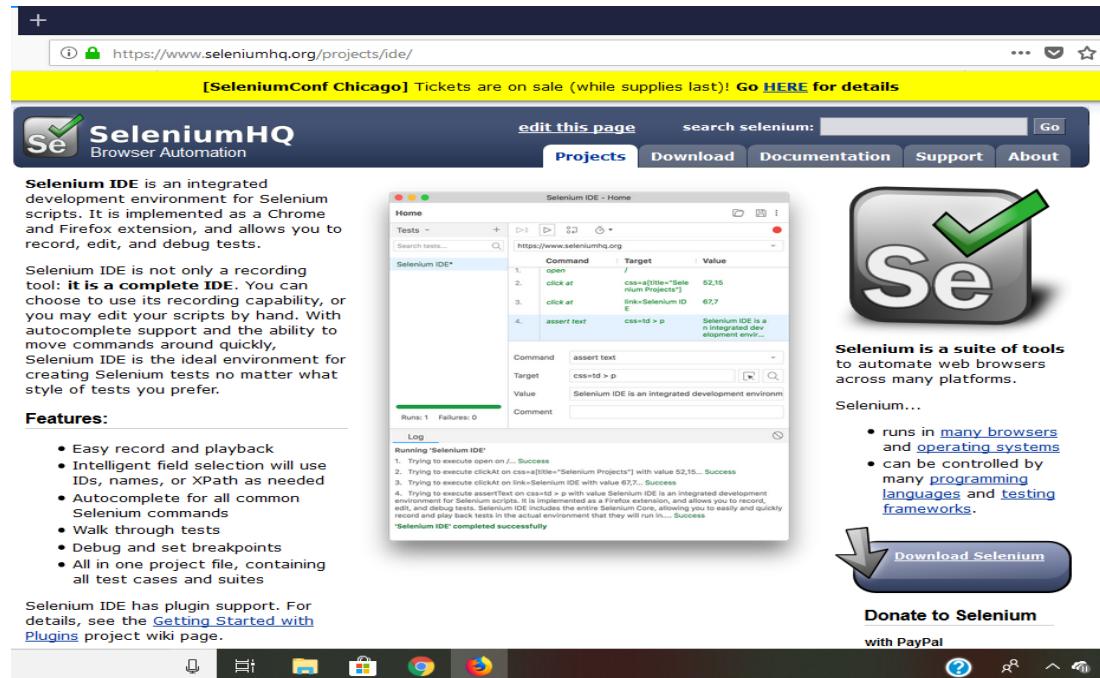


### PRACTICAL NO. 1

Aim:-Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.

Following are the steps for installing Selenium IDE on Mozilla Firefox.

1. In Firefox, search for <https://www.seleniumhq.org/>.



2. Click on 'for Firefox'

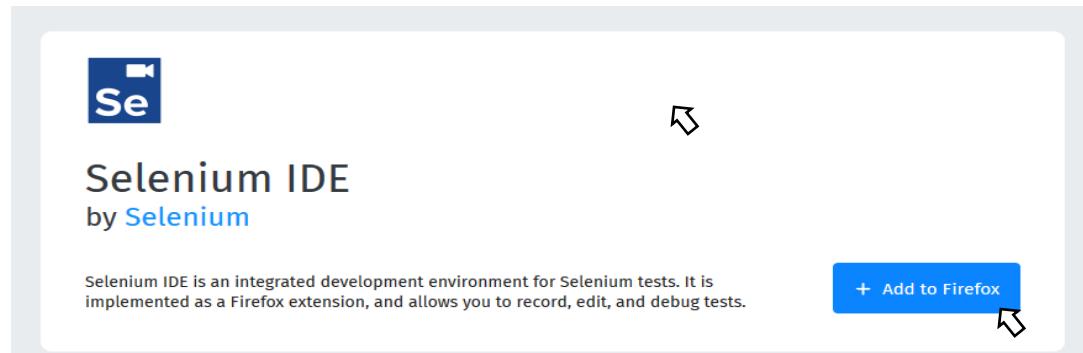
#### Selenium IDE

Selenium IDE is a Chrome and Firefox plugin which records and plays back user interactions with the browser. Use this to either create simple scripts or assist in exploratory testing.

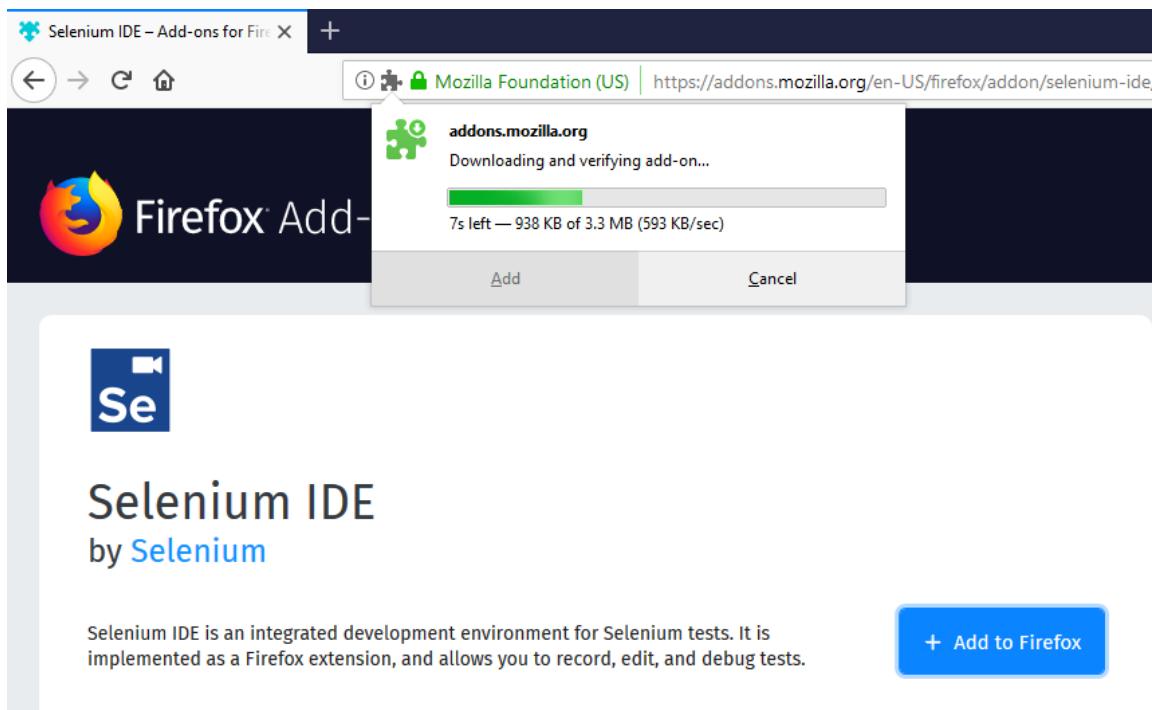
Download latest released version [for Chrome](#) or [for Firefox](#) or view the [Release Notes](#).

Download [previous IDE versions here](#).

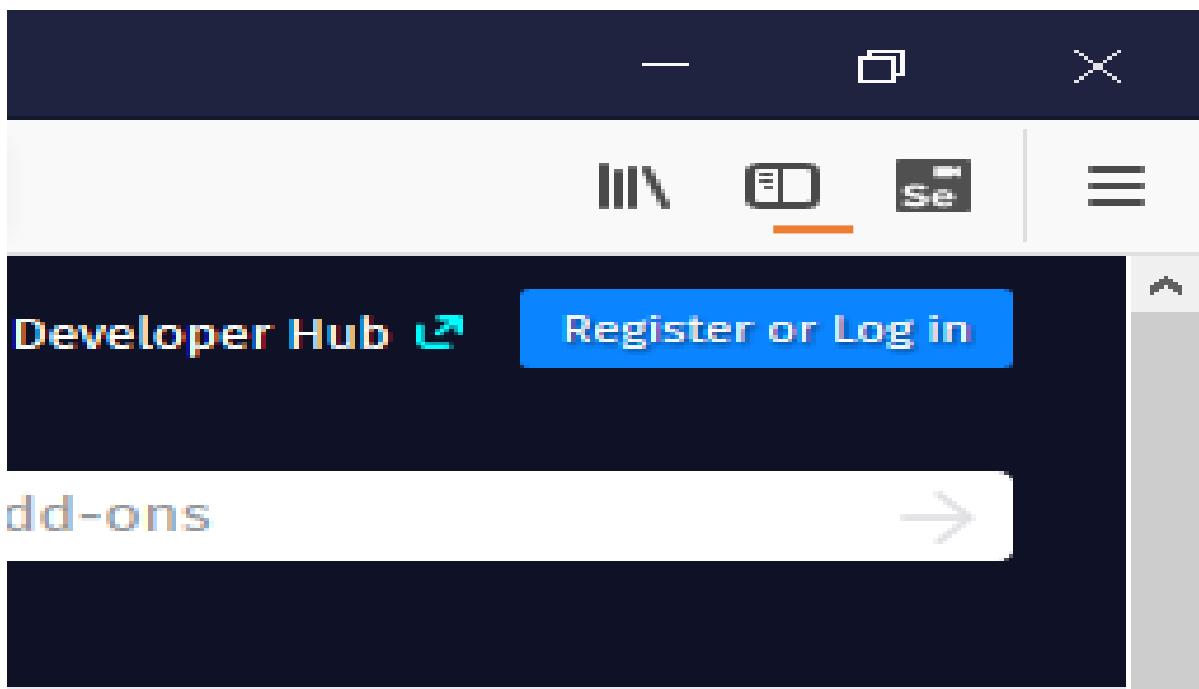
3. Click on '+Add to Firefox'



4. Wait for the download to finish



5. Click on the 'Se' icon which is on the Firefox toolbar

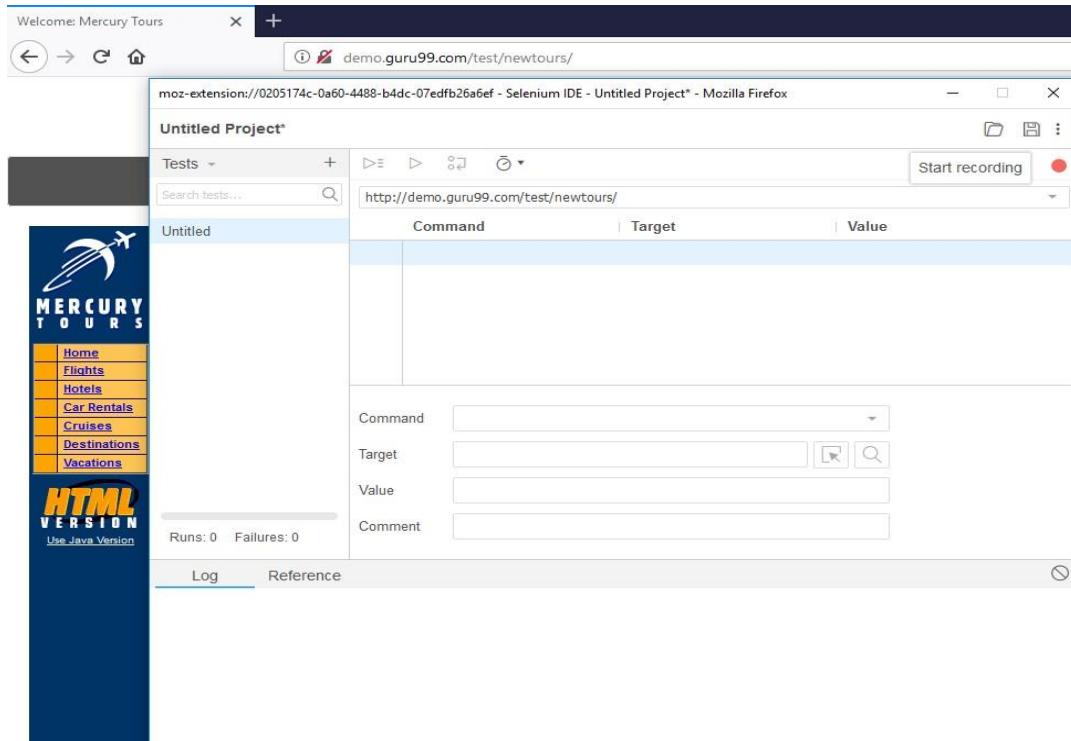


## Software Testing And Quality Assurance

For demo.guru99.com →

Step 1: Open Mozilla Firefox and search for

<http://demo.guru99.com/test/newtours/>. Copy and paste the same URL into Selenium IDE, start recording subsequently.



Step 2: Right click on the web page and cursor over Selenium IDE. Then, select Assert Title from the side bar.



## Software Testing And Quality Assurance

Step 3: Type anything into the user name and password blocks. Click Sign-In after entering.

The screenshot shows a login form on a website. At the top left is a yellow button labeled "Find A Flight". Below it is a text block: "Registered users can sign-in here to find the lowest fare on participating airlines." The form has two input fields: "User Name:" containing "invalidUNN" and "Password:" containing a series of black dots. At the bottom is a yellow "Sign-In" button with a right-pointing arrow.

Step 4: Abreast, you will see the commands in Selenium IDE which you had performed on website.

The screenshot shows the Selenium IDE interface in Mozilla Firefox. The title bar reads "moz-extension://0205174c-0a60-4488-b4dc-07edfb26a6ef - Selenium IDE - Untitled Project\* - Mozilla Firefox". The main area displays a table of recorded test steps:

	Command	Target	Value
1.	open	/test/newtours/	
2.	double click at	css=img[alt="Mercury Tours"]	56,74
3.	assert title	Welcome: Mercury Tours	
4.	click at	name=password	20,10
5.	type	name=password	invalidPWD
6.	click at	name=userName	27,14

Below the table are four input fields: "Command", "Target", "Value", and "Comment". The status bar at the bottom shows "Runs: 0 Failures: 0".

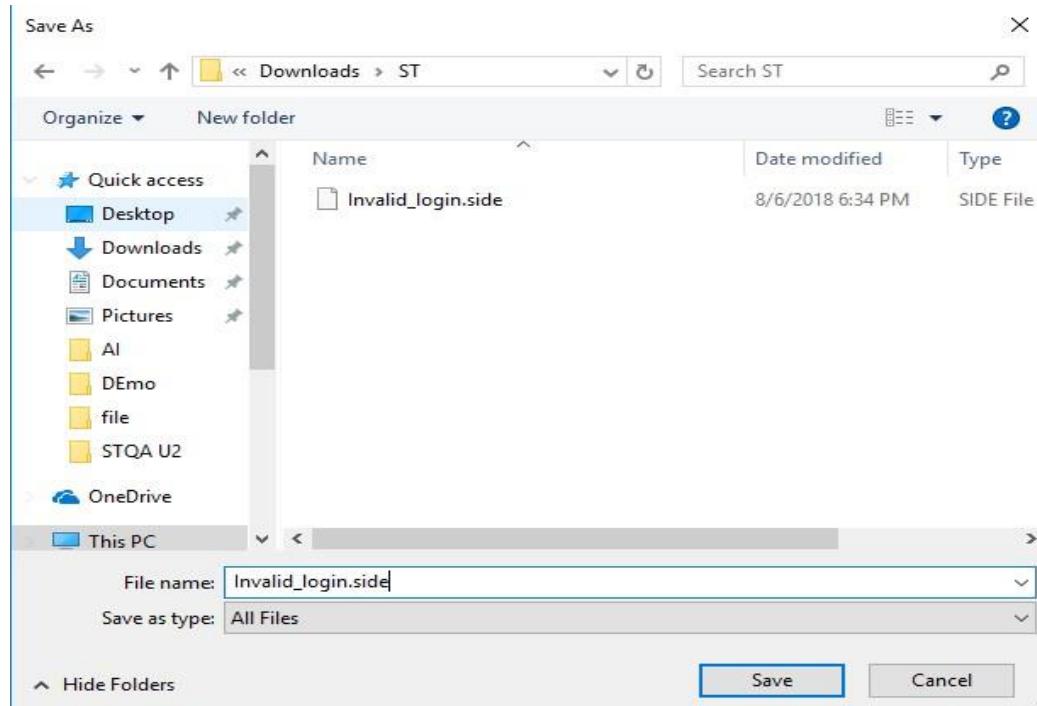
## Software Testing And Quality Assurance

The screenshot shows the Selenium IDE interface. At the top, it displays the URL `http://demo.guru99.com`. Below the URL, a table lists the recorded steps:

	Command	Target	Value
7.	click at	name=userName	23,6
8.	type	name=userName	invalidUNN
9.	click at	//div[3]	846,197
10.	click at	name=submit	24,8

Below the table, there are input fields for Command, Target, Value, and Comment. At the bottom left, it shows "Runs: 0 Failures: 0". At the bottom right, there are tabs for Log and Reference.

Step 5: Save your test case with a name followed by the **.side** extension.



## Software Testing And Quality Assurance

Step 6: Go to the Selenium IDE and click onto Run current test then you will see the 'test case has completed successfully' in the Log section.

Tests + Run current test: i9.com

	Command	Target	Value
7.	click at	name=userName	23,6
8.	type	name=userName	invalidUNN
9.	click at	//div[3]	846,197
10.	click at	name=submit	24,8

Command:   
Target:     
Value:   
Comment:

Runs: 0 Failures: 0

Log Reference

Tests + Run current test: http://demo.guru99.com

	Command	Target	Value
6.	click at	name=username	27,14
7.	click at	name=userName	23,6
8.	type	name=userName	invalidUNN
9.	click at	//div[3]	846,197
10.	click at	name=submit	24,8

Command: open  
Target: /test/newtours/    
Value:   
Comment:

Runs: 1 Failures: 0

Log Reference

5. type on name=password with value invalidPWD... OK  
6. clickAt on name=userName with value 27,14... OK  
7. clickAt on name=userName with value 23,6... OK  
8. type on name=userName with value invalidUNN... OK  
9. clickAt on //div[3] with value 846,197... OK  
10. clickAt on name=submit with value 24,8... OK  
**'Invalid\_login' completed successfully**

## Software Testing And Quality Assurance

For Gmail→

1. Go to [www.google.com](http://www.google.com) and search for Gmail.

The screenshot shows a browser window with the URL <https://www.google.com/gmail/about/>. On the left, there is a large image of a hand holding a smartphone displaying the Gmail mobile application. The phone screen shows the 'Primary' inbox with several messages from contacts like Salit Kulla, me, Tom, Anissa, and Tim Greer. On the right, the Selenium IDE extension is open, titled 'Untitled Project\*'. It has a toolbar with icons for running tests, saving, and deleting. Below the toolbar is a search bar labeled 'Search tests...'. A table lists a single test named 'gmail'. The table columns are 'Command', 'Target', and 'Value'. The 'Command' column contains 'open', 'click at', 'type', and 'click at'. The 'Target' column contains 'url=https://www.google.com/gmail/about/' and various CSS and ID-based targets. The 'Value' column contains 'purvagashe10' and other numerical values. At the bottom of the Selenium IDE interface, there are buttons for 'Command', 'Target', 'Value', and 'Comment', and status indicators 'Runs: 0' and 'Failures: 0'.

2. Click on Gmail icon

This screenshot shows the Selenium IDE after the test has been run. The title bar says 'Untitled Project\*' and the URL is 'moz-extension://4475d5ac-e5b9-4f14-9ef5-4914cf0e6d11 - Selenium ID...'. The main area displays the recorded test steps:

Index	Command	Target	Value
2.	click at	css=a.gmail-nav__nav-link.gmail-nav__nav-link__sign-in	47,31
3.	type	id=identifierId	purvagashe10
4.	click at	css=#identifierNext > content.CwaK	21,8

Below the table, there are input fields for 'Command', 'Target', 'Value', and 'Comment'. At the bottom, there are tabs for 'Log' (which shows 'Running \'gmail\'') and 'Reference'. The log also lists the executed steps:

1. open on /gmail/about/... OK
2. clickAt on css=a.gmail-nav\_\_nav-link.gmail-nav\_\_nav-link\_\_sign-in with value 47,31...

## Software Testing And Quality Assurance

### 3. Enter some ID and Password

The screenshot shows the Selenium IDE interface with the following details:

- Title Bar:** moz-extension://4475d5ac-e5b9-4f14-9ef5-4914cf0e6d11 - Selenium IDE...
- Toolbar:** Includes icons for file operations (New, Save, etc.) and a red circular button.
- Project:** Untitled Project\*
- Test List:** Shows a single test named "gmail\*".
- URL:** https://www.google.com
- Table:** Displays recorded steps:

	Command	Target	Value
5.	click at	css=#forgotPassword > content.Cwa	38,7
6.	click at	//div[@id='view_container']/div/div[2]/div/div[2]/div[2]/div/content/span	76,16
- Form Fields:** Command, Target, Value, Comment input fields.
- Log:** Shows the execution log:

```
3. type on id=identifierId with value purvagashe10... OK
4. clickAt on css=#identifierNext > content.Cwak9 > span.RveJvd.snByac with value 21,8... OK
5. clickAt on css=#forgotPassword > content.Cwak9 > span.RveJvd.snByac with value 38,7... OK
6. clickAt on //div[@id='view_container']/div/div[2]/div/div[2]/div[2]/div/content/span with value 76,16... OK
'gmail' completed successfully
```
- Status:** Runs: 1 Failures: 0

### 4. Stop recording and then click on play

The screenshot shows the Selenium IDE interface with the following details:

- Title Bar:** moz-extension://4475d5ac-e5b9-4f14-9ef5-4914cf0e6d11 - Selenium IDE...
- Toolbar:** Includes icons for file operations (New, Save, etc.) and a red circular button.
- Project:** Untitled Project\*
- Test List:** Shows a single test named "gmail\*".
- URL:** https://www.google.com
- Table:** Displays recorded steps:

	Command	Target	Value
13.	mouse over	id=gb_71	
14.	click at	id=gb_71	42,15
15.	mouse out	id=gb_71	
16.	click at	css=h3.r > a	104,13
- Form Fields:** Command, Target, Value, Comment input fields.
- Status:** Runs: 0 Failures: 0

## Software Testing And Quality Assurance

For Yahoo→

1. Go to [www.yahoo.com](http://www.yahoo.com) and start recording.

The screenshot shows the Selenium IDE interface with the title "Untitled Project\*". The URL "https://in.yahoo.com" is entered in the address bar. A table below lists five recorded actions:

	Command	Target	Value
2.	run script	window.scro	lITo(0,441)
3.	run script	window.scro	lITo(0,792)
4.	run script	window.scro	lITo(0,1992)
5.	run script	window.scro	lITo(0,2756)

Below the table are input fields for Command, Target, Value, and Comment. The status bar at the bottom shows "Runs: 0 Failures: 0".

2. Perform some actions and then click on play button.

The screenshot shows the Selenium IDE interface with the title "Untitled Project\*". The URL "https://in.yahoo.com" is entered in the address bar. The recorded actions remain the same as in the previous screenshot. The status bar at the bottom now shows "Runs: 1 Failures: 0".

In the Log tab, the following output is displayed:

```
2. runScript on window.scrollTo(0,441)... OK
3. runScript on window.scrollTo(0,792)... OK
4. runScript on window.scrollTo(0,1992)... OK
5. runScript on window.scrollTo(0,2756)... OK
'yahoo' completed successfully
```

## Software Testing And Quality Assurance

### For Selenium→

1. Go to <https://www.seleniumhq.org/projects/ide/> and start recording .

The screenshot shows the Selenium IDE homepage. At the top, there's a banner for 'SeleniumConf Chicago' with a link to 'Go HERE for details'. Below the banner, the 'Selenium HQ' logo is displayed, followed by the text 'Browser Automation'. A sidebar on the left contains sections for 'Selenium IDE' (described as an integrated development environment for Selenium scripts), 'Features' (listing record and playback, intelligent field selection, autocomplete, etc.), and 'Selenium IDE has plugin support'. The main content area shows the 'Selenium IDE - Home' interface with a test log containing four recorded commands:

Command	Target	Value
open	/projects/	
click at	css=a[title="Selenium Projects"]	52,15
click at	link=Selenium ID	67,7
assert text	css=td > p	Selenium IDE is a... environment

Below the log, there's a 'Log' section with the message "'Selenium IDE' completed successfully'". On the right side of the interface, there's a summary of what Selenium does, a 'Selenium...' section with bullet points, and a 'Donate to Selenium with PayPal' button.

2. Perform some actions and then click on play button.

The screenshot shows the Selenium IDE interface with an 'Untitled Project' titled 'selenium\*'. The test log on the left lists the following recorded actions:

Command	Target	Value
open	/projects/	
click at	css=a[title="Selenium Projects"]	44,18
click at	css=a[title="Technical references and"]	60,5
run script	window.scre	

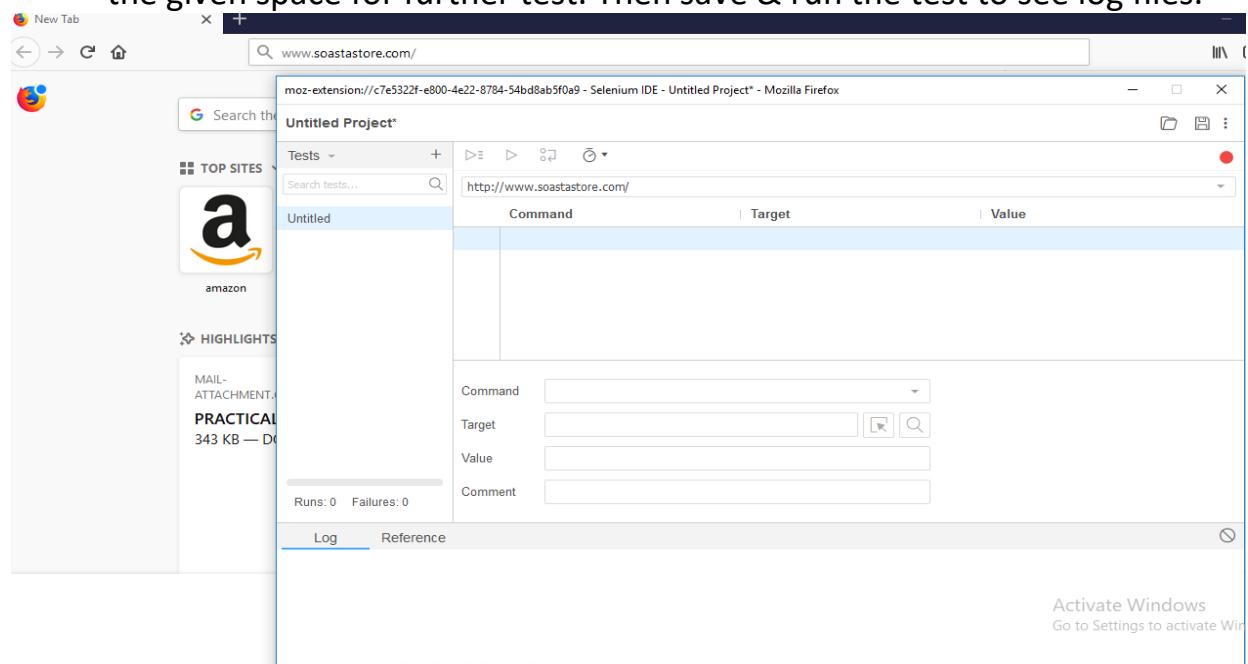
The status bar at the bottom shows 'Runs: 1' and 'Failures: 0'. The bottom navigation bar includes tabs for 'Log' (which displays the message "'selenium' completed successfully') and 'Reference'.

### PRACTICAL NO.2

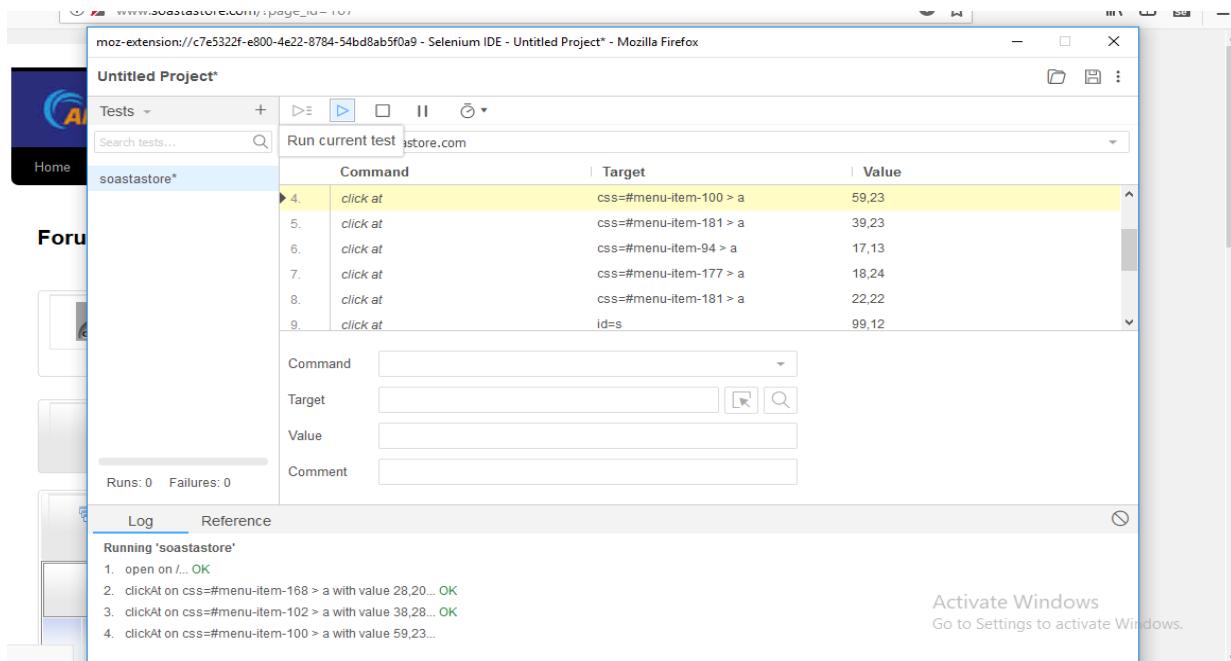
Aim: To conduct a test suite for any 2 websites.

#### A. For [www.soastastore.com](http://www.soastastore.com)

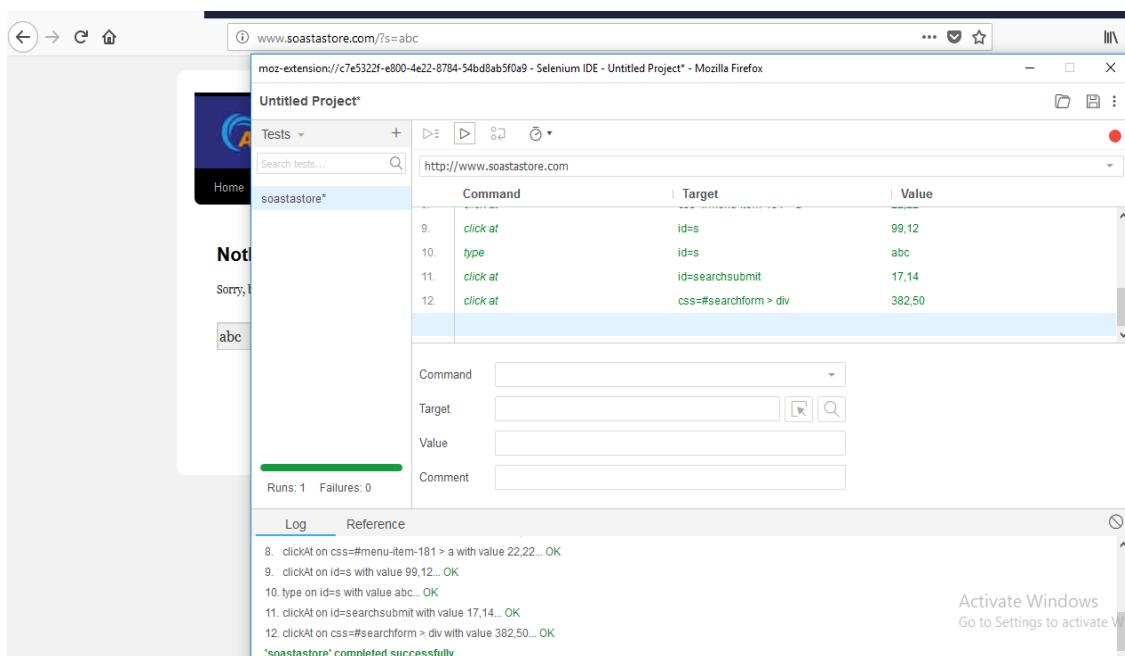
1. Open the website and open selenium IDE. Copy the URL and paste it in the given space for further test. Then save & run the test to see log files.



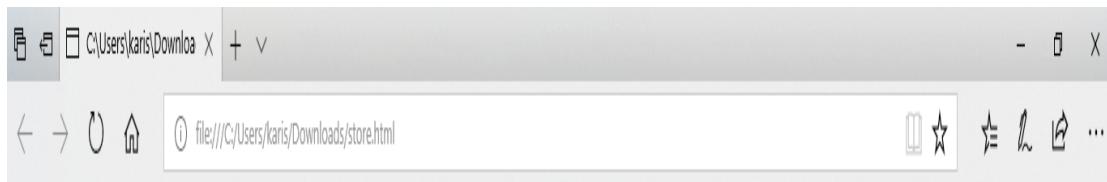
## Software Testing And Quality Assurance



- After every successful run it gives message “completed successfully” with green text highlighted.



- Save it in the HTML file and open it on your browser webpage.

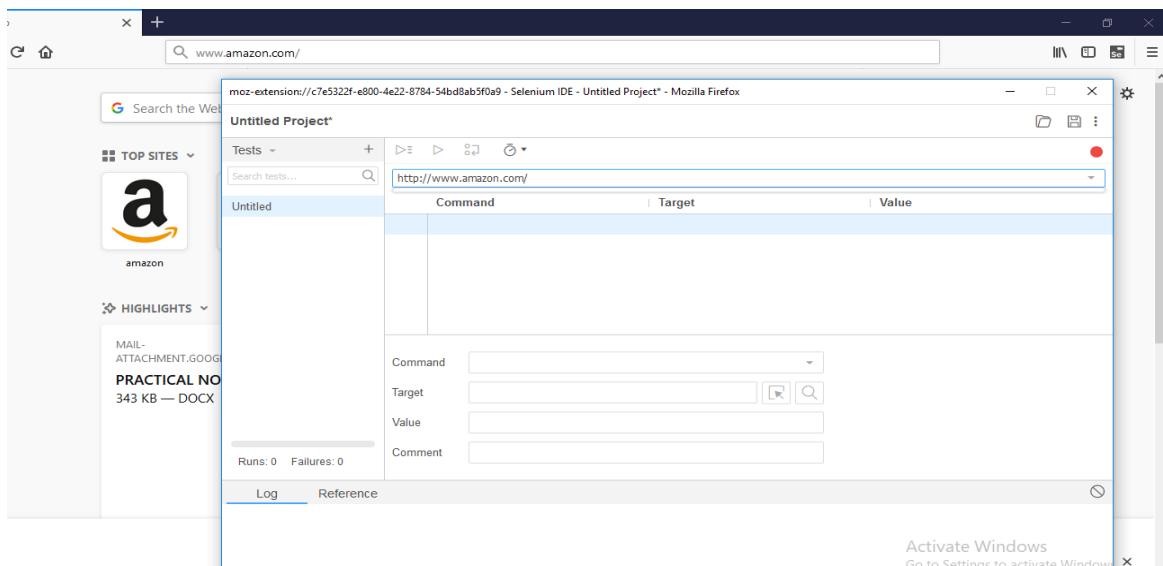


```
{ "id": "86df13fa-e9cb-48e8-b445-22e94635deb0", "name": "Untitled Project", "url": "http://www.soastastore.com", "tests": [ { "id": "8d2f0f4f-73b9-4684-bf36-bf94b918d205", "name": "soastastore", "commands": [ { "id": "0509468d-720e-47c4-91e5-cad2efc9baef", "comment": "", "command": "open", "target": "/", "value": "" }, { "id": "26d3b7fd-feb2-4876-b43b-d9c26ee92366", "comment": "", "command": "clickAt", "target": ".css>#menu-item-168>a", "value": "28.20" }, { "id": "801bce3e-b731-4bce-a82f-5e2c5b2faa2b", "comment": "", "command": "clickAt", "target": ".css>#menu-item-102>a", "value": "38.28" }, { "id": "315d7ba7-9954-4f05-83a9-355bf8306ad5", "comment": "", "command": "clickAt", "target": ".css>#menu-item-100>a", "value": "59.23" }, { "id": "d44e365f-72c2-4e2c-9a40-6698b8952bbf", "comment": "", "command": "clickAt", "target": ".css>#menu-item-181>a", "value": "39.23" }, { "id": "8159f341-de7f-4611-b398-45c19e8c29e0", "comment": "", "command": "clickAt", "target": ".css>#menu-item-94>a", "value": "17.13" }, { "id": "b00c8f40-197d-4408-9715-0aea62f45a27", "comment": "", "command": "clickAt", "target": ".css>#menu-item-177>a", "value": "18.24" }, { "id": "aeb46986-e2f0-44fb-8010-8e9dd773c9f8", "comment": "", "command": "clickAt", "target": ".css>#menu-item-181>a", "value": "22.22" }, { "id": "b2384d21-f230-45d6-b0d3-8f318ff1a3d5", "comment": "", "command": "clickAt", "target": "#id=s", "value": "99.12" }, { "id": "29295a3e-cc3e-4662-a3f6-0224439d24ad", "comment": "", "command": "type", "target": "#id=s", "value": "abc" }, { "id": "a626593f-24d8-4871-9add-9668bef036ad", "comment": "", "command": "clickAt", "target": "#searchsubmit", "value": "17.14" }, { "id": "c6dcfd47-8601-4382-89f7-988ab0c6c423", "comment": "", "command": "clickAt", "target": ".css=searchform>div", "value": "382.30" } ] }, "suites": [ { "id": "a00c5748-4fb6-4565-88dc-64e11f379ee2", "name": "Default Suite", "parallel": false, "timeout": 300, "tests": [ "8d2f0f4f-73b9-4684-bf36-bf94b918d205" ] } ], "urls": [ "http://www.soastastore.com" ], "plugins": [], "version": "1.0" }
```

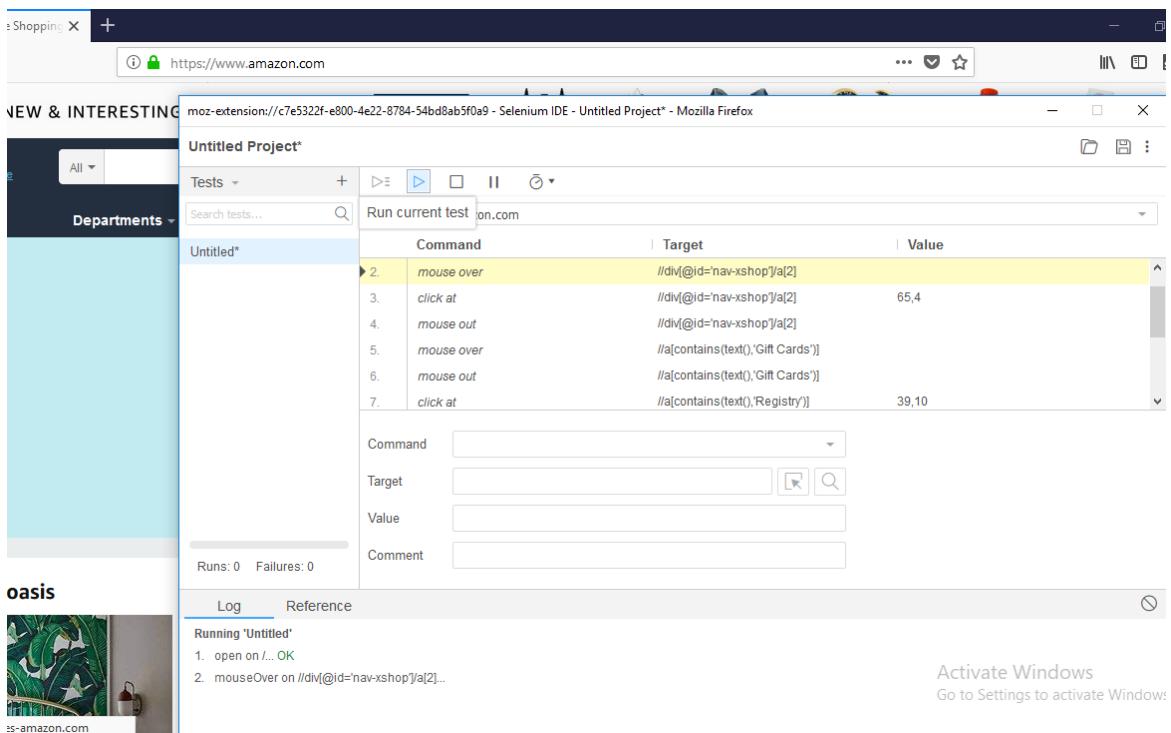
### B. For [www.amazon.com](http://www.amazon.com)

1. Open the website and open selenium IDE. Copy the URL and paste it in the given space for further test. Then save & run the test to see log files.

## Software Testing And Quality Assurance

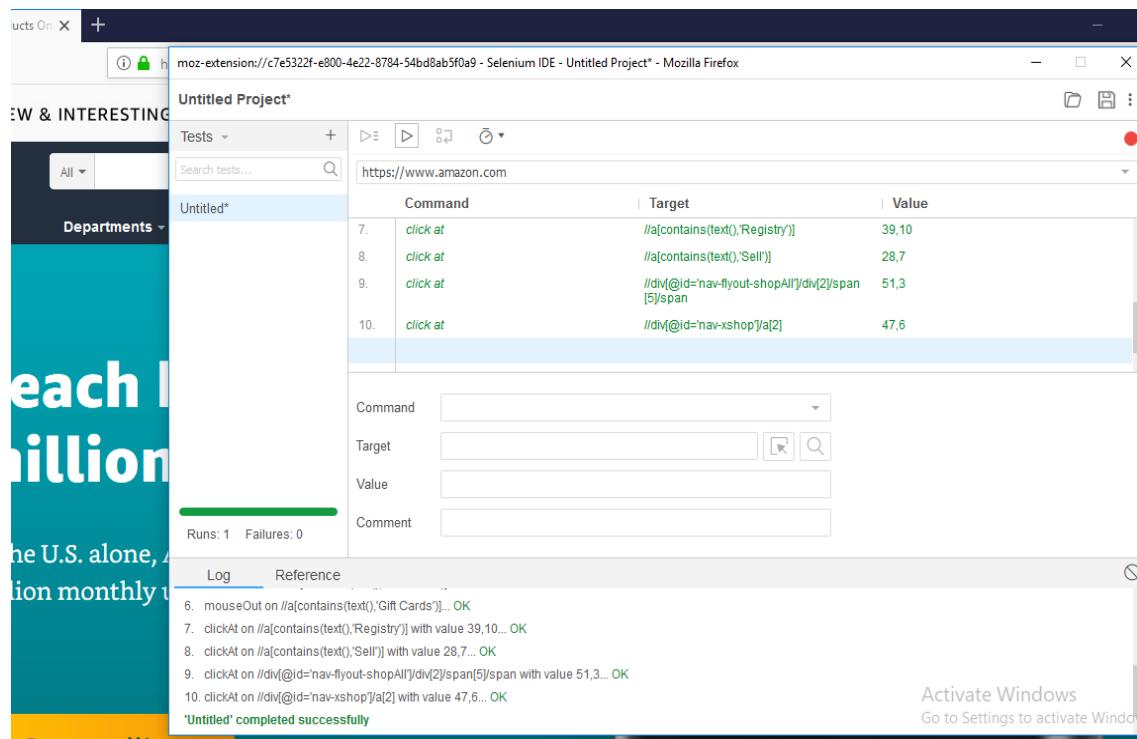


2. After saving the project, click on play and see how commands execute...

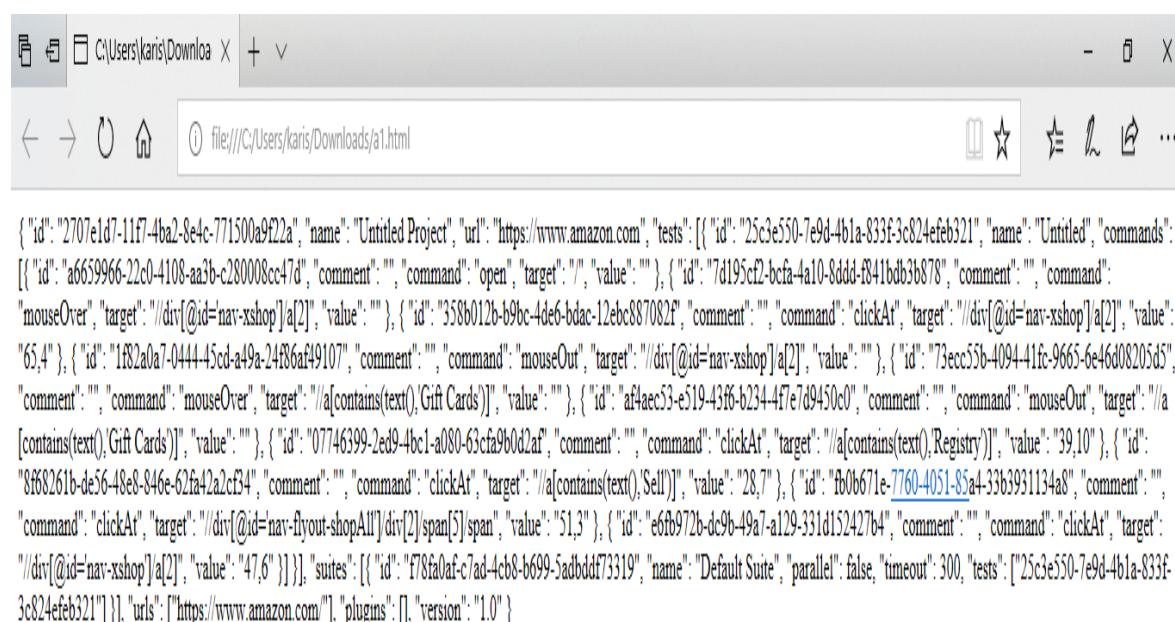


## Software Testing And Quality Assurance

3. After successfully completing you will see message "completed successfully" with green text highlighted with log .



4. Save it in the HTML file and open it on your browser webpage.



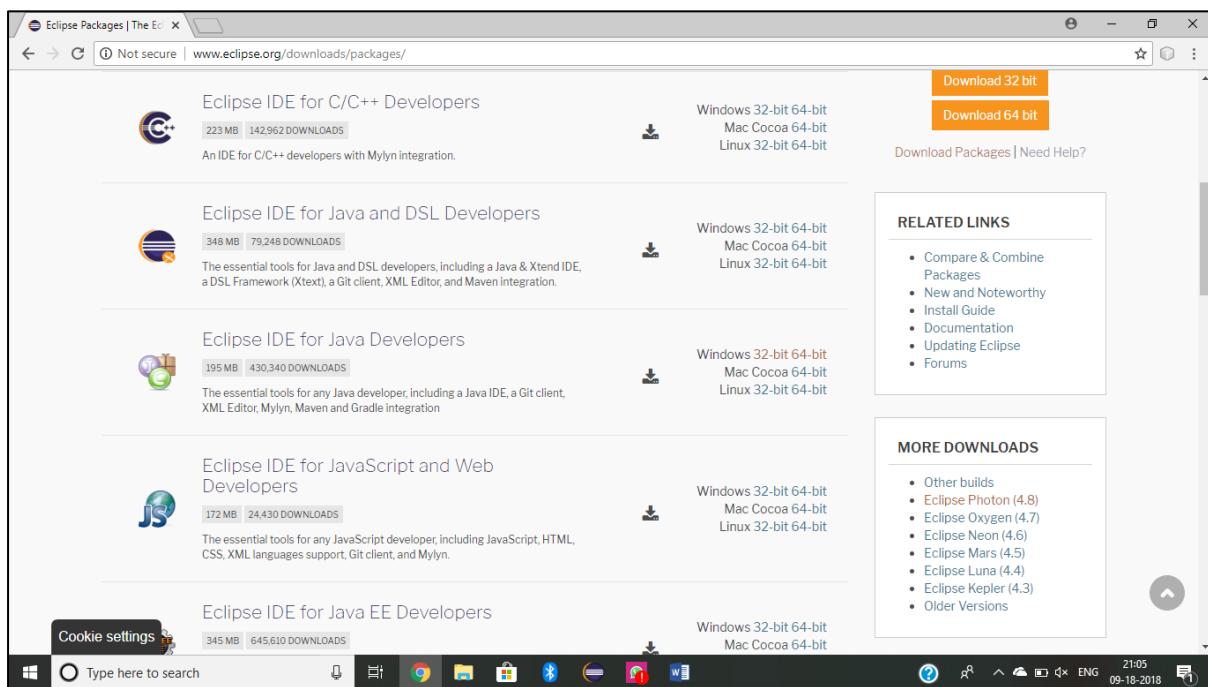
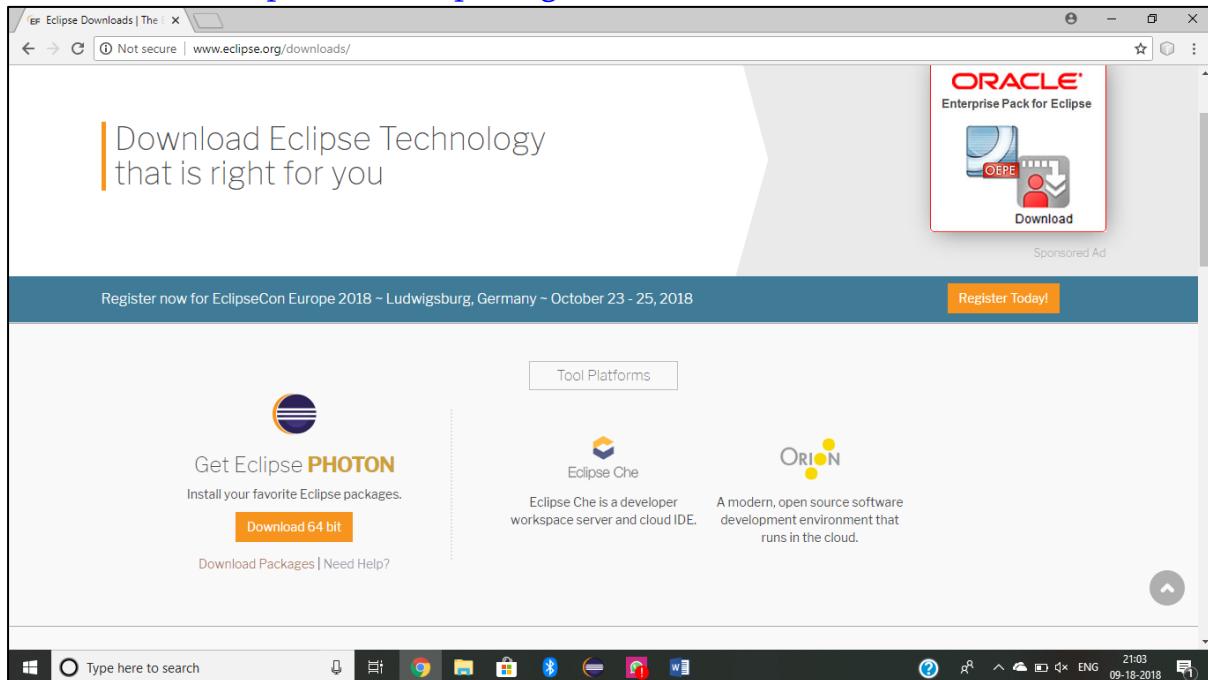
## Software Testing And Quality Assurance

### Practical 3

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

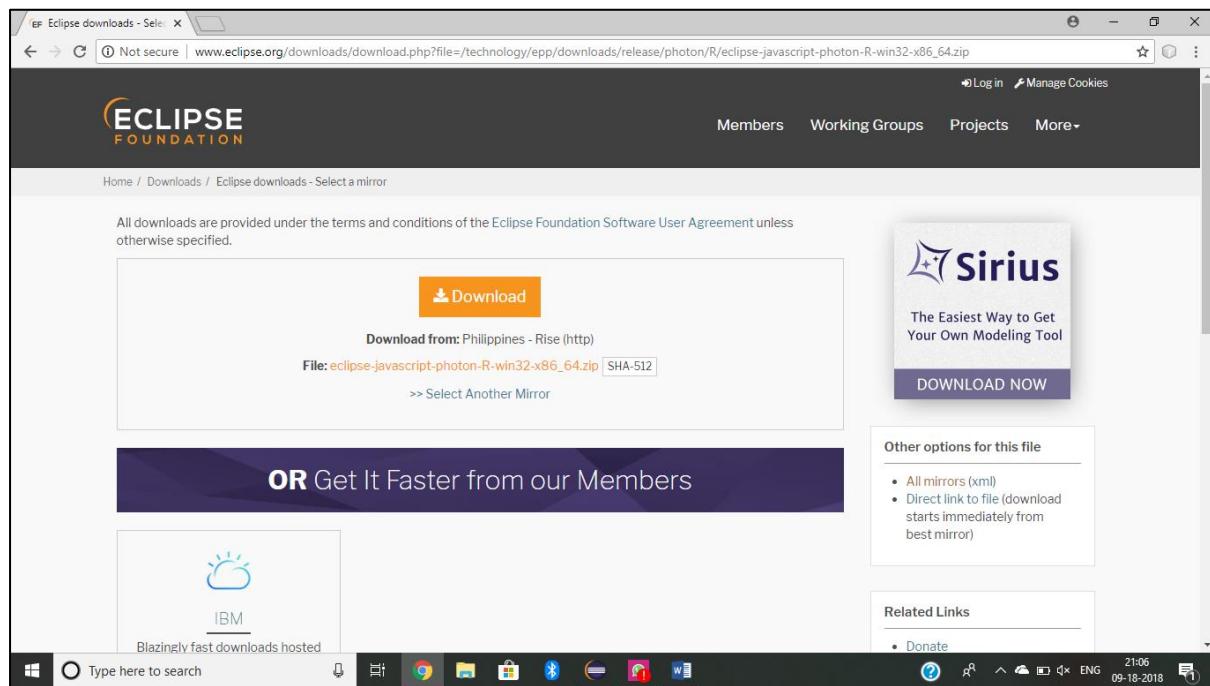
**Steps:**

1. Go to URL – <http://www.eclipse.org/downloads/>

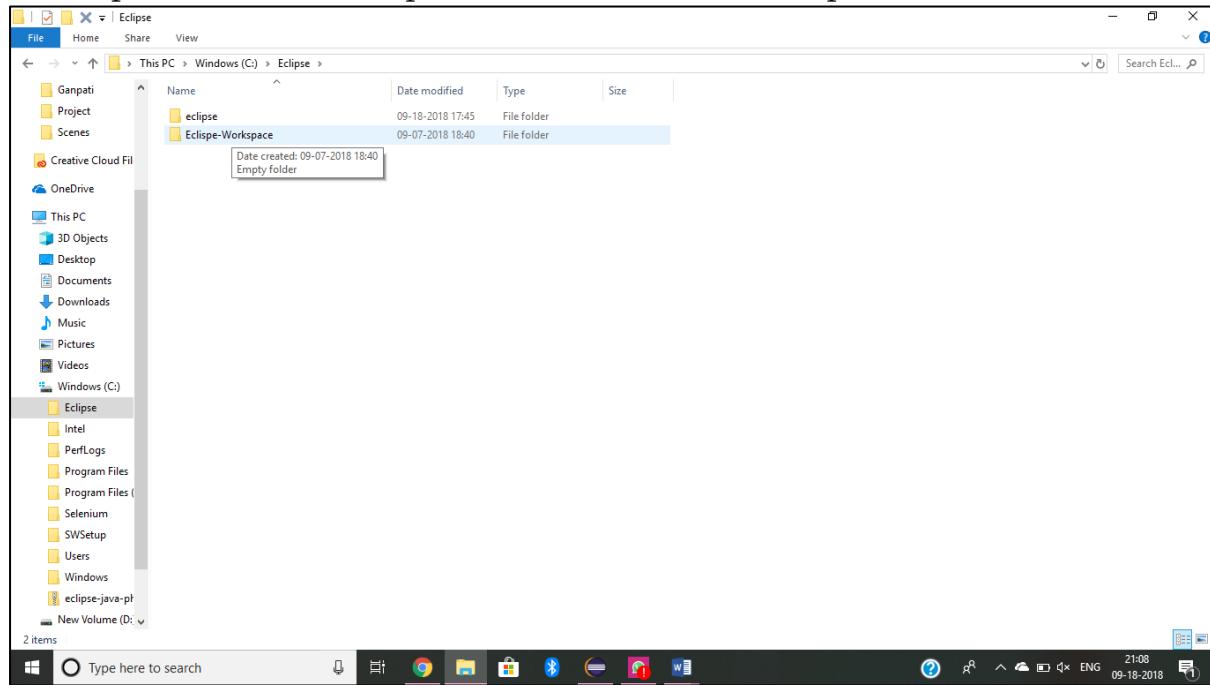


## Software Testing And Quality Assurance

### 2. Select Eclipse IDE for Java Developers (Click on Windows 64 bit platform)

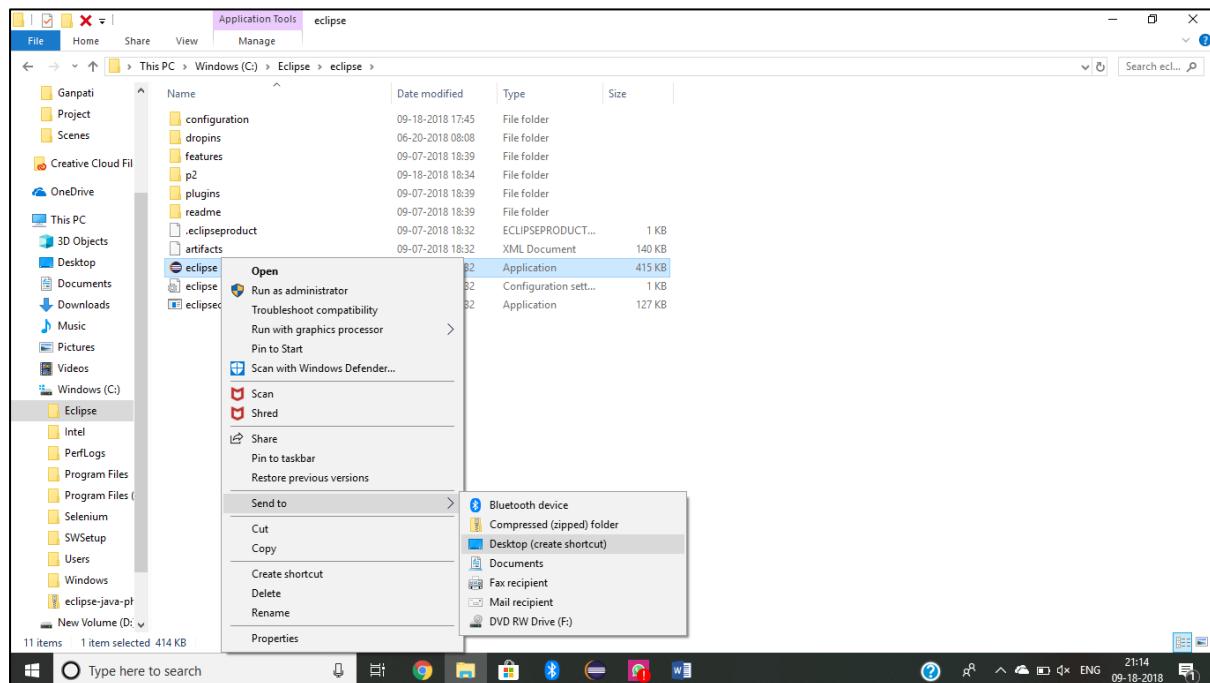


### 3. Click on OK button and save to a local drive (i.e. C: ) Unzip the downloaded zip file and rename that to Eclipse.

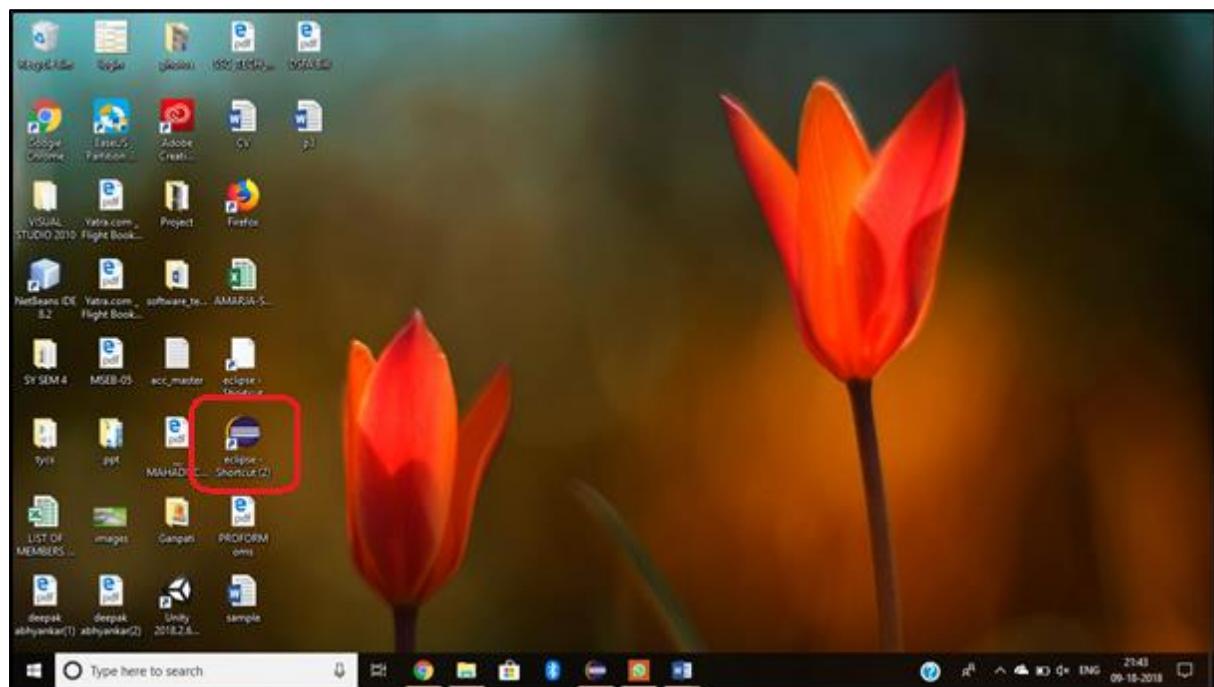


## Software Testing And Quality Assurance

4. Create one more folder “Eclipse-Workspace” (i.e. C:Eclipse-Workspace)in the same drive where Eclipse is unzipped and renamed.

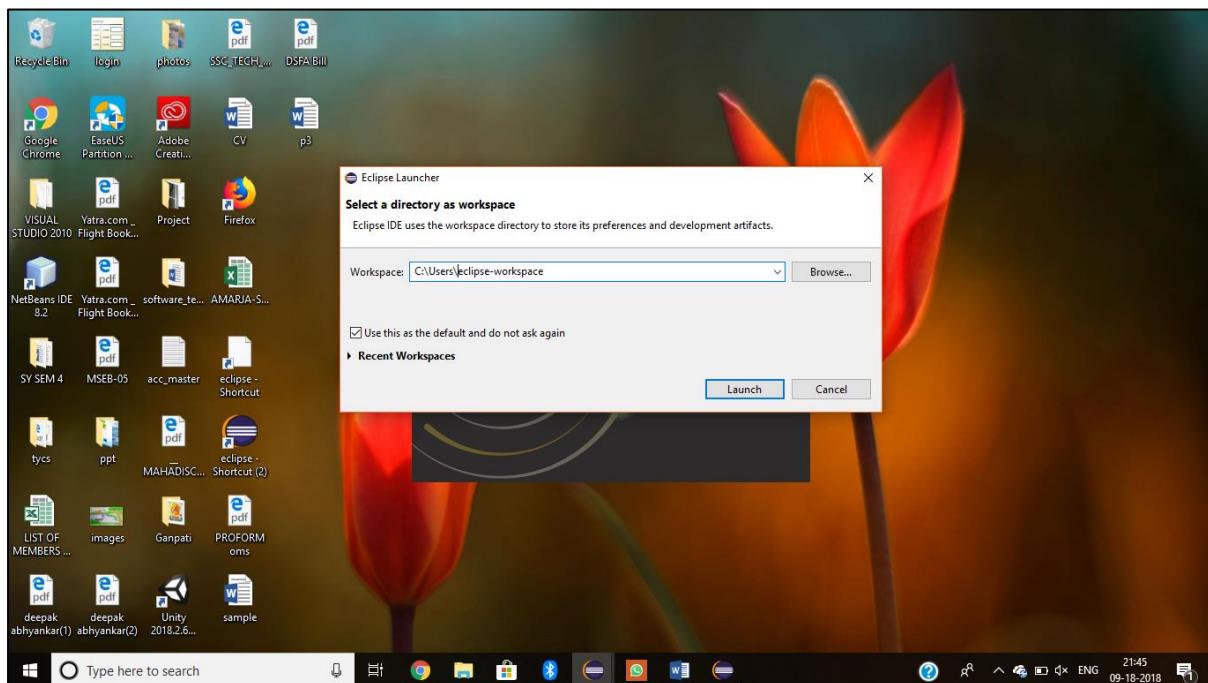
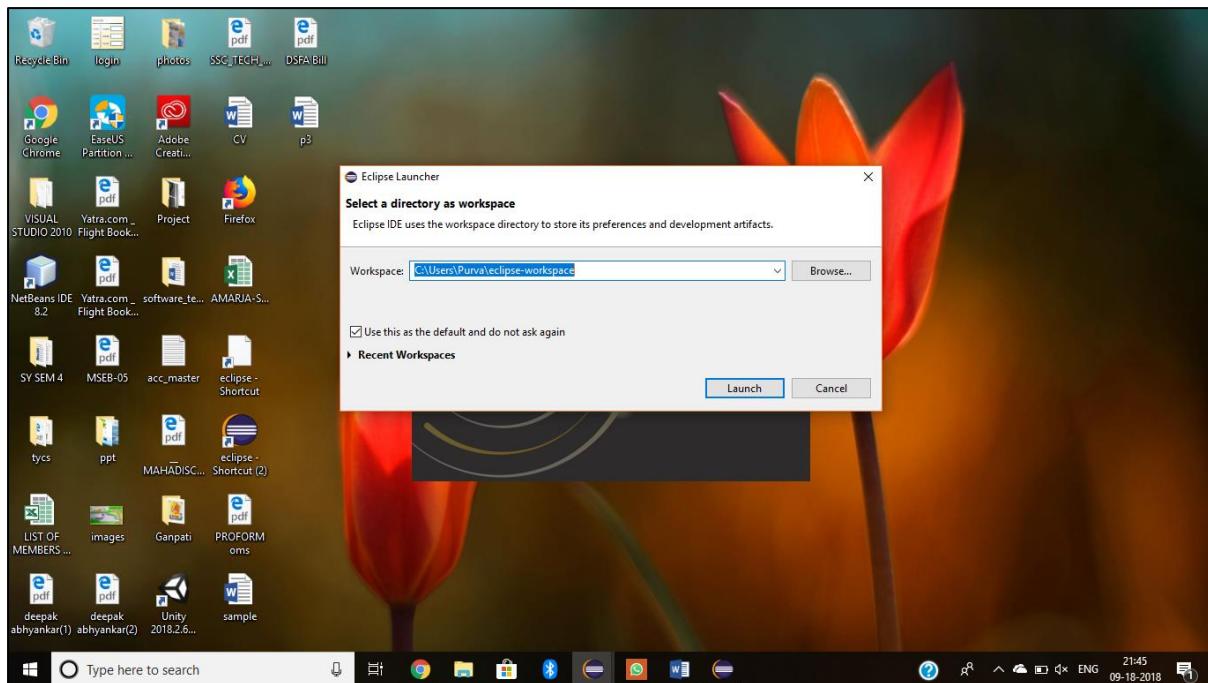


5. Create Eclipse desktop shortcut (go to C:Eclipse folder → right click Eclipse.exe and then click on “desktop create shortcut”).



## Software Testing And Quality Assurance

- Now we need to create a workspace folder ->  
C:Eclipse WorkspaceSeleniumTests .  
Double click on “Eclipse shortcut on Desktop”



## Software Testing And Quality Assurance

Download Selenium server: <http://seleniumhq.org/download/>

Download Selenium Client driver for Java (from Selenium Client Drivers section)

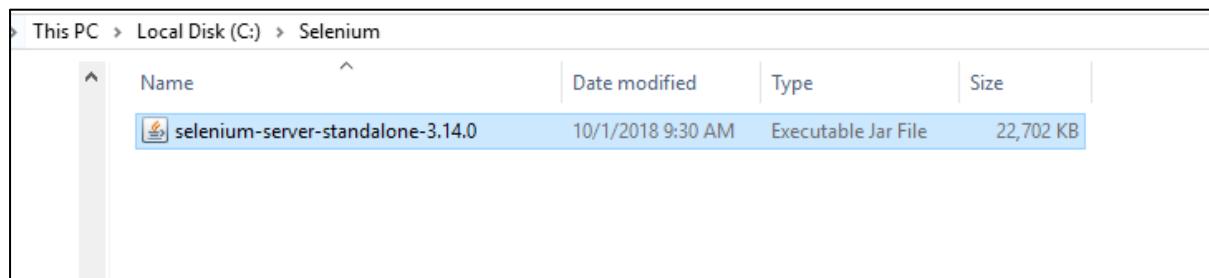
### Selenium Standalone Server

The Selenium Server is needed in order to run Remote Selenium WebDriver. Selenium 3.X is no longer capable of running Selenium RC directly, rather it does it through emulation and the WebDriverBackedSelenium interface.

Download version [3.14.0](#)

To run Selenium tests exported from the legacy IDE, use the [Selenium Html Runner](#).

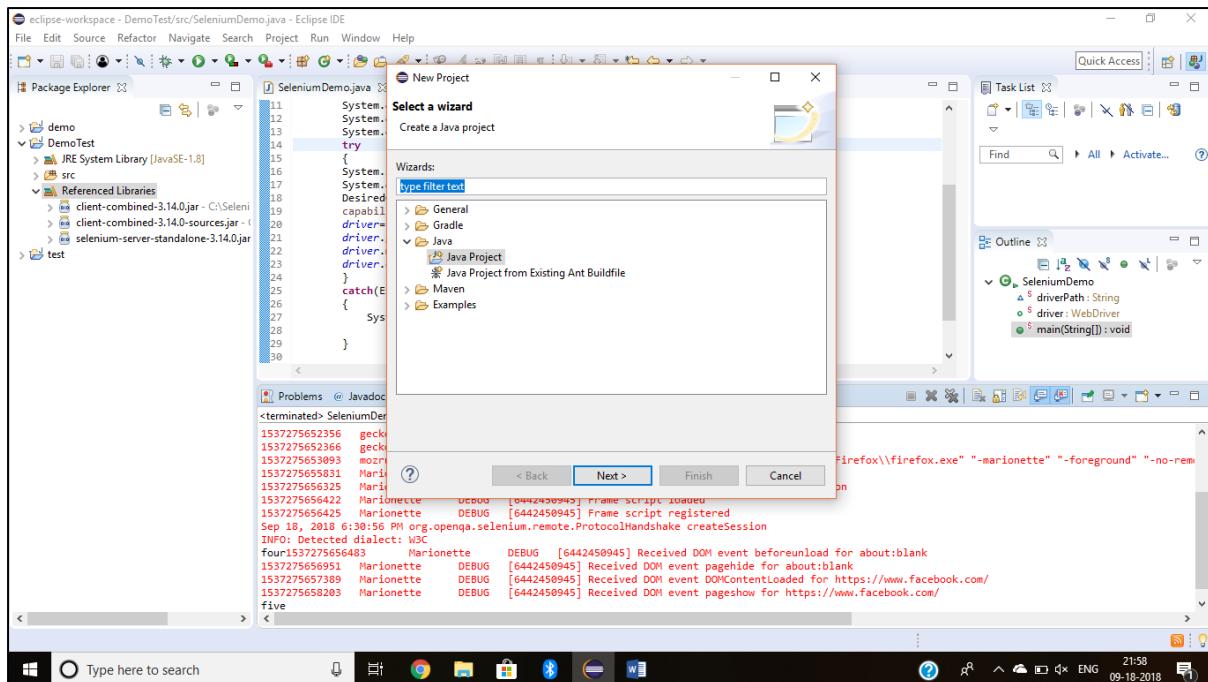
To use the Selenium Server in a Grid configuration [see the wiki page](#).



## Software Testing And Quality Assurance

Go to Eclipse → Click File → New → Project (from various options need to select just “project”)

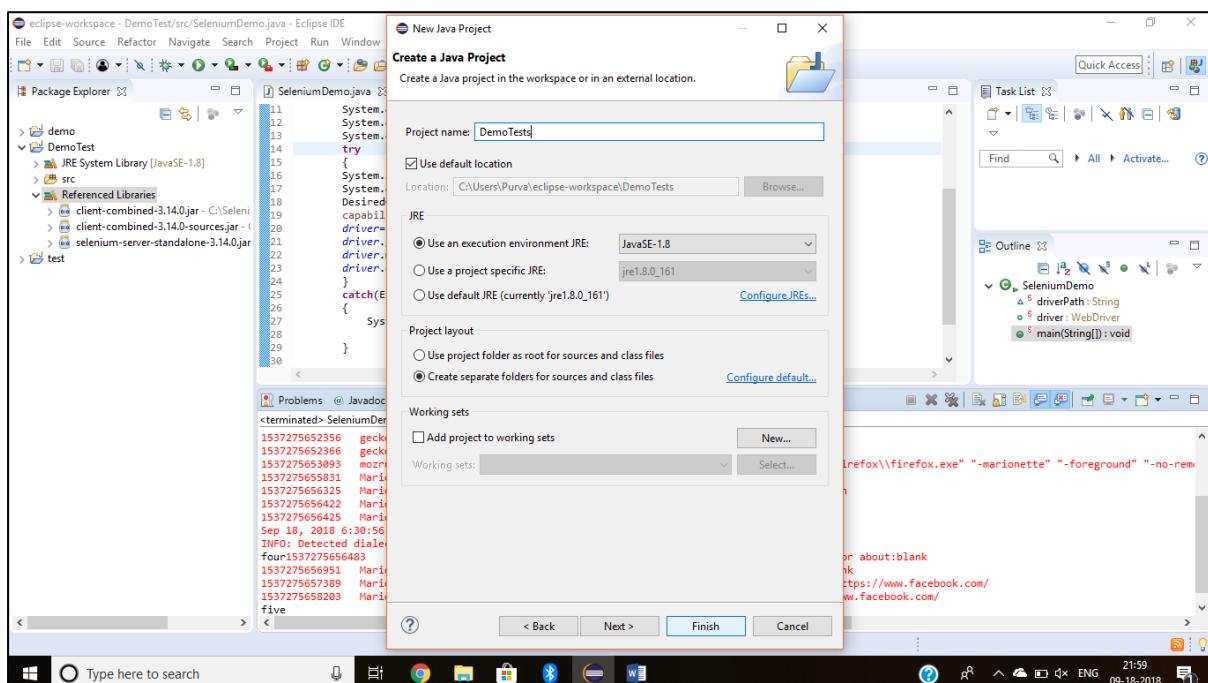
In Select Wizard → Click Java → “Java Project”



Now we are done with creation of project and need to configure the Selenium

Client driver to this Project

Right Click “DemoTests” project



## Software Testing And Quality Assurance

Click “Java Build Path”

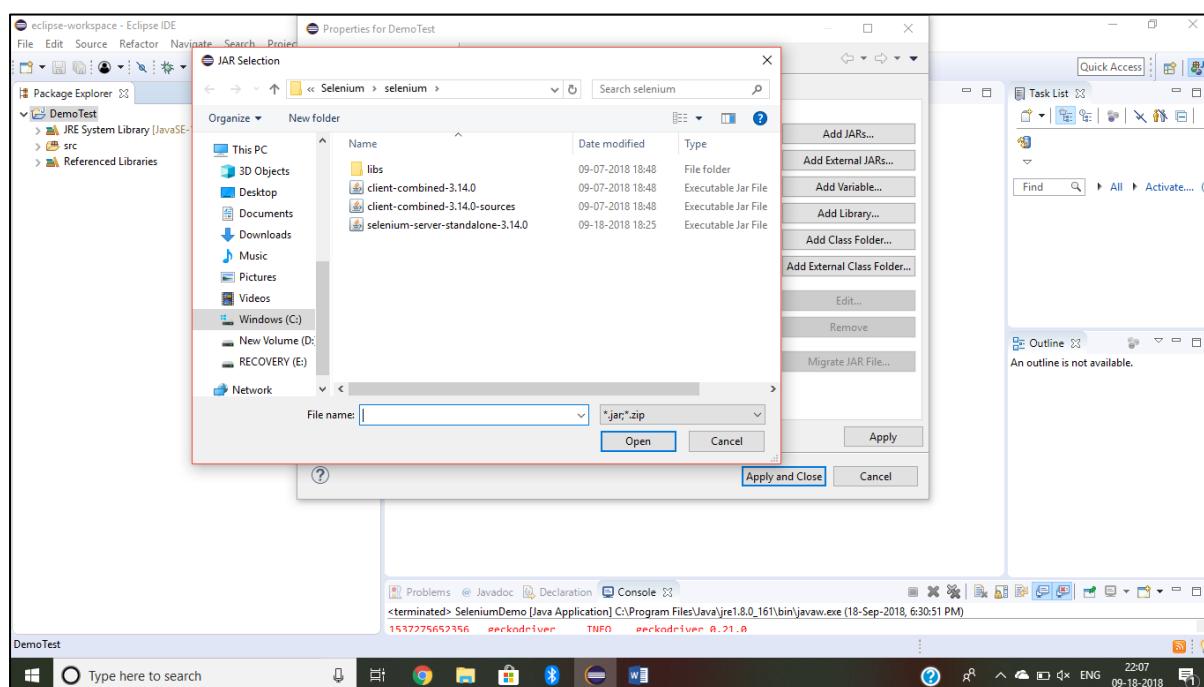
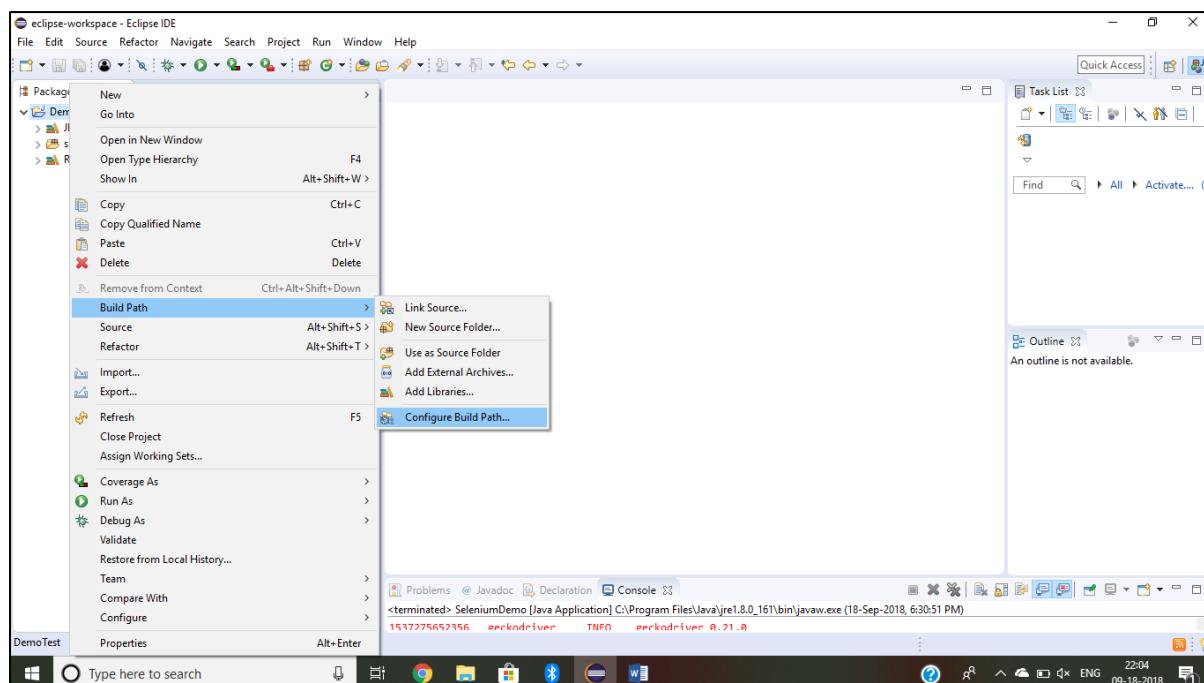
Click Libraries tab

Click “Add External JARs” button

Select “Selenium Client Drivers” unzipped in C:Selenium folder (Selenium

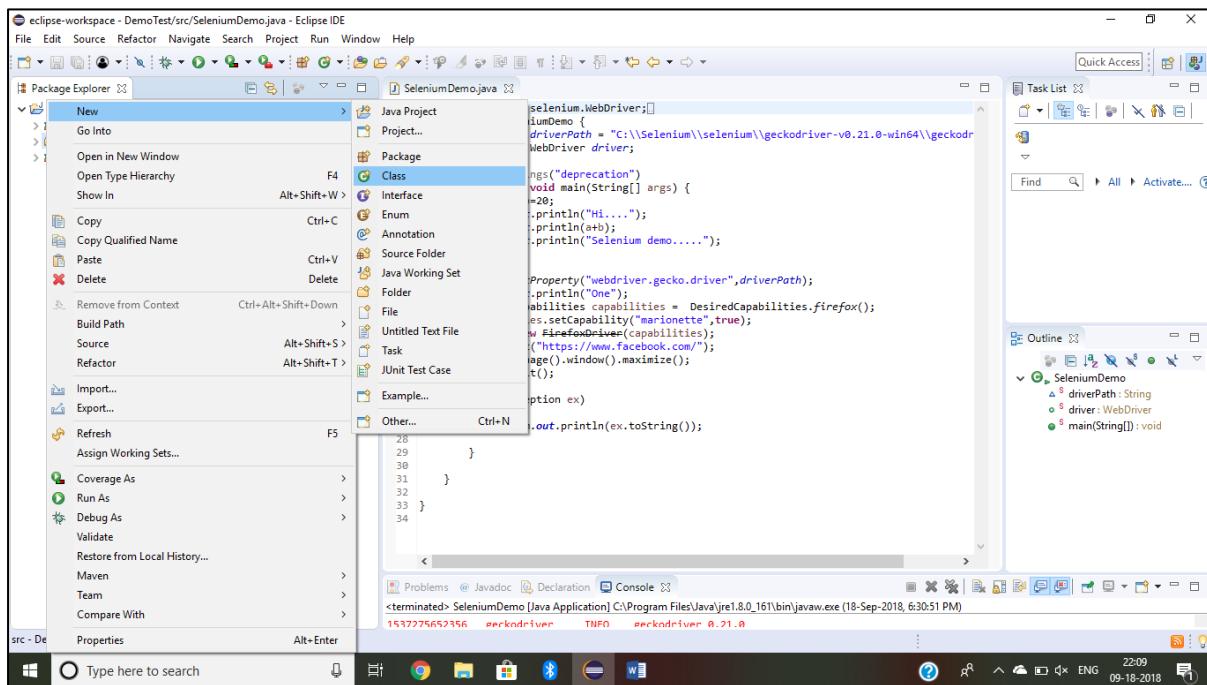
Server JAR file should not be added)

Click OK

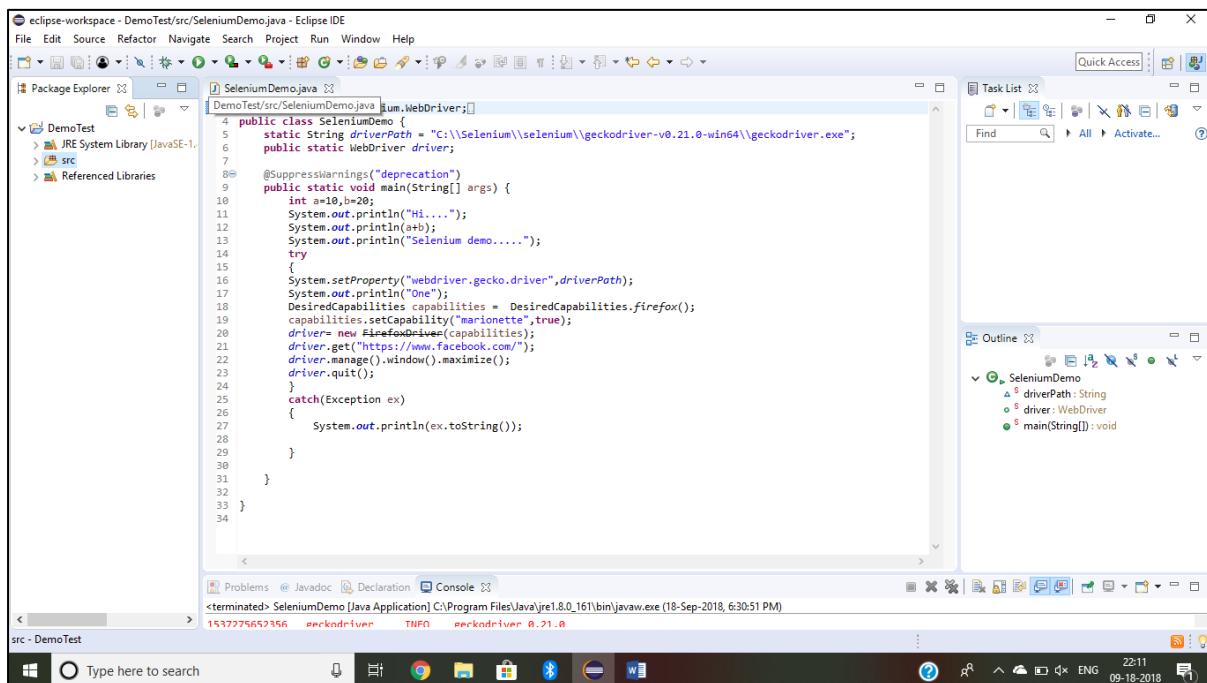


## Software Testing And Quality Assurance

Create a new class file as “SeleniumDemo” in the “DemoTest” by right click on src folder.



### Code



# **Software Testing And Quality Assurance**

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the `SeleniumDemo` project with files `SeleniumDemo.java` and `src`.
- SeleniumDemo.java Content:** The Java code initializes a WebDriver instance using geckodriver.
- Task List:** Shows the outline of the class `SeleniumDemo` with methods `main`, `driverPath`, and `driver`.
- Console Output:** Displays the terminal output of the application's execution, showing the driver path and the message "Selenium demo.....".

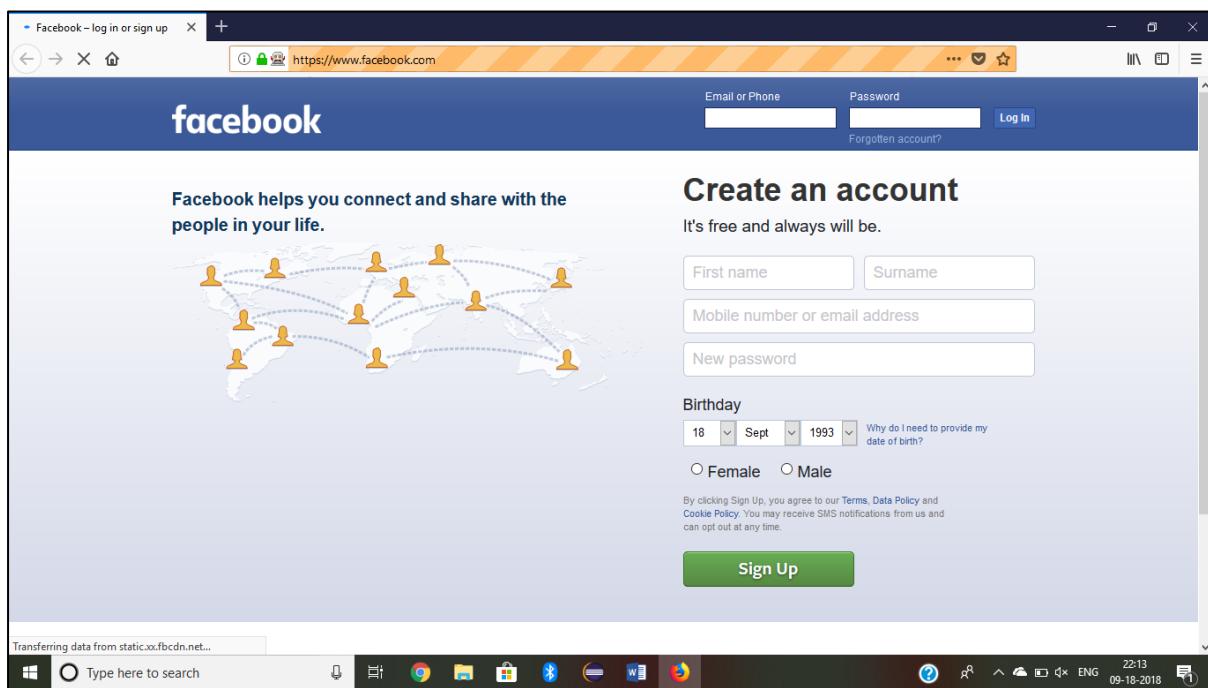
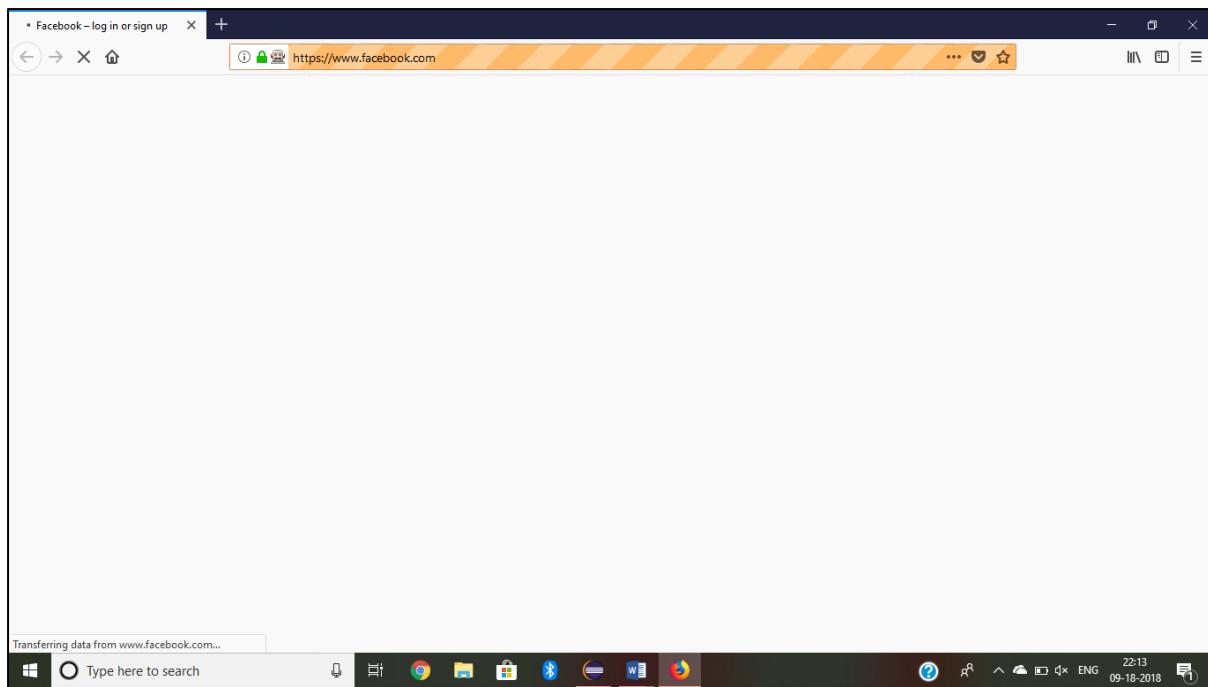
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project "DemoTest" with its source code files.
- SeleniumDemo.java:** The code is as follows:

```
11 System.out.println("Hi....");
12 System.out.printin(a+b);
13 System.out.printin("Selenium demo.....");
14 try
15 {
16     System.setProperty("webdriver.gecko.driver",driverPath);
17     System.out.println("One");
18     DesiredCapabilities capabilities = DesiredCapabilities.firefox();
19     capabilities.setCapability("marionette",true);
20     driver = new FirefoxDriver(capabilities);
21     driver.get("https://www.facebook.com/");
22     driver.manage().window().maximize();
23     driver.quit();
24 }
25 catch(Exception ex)
26 {
27     System.out.println(ex.toString());
28 }
29 }
```

- Task List:** Shows the tasks related to the current file.
- Outline:** Shows the class structure with fields and methods.
- Terminal:** A terminal window titled "terminated: SeleniumDemo [Java Application] C:\Program Files\Java\jre1.8.0\_161\bin\javaw.exe (18-Sep-2018, 6:30:51 PM)" displaying Java application logs.

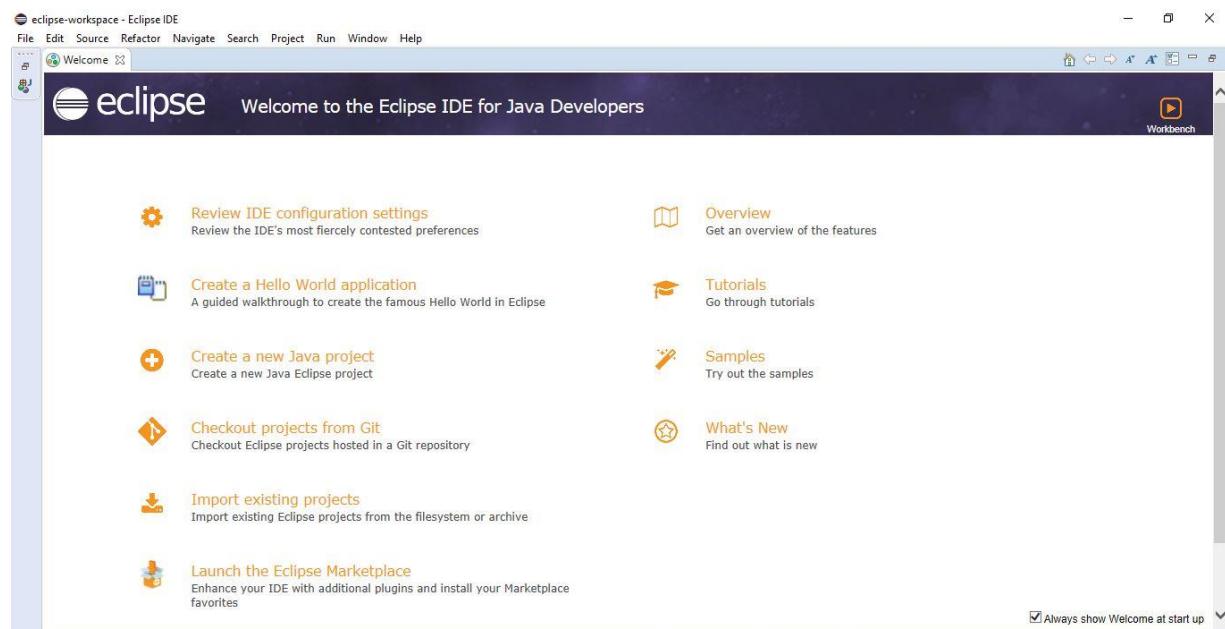
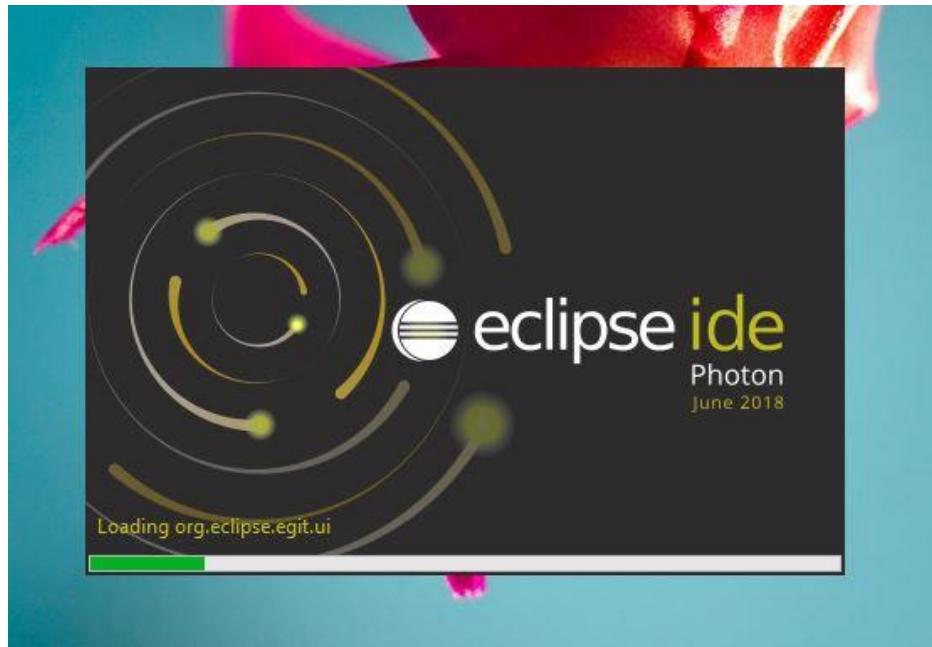
## Software Testing And Quality Assurance



## Practical No : 4

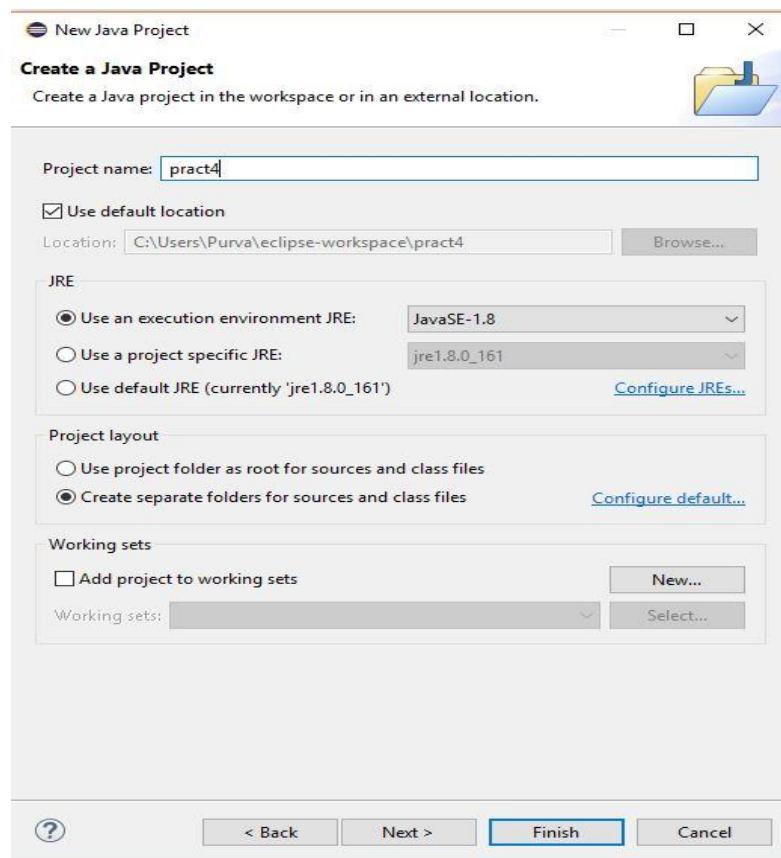
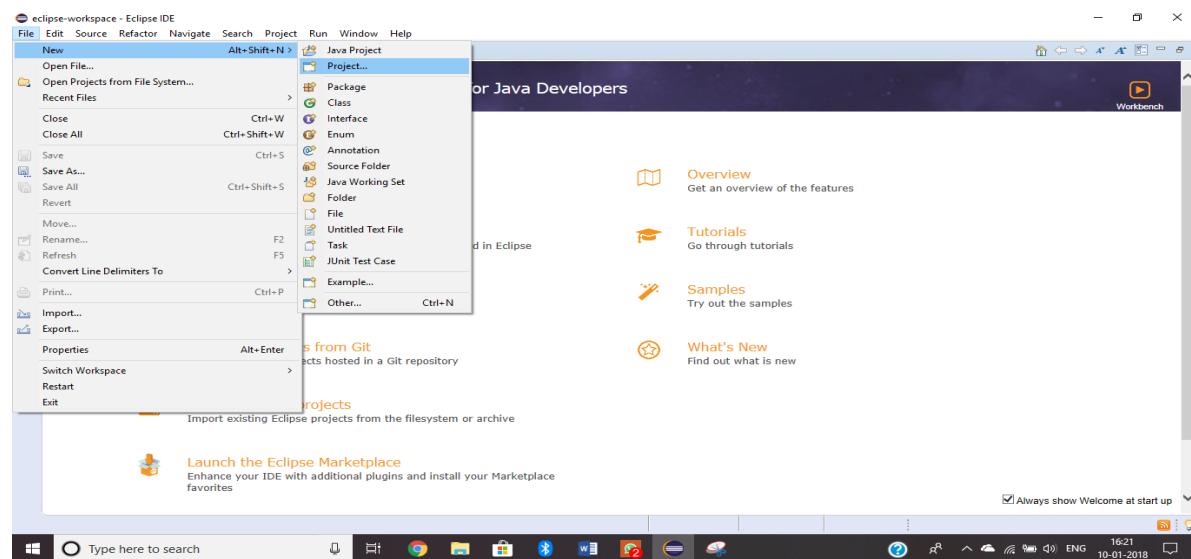
Aim: Write and test a program to login a specific web page.  
(Using JUnit)

### 1. Open Eclipse IDE

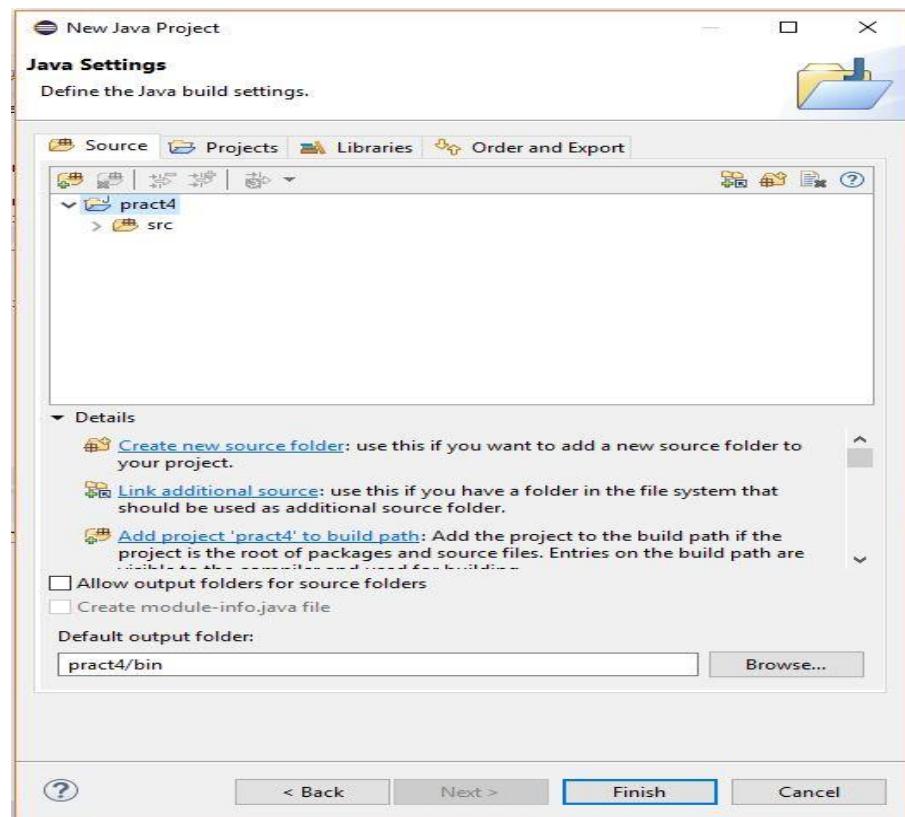


## Software Testing And Quality Assurance

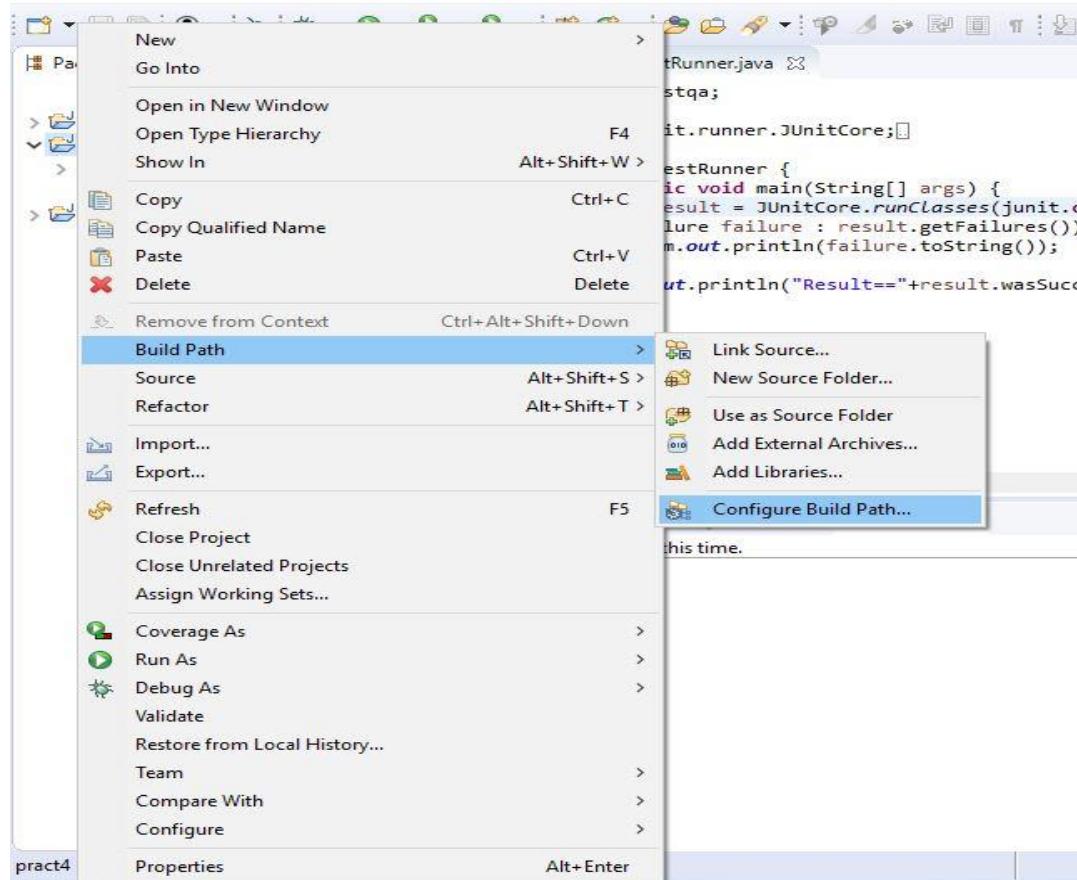
### 2. Open New Project



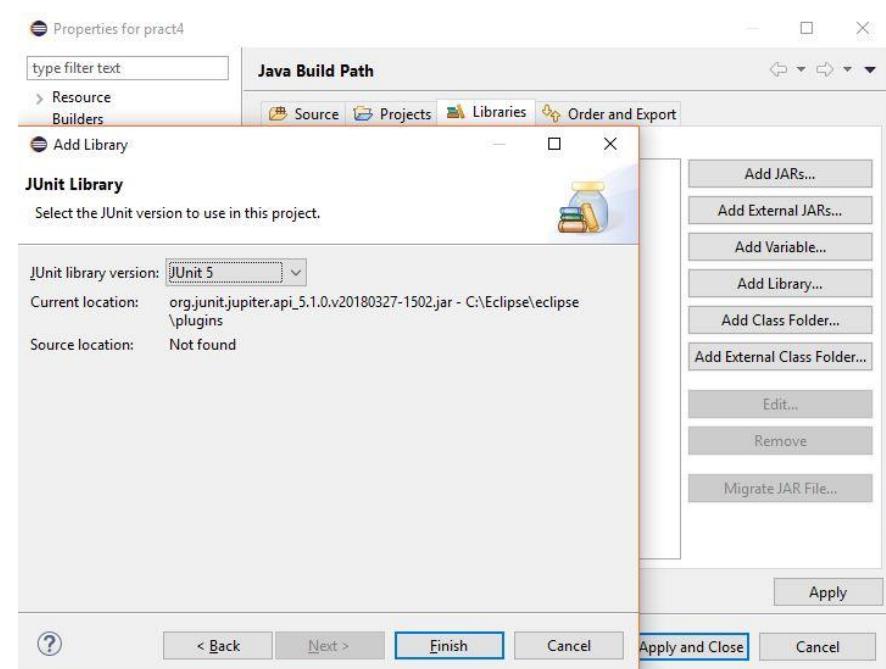
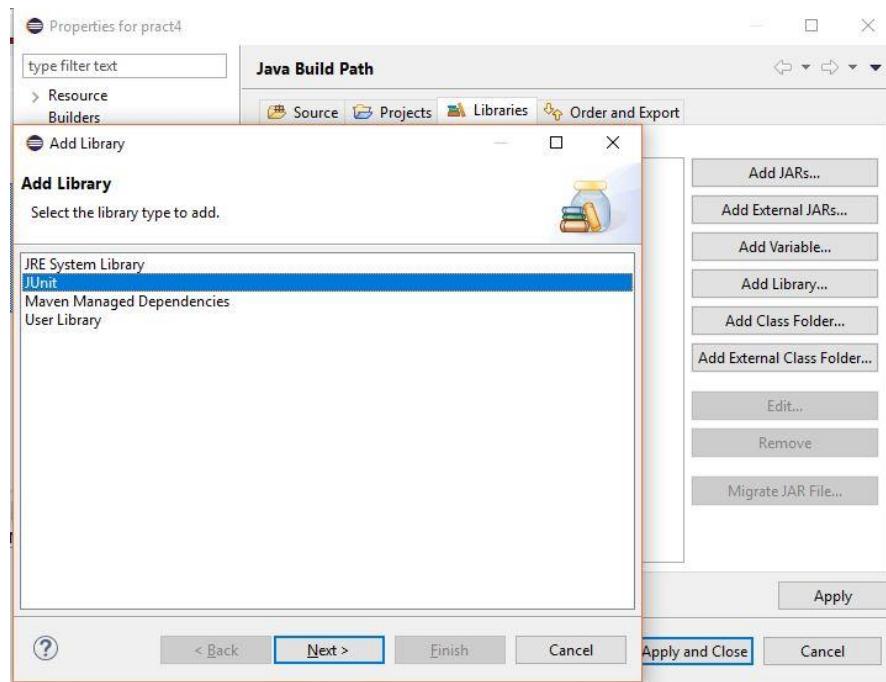
## Software Testing And Quality Assurance



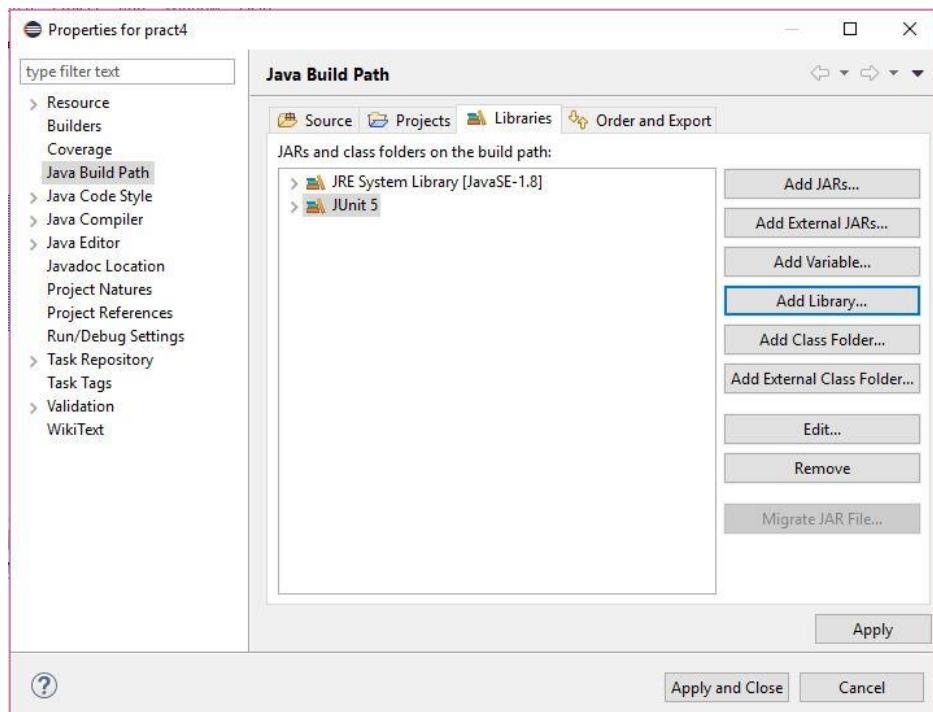
### 3. Add Junit jar file



## Software Testing And Quality Assurance



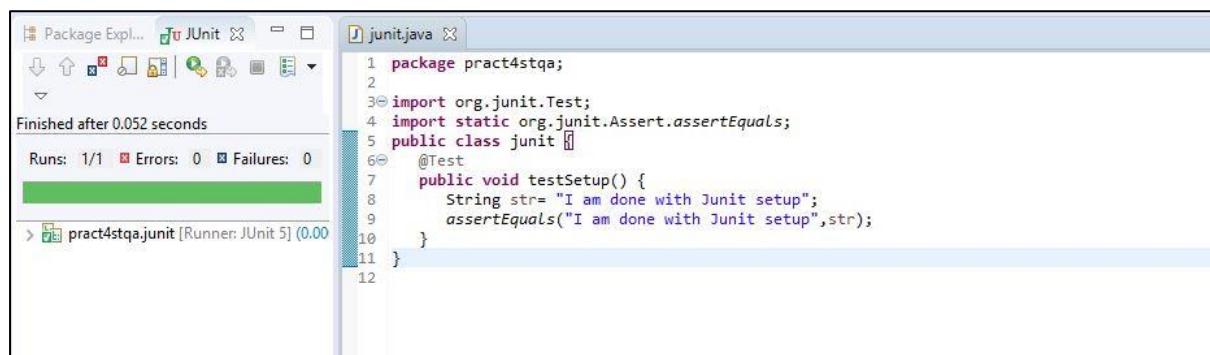
## Software Testing And Quality Assurance



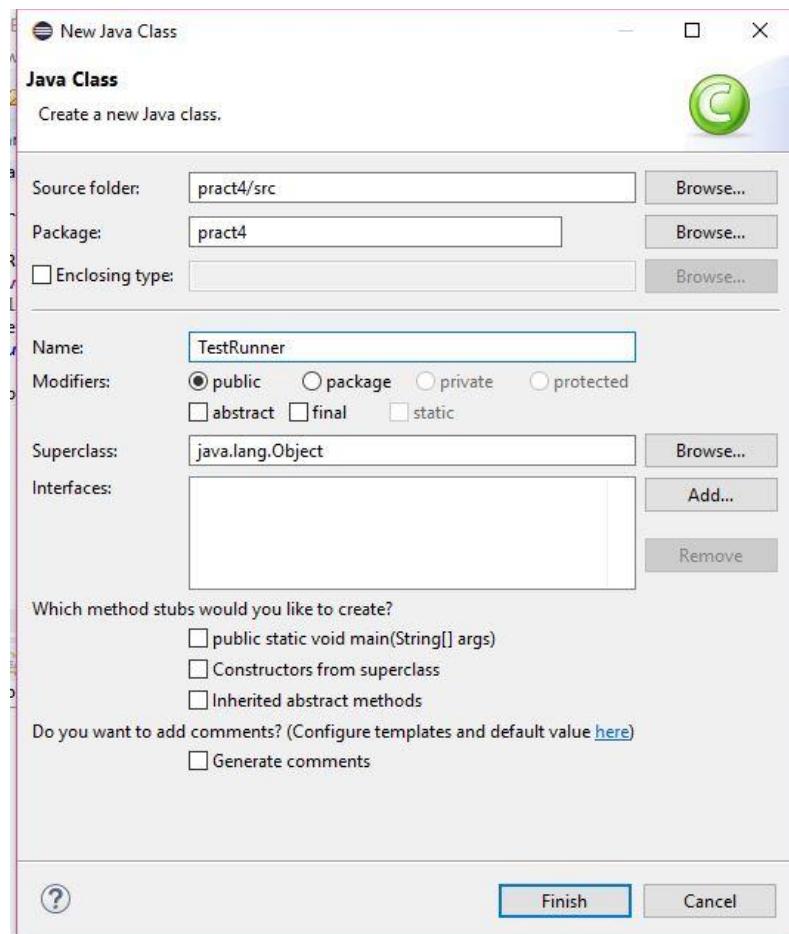
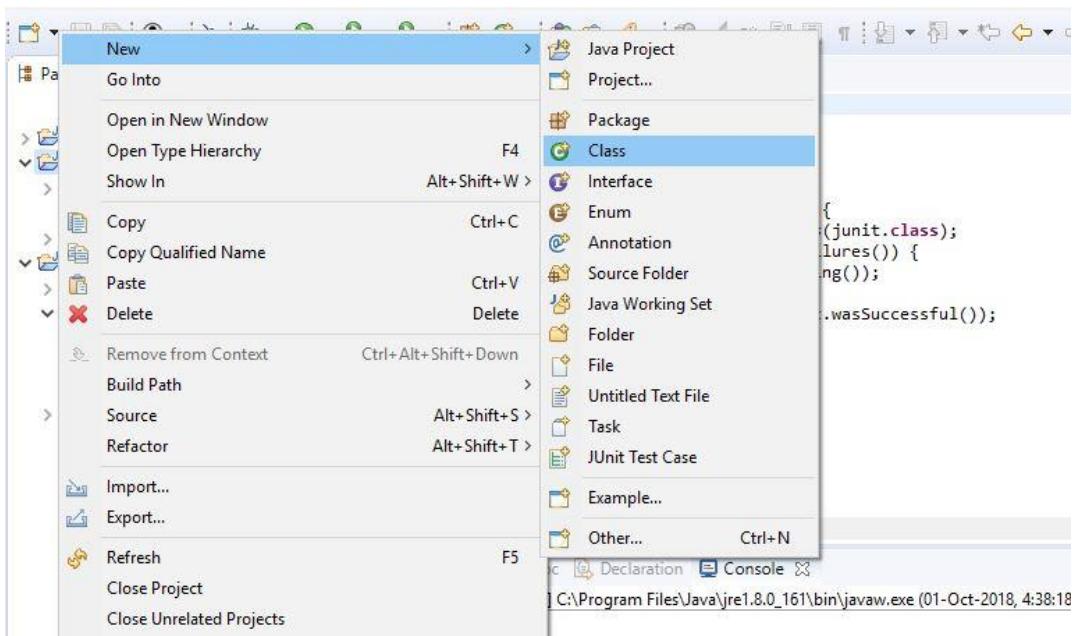
### 4. Add New Class File by name junit

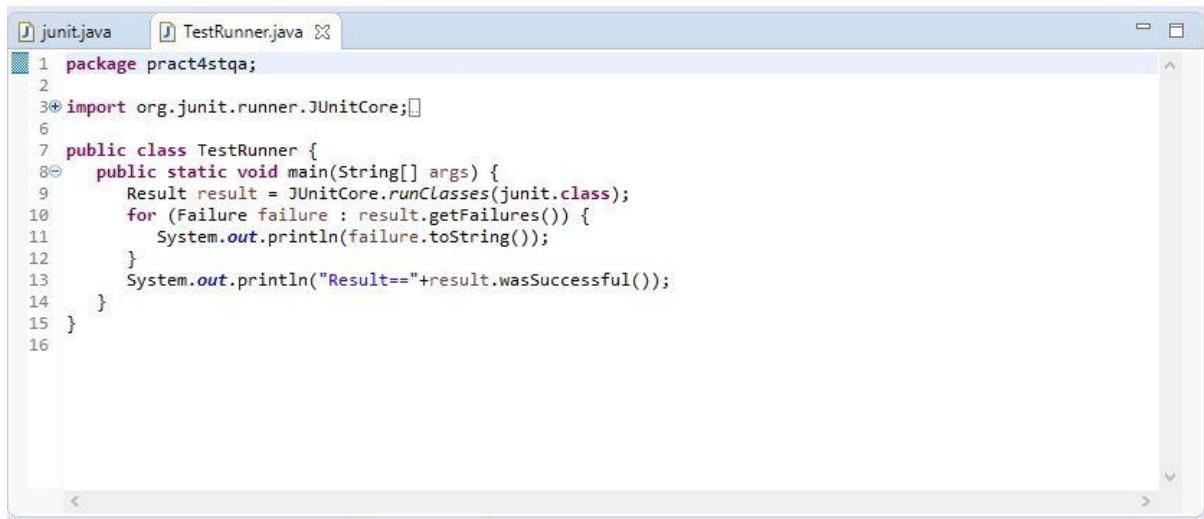
```
1 package pract4stqa;
2
3 import org.junit.Test;
4 import static org.junit.Assert.assertEquals;
5 public class junit {
6     @Test
7     public void testSetup() {
8         String str= "I am done with Junit setup";
9         assertEquals("I am done with Junit setup",str);
10    }
11 }
```

### 5. Run the file junit to check junit is installed



### 6. Add new class file name TestRunner





A screenshot of a Java code editor window. The title bar shows two tabs: "junit.java" and "TestRunner.java". The "TestRunner.java" tab is active. The code in the editor is:

```
1 package pract4stqa;
2
3 import org.junit.runner.JUnitCore;
4
5 public class TestRunner {
6     public static void main(String[] args) {
7         Result result = JUnitCore.runClasses(junit.class);
8         for (Failure failure : result.getFailures()) {
9             System.out.println(failure.toString());
10        }
11        System.out.println("Result=="+result.wasSuccessful());
12    }
13 }
14
15 }
```

### 7. Run the program Test Runner



A screenshot of a Java code editor window with tabs for "junit.java" and "TestRunner.java". The "TestRunner.java" tab is active. Below the editor is a "Console" tab showing the output of the program's execution.

```
1 package pract4stqa;
2
3 import org.junit.runner.JUnitCore;
4
5 public class TestRunner {
6     public static void main(String[] args) {
7         Result result = JUnitCore.runClasses(junit.class);
8         for (Failure failure : result.getFailures()) {
9             System.out.println(failure.toString());
10        }
11        System.out.println("Result=="+result.wasSuccessful());
12    }
13 }
14
15 }
```

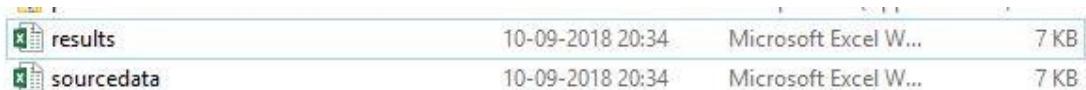
The "Console" tab displays the following output:

```
<terminated> TestRunner [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (01-Oct-2018, 4:41:48 PM)
Result==true
```

## **Practical No : 5**

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

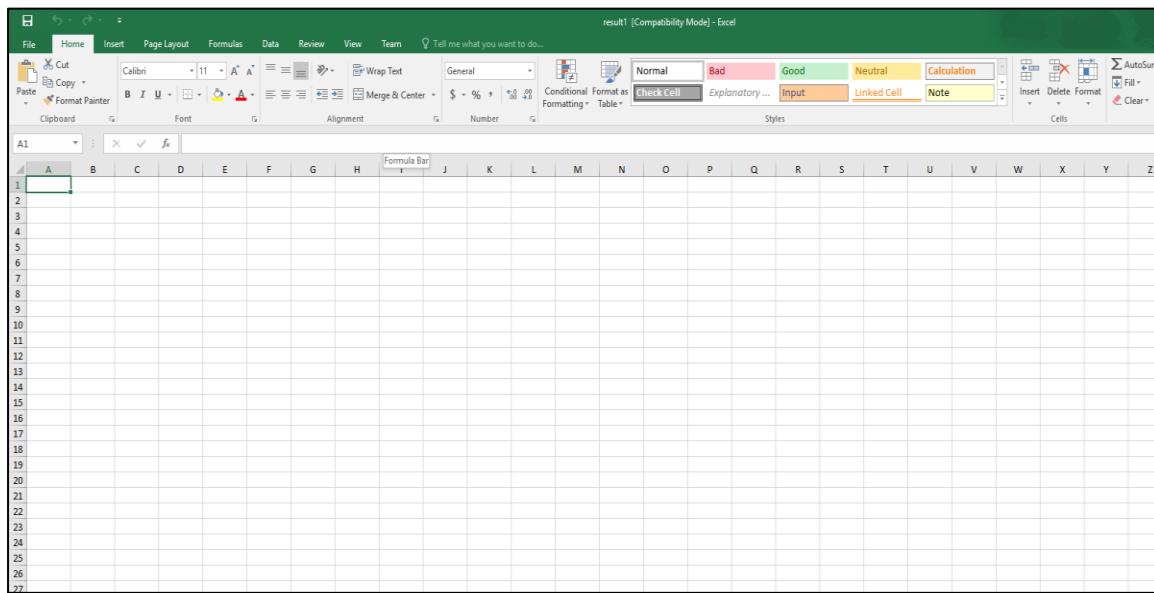
1. Create 2 files 'sourcedata' and 'results'



2. Insert data in 'sourcedata' file

	A	B	C	D	E	F	G
1	roll	name	x	y	z	total	
2	1	a	60	70	65	195	
3	2	s	25	23	54	102	
4	3	d	3	43	4	50	
5	4	f	2	54	23	79	
6	5	c	45	76	54	175	
7	6	d	3	43	21	67	
8	7	s	5	23	21	49	
9							
10							

### 3. Empty file for results .



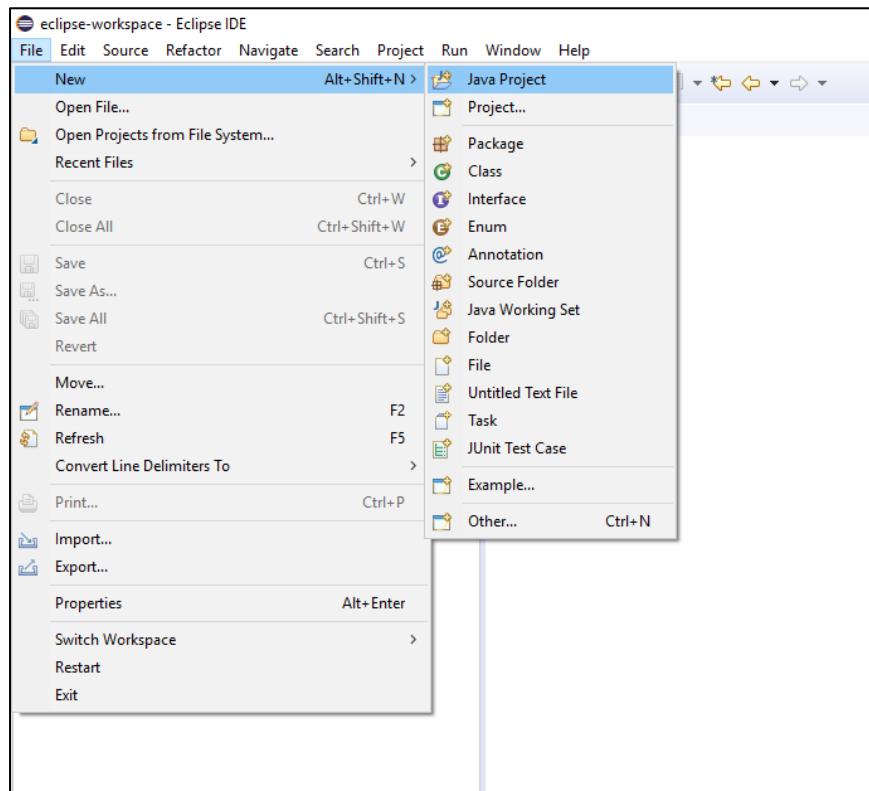
### 4. Save As Both the files by 'Excel 97-2013 WorkBook'

The screenshot shows the 'Save As' dialog box in Microsoft Excel. The left sidebar has options like Info, New, Open, Save, Save As, History, Print, Share, Export, Publish, Close, Account, Feedback, and Options. The main area shows 'Save As' with 'Recent' and 'This PC' selected. The 'Save as type' dropdown is open, showing various file formats. The 'Excel Workbook' option is highlighted. The 'File name:' field contains 'Purva' and the 'Save as type:' dropdown also has 'Excel Workbook' selected. At the bottom are 'Save' and 'Cancel' buttons.

File Name	Modified	Type	Size
results	10-09-2018 20:36	Microsoft Excel 97...	25 KB
results	10-09-2018 20:34	Microsoft Excel W...	7 KB
sourcedata	10-09-2018 20:34	Microsoft Excel W...	7 KB
sourcedata.xlsx	10-09-2018 20:36	Microsoft Excel 97...	25 KB

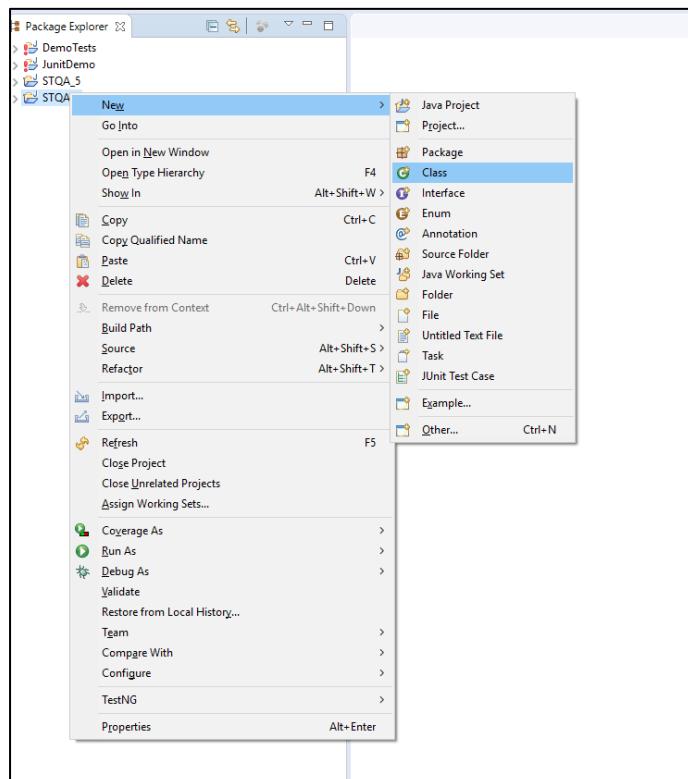
## Software Testing And Quality Assurance

### 5. Open new project in Eclipse.

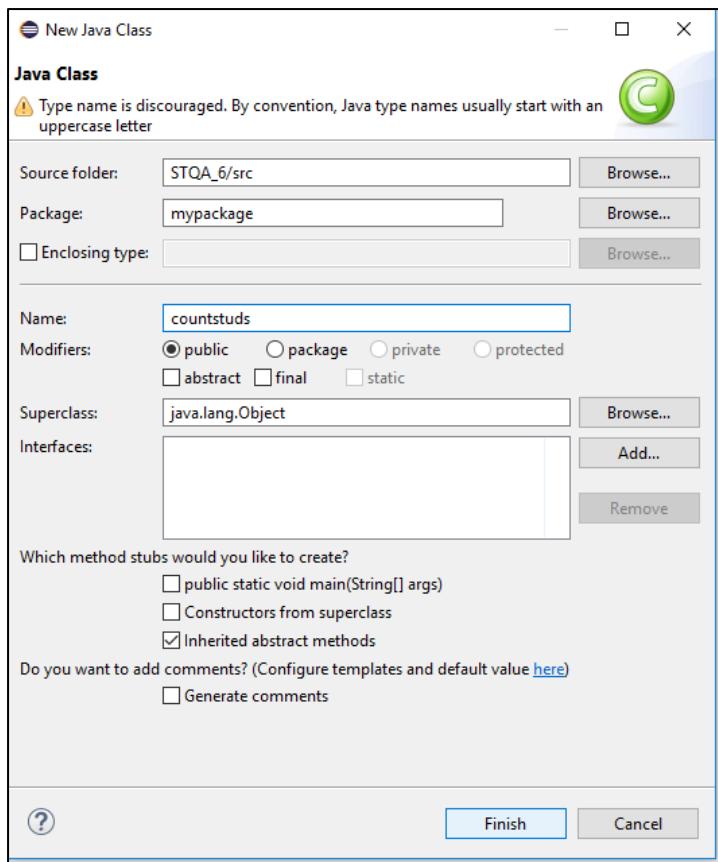


Name It As 'STQApract'

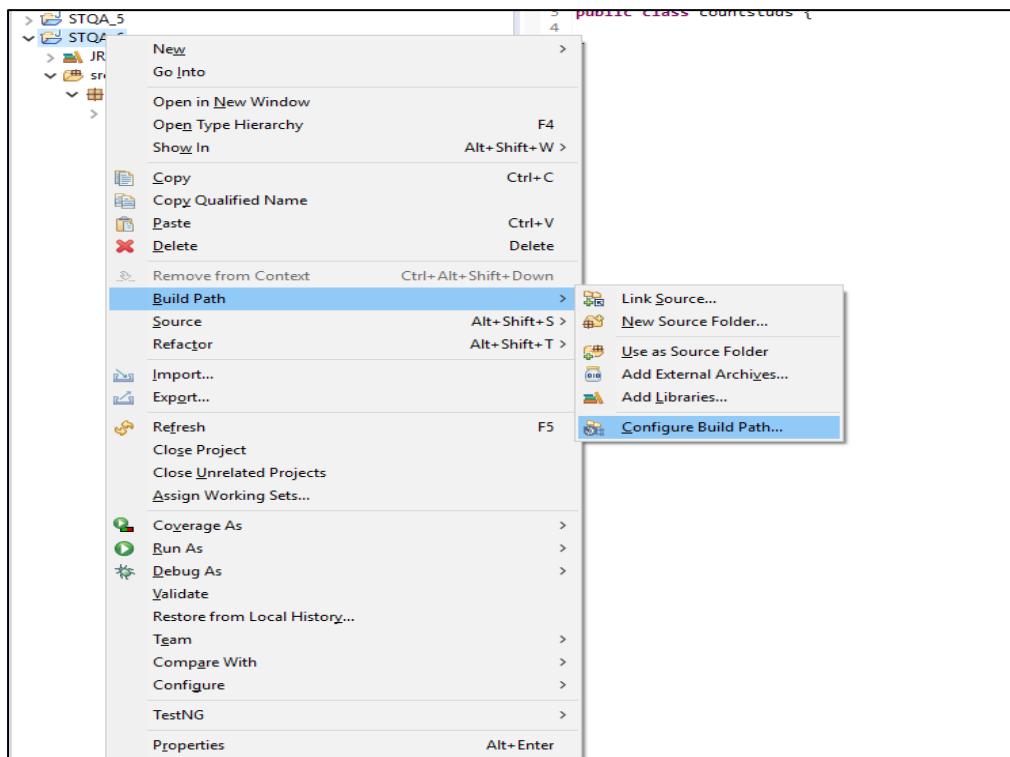
### 6. Add new Class file name it as countstuds



## Software Testing And Quality Assurance

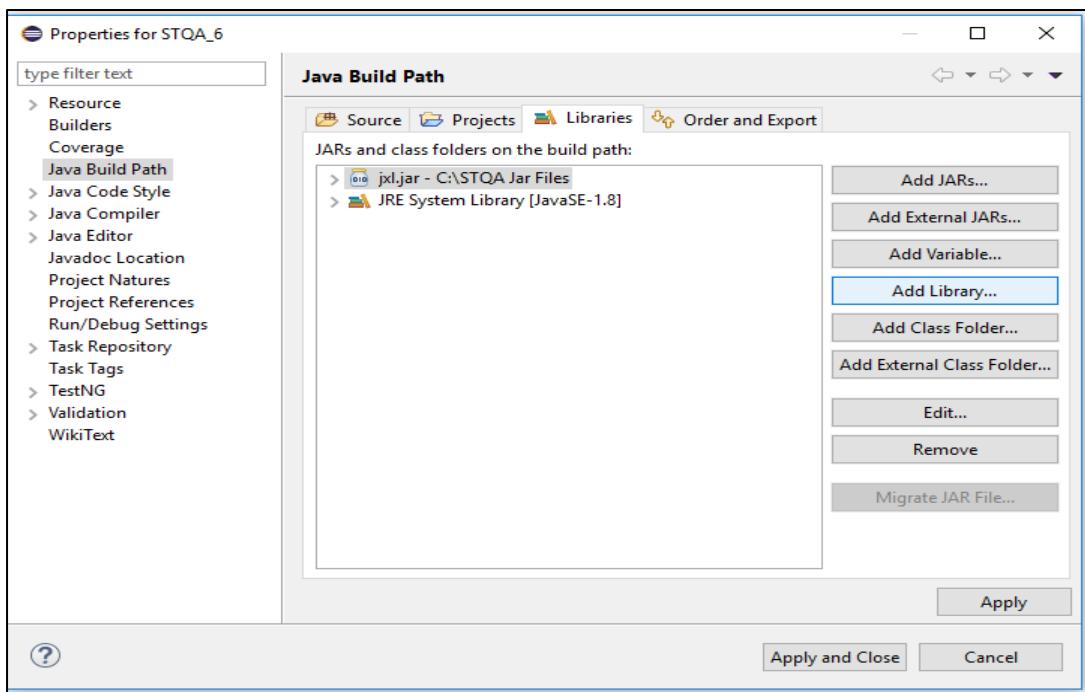
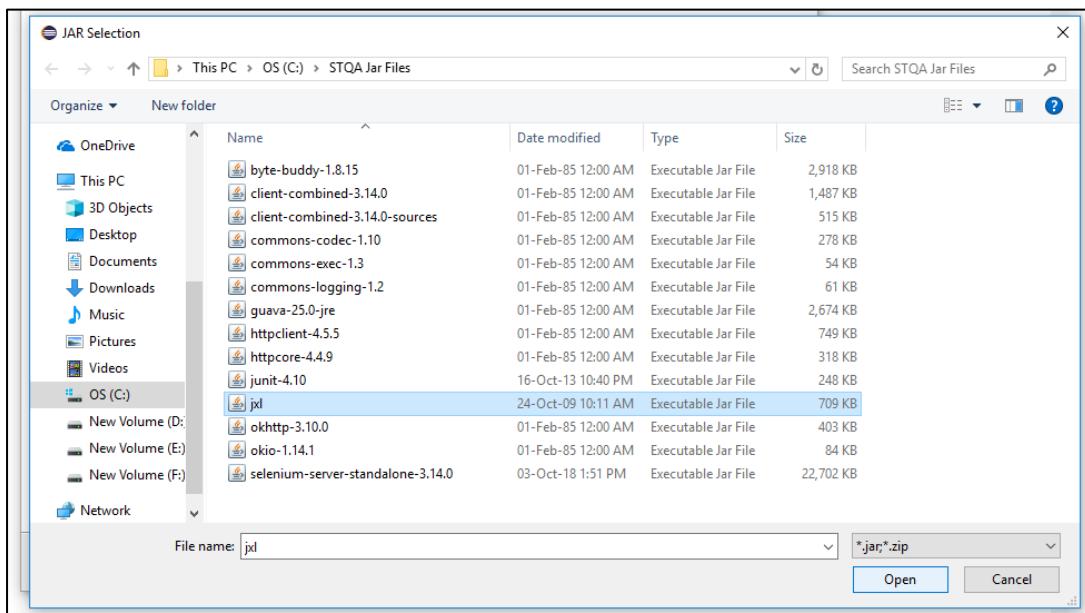


## 7. Add Configure Files in Build Path

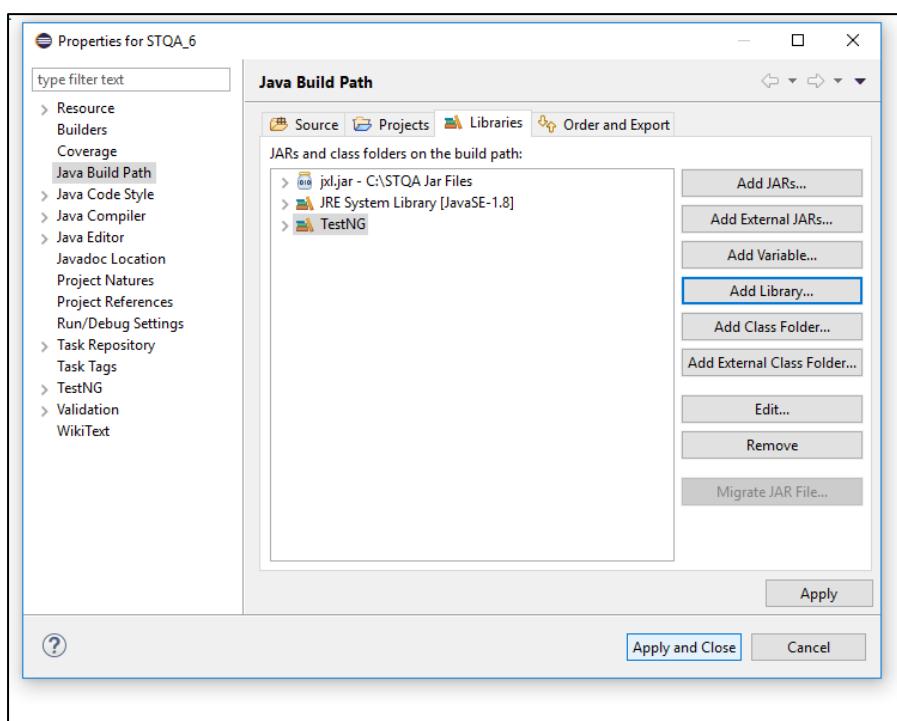
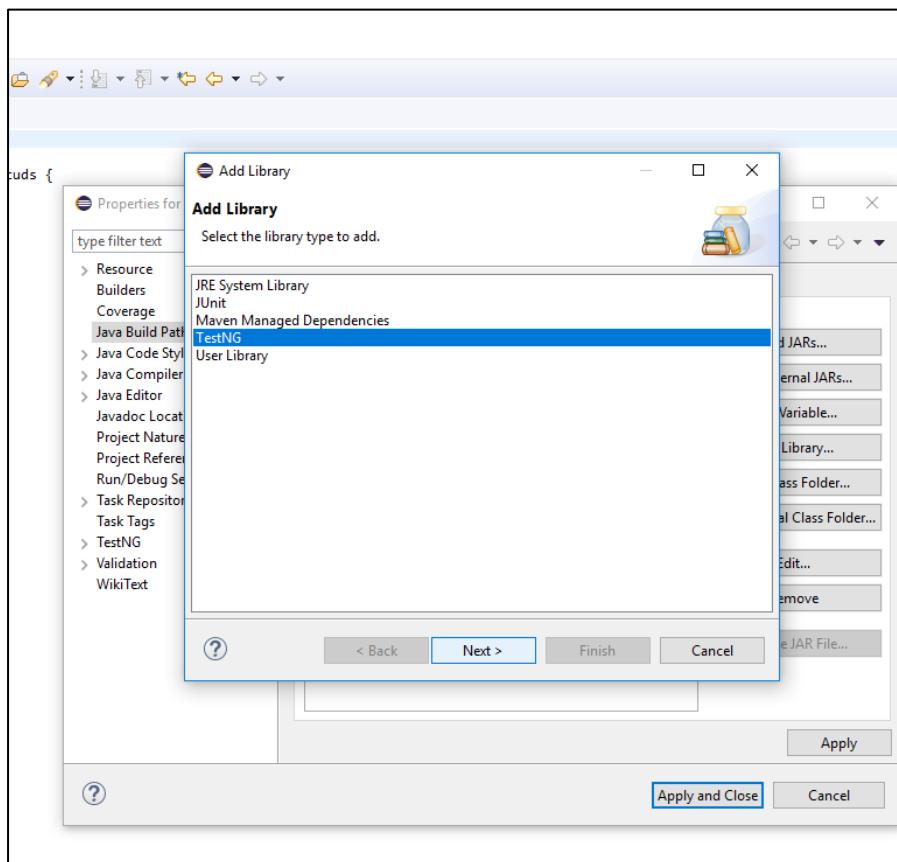


## Software Testing And Quality Assurance

### 8. Add External Jar files jxl.exe



### 9. Add New Library TestNG



### **10. Write Code In class file countstuds**

```
package p5;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import jxl.*;

import jxl.read.*;

import jxl.write.*;

import java.io.*;

public class countstuds {

    @BeforeClass
    public void f1()
    {}

    @Test
    public void testImportexport1() throws Exception {
        FileInputStream fi = new FileInputStream("D:\\pract5\\sourcedata.xls");
        Workbook w = Workbook.getWorkbook(fi);
        Sheet s = w.getSheet(0);
        String a[][] = new String[s.getRows()][s.getColumns()];
        FileOutputStream fo = new FileOutputStream("D:\\pract5\\result.xls");
        WritableWorkbook wwb = Workbook.createWorkbook(fo);
        WritableSheet ws = wwb.createSheet("result1", 0);
        for (int i = 0; i < s.getRows(); i++)
        {
            for (int j = 0; j < s.getColumns(); j++)
            {
                a[i][j]=s.getCell(j,i).getContents();
                Label l2=new Label(j,i,a[i][j]);
                ws.addCell(l2);
                Label l1=new Label(6,0,"Results");
            }
        }
    }
}
```

## Software Testing And Quality Assurance

```
ws.addCell(l1);

}

}

for (int i = 1; i < s.getRows(); i++)

{

    for (int j = 2; j < s.getColumns(); j++)

    {

        a[i][j]=s.getCell(j,i).getContents();

        int x=Integer.parseInt(a[i][j]);

        if(x>35)

        {

            Label l1=new Label(6,i,"Pass");

            ws.addCell(l1);

        }

        else

        {

            Label l1= new Label(6,i,"Fail");

            ws.addCell(l1);

            break;

        }

    }

}

wwb.write();

wwb.close();

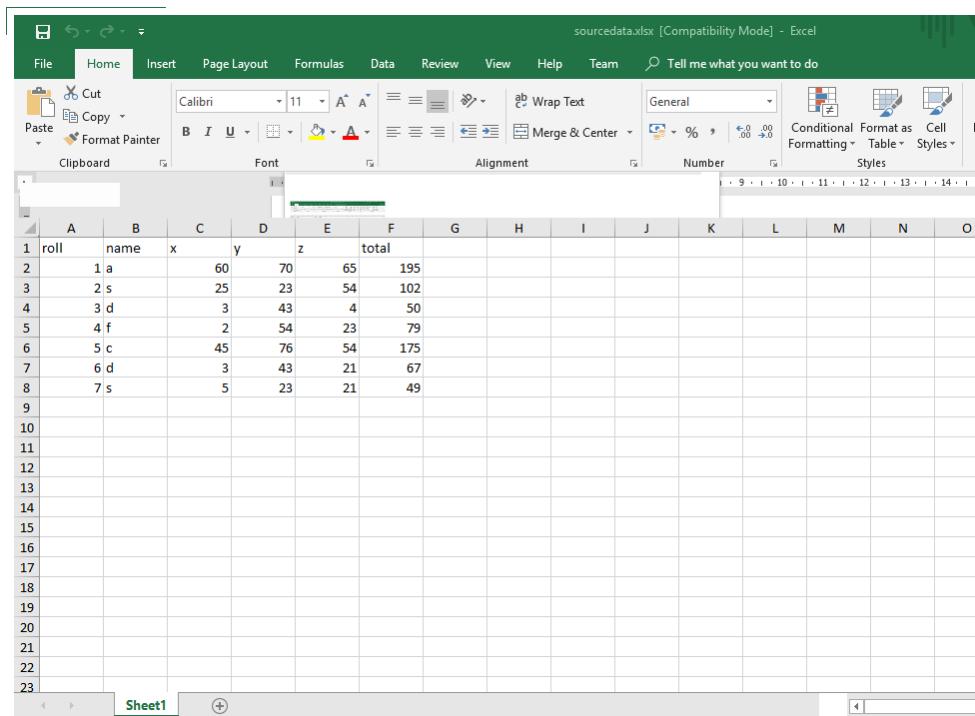
}

}
```

## Software Testing And Quality Assurance

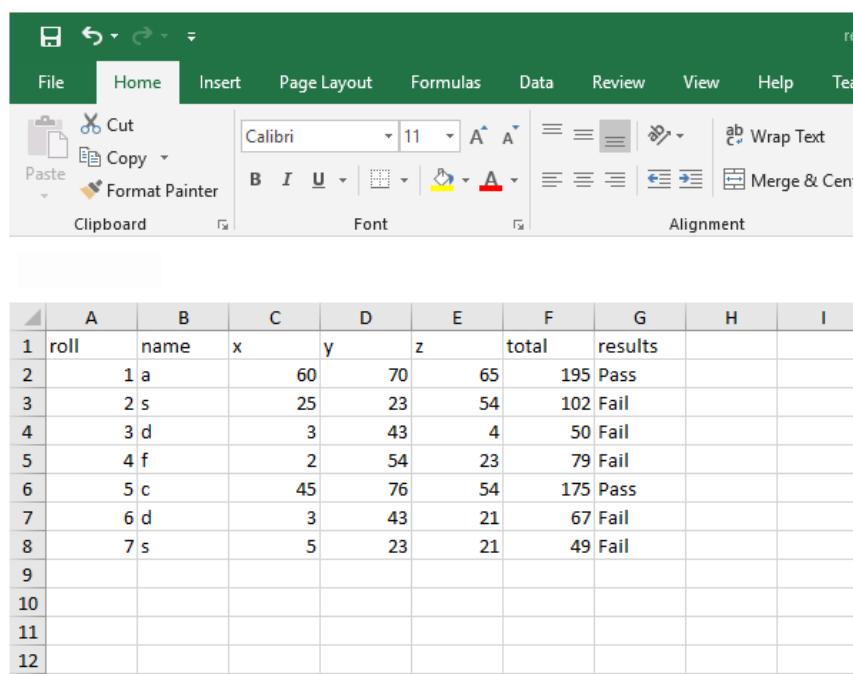
11. Now run the program u can see all the data from sourcedata in results including results pass and fail

'sourcedata' file



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	roll	name	x	y	z	total									
2	1	a	60	70	65	195									
3	2	s	25	23	54	102									
4	3	d	3	43	4	50									
5	4	f	2	54	23	79									
6	5	c	45	76	54	175									
7	6	d	3	43	21	67									
8	7	s	5	23	21	49									
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															

'results' file



	A	B	C	D	E	F	G	H	I
1	roll	name	x	y	z	total	results		
2	1	a		60	70	65	195	Pass	
3	2	s		25	23	54	102	Fail	
4	3	d		3	43	4	50	Fail	
5	4	f		2	54	23	79	Fail	
6	5	c		45	76	54	175	Pass	
7	6	d		3	43	21	67	Fail	
8	7	s		5	23	21	49	Fail	
9									
10									
11									
12									

## Practical No : 6

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

1. Create 2 files 'sourcedata' and 'results'

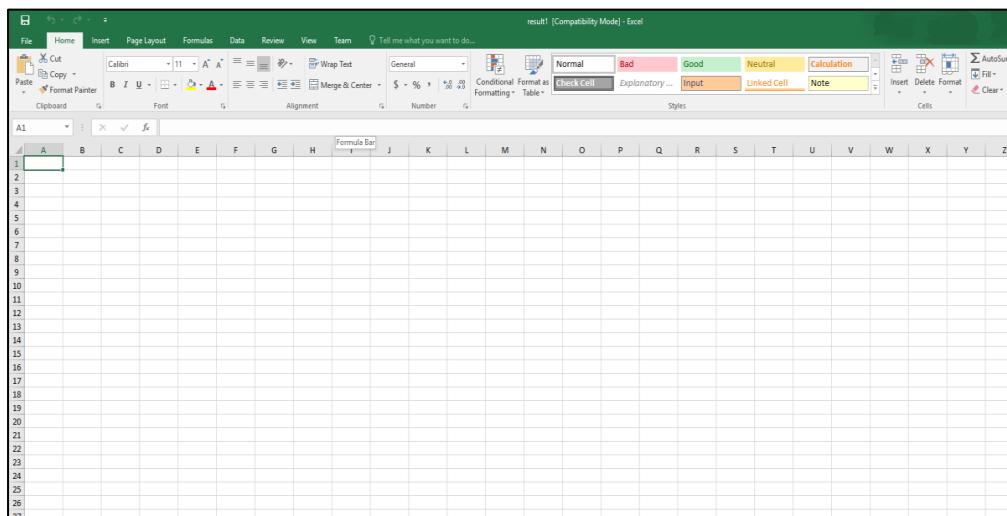
results	10-09-2018 20:34	Microsoft Excel W...	7 KB
sourcedata	10-09-2018 20:34	Microsoft Excel W...	7 KB

2. Insert data in 'sourcedata' file

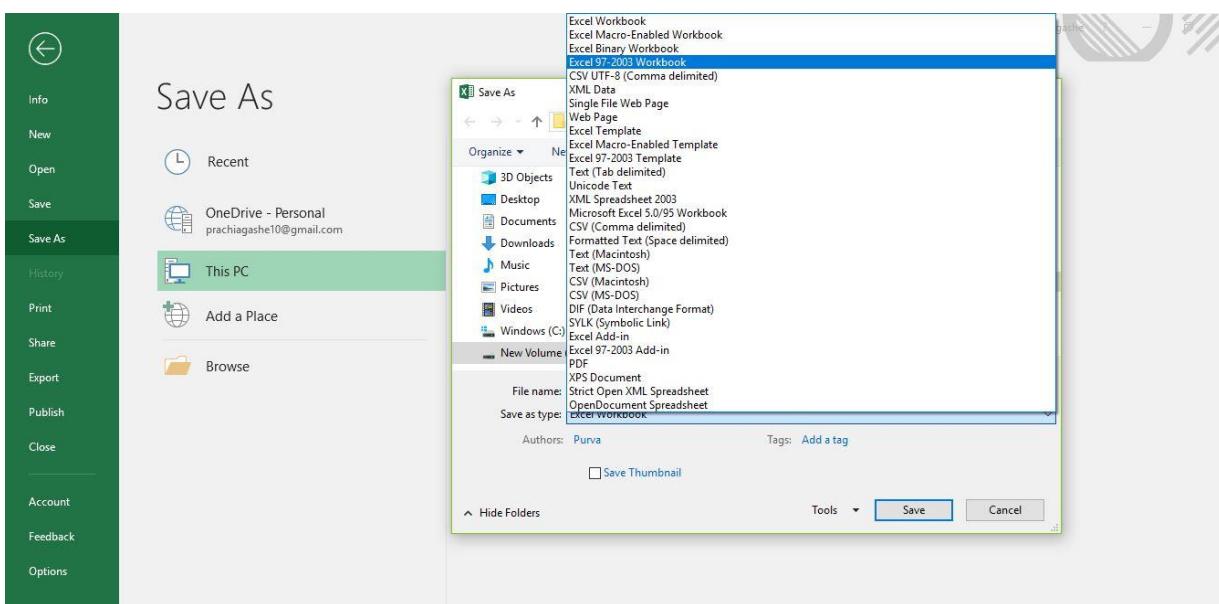
	roll	name	x	y	z	total
2	1	a	60	70	65	195
3	2	s	25	23	54	102
4	3	d	3	43	4	50
5	4	f	2	54	23	79
6	5	c	45	76	54	175
7	6	d	3	43	21	67
8	7	s	5	23	21	49
9						
10						

## Software Testing And Quality Assurance

### 3. Empty file for results .



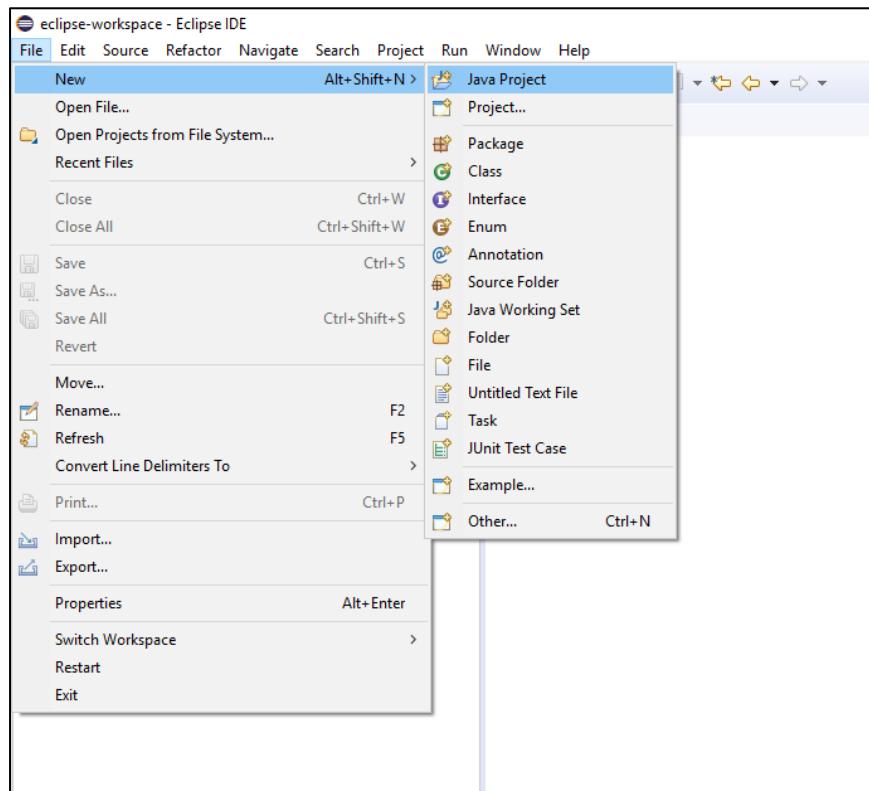
### 4. Save As Both the files by 'Excel 97-2013 WorkBook'



results	10-09-2018 20:36	Microsoft Excel 97...	25 KB
results	10-09-2018 20:34	Microsoft Excel W...	7 KB
sourcedata	10-09-2018 20:34	Microsoft Excel W...	7 KB
sourcedata.xlsx	10-09-2018 20:36	Microsoft Excel 97...	25 KB

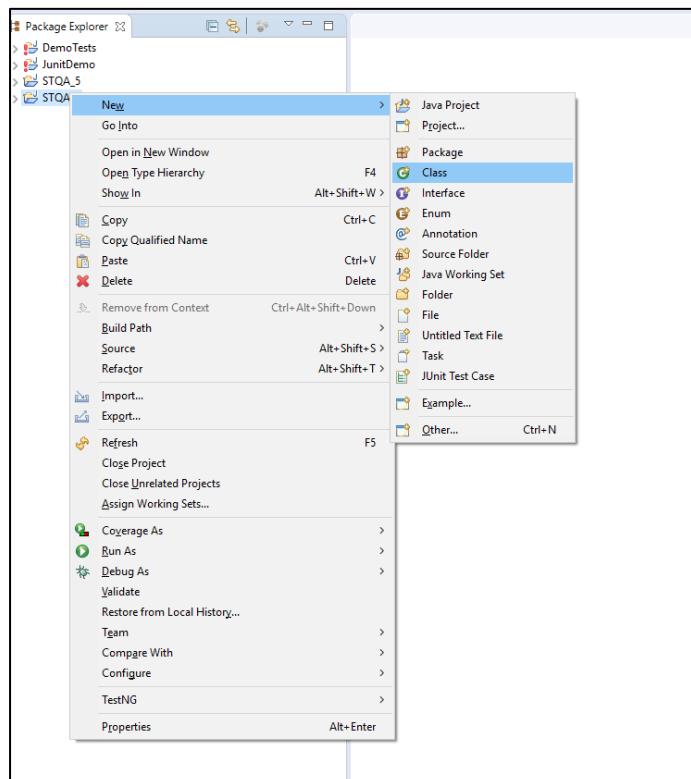
## Software Testing And Quality Assurance

### 5. Open new project in Eclipse.

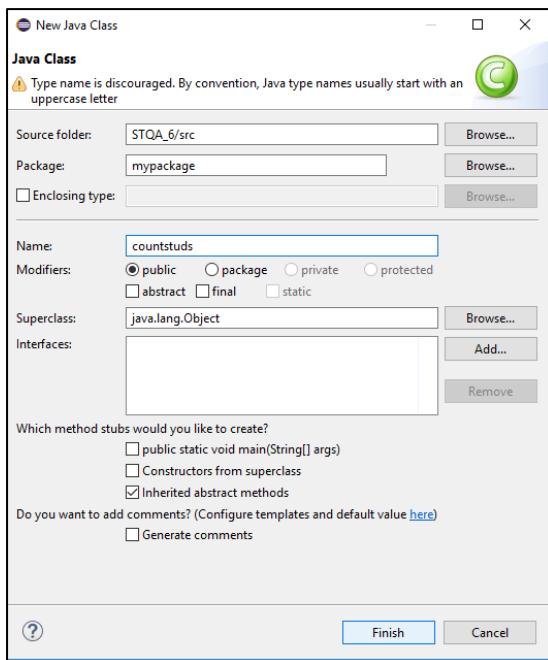


Name It As 'STQApract'

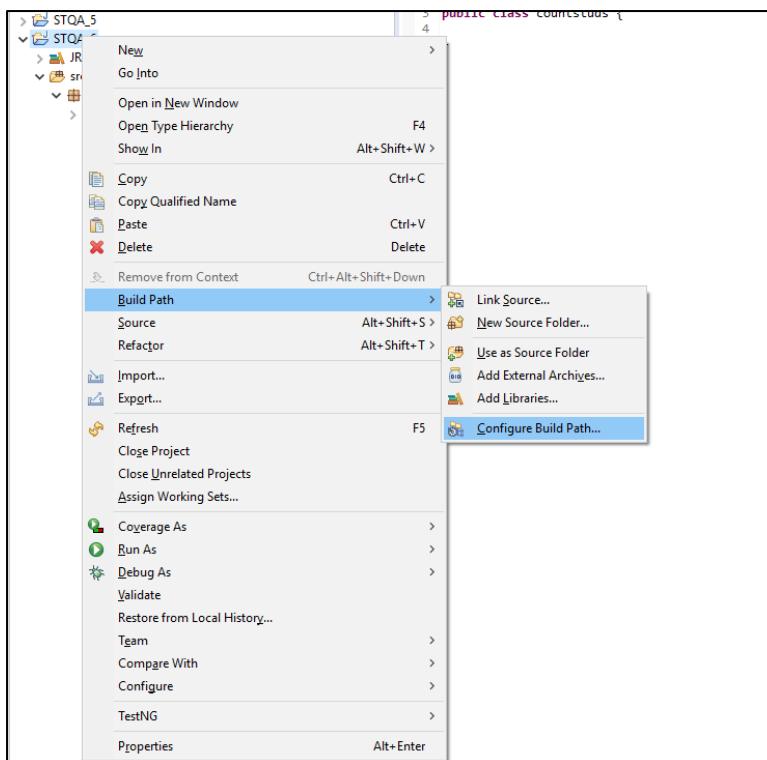
### 6. Add new Class file name it as countstuds



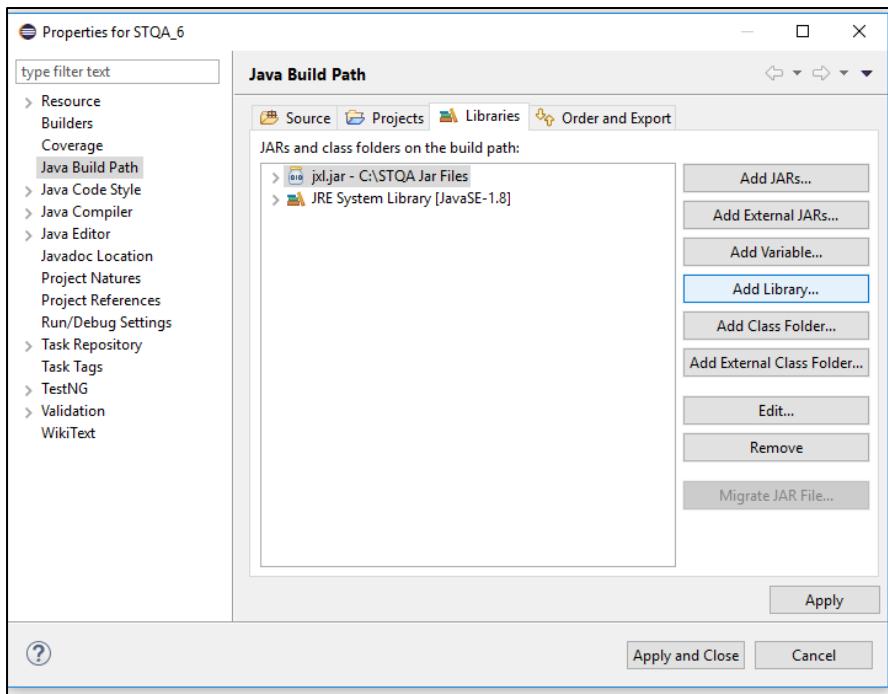
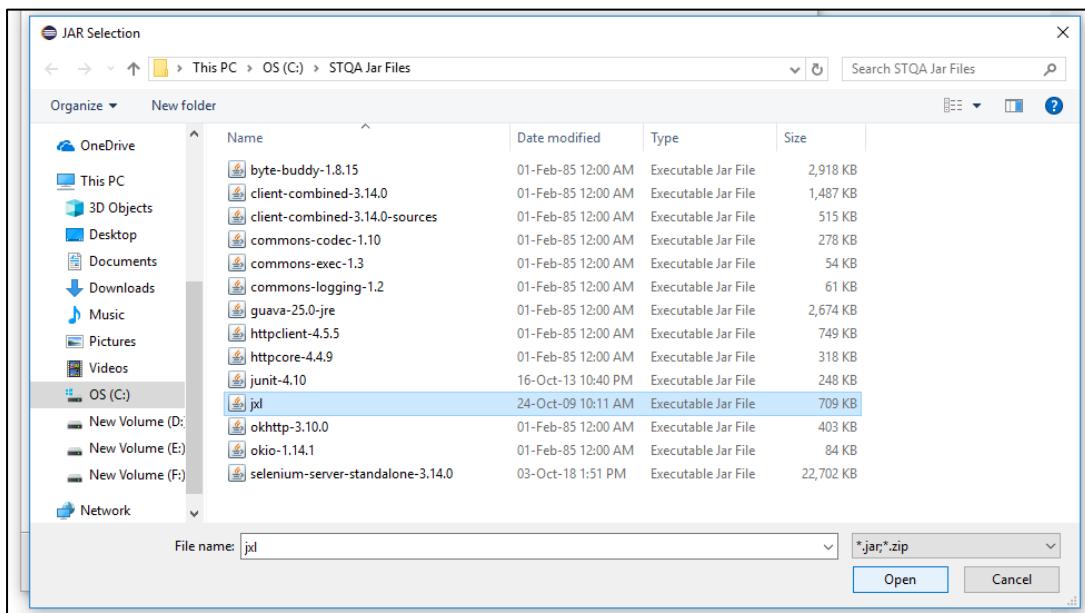
## Software Testing And Quality Assurance



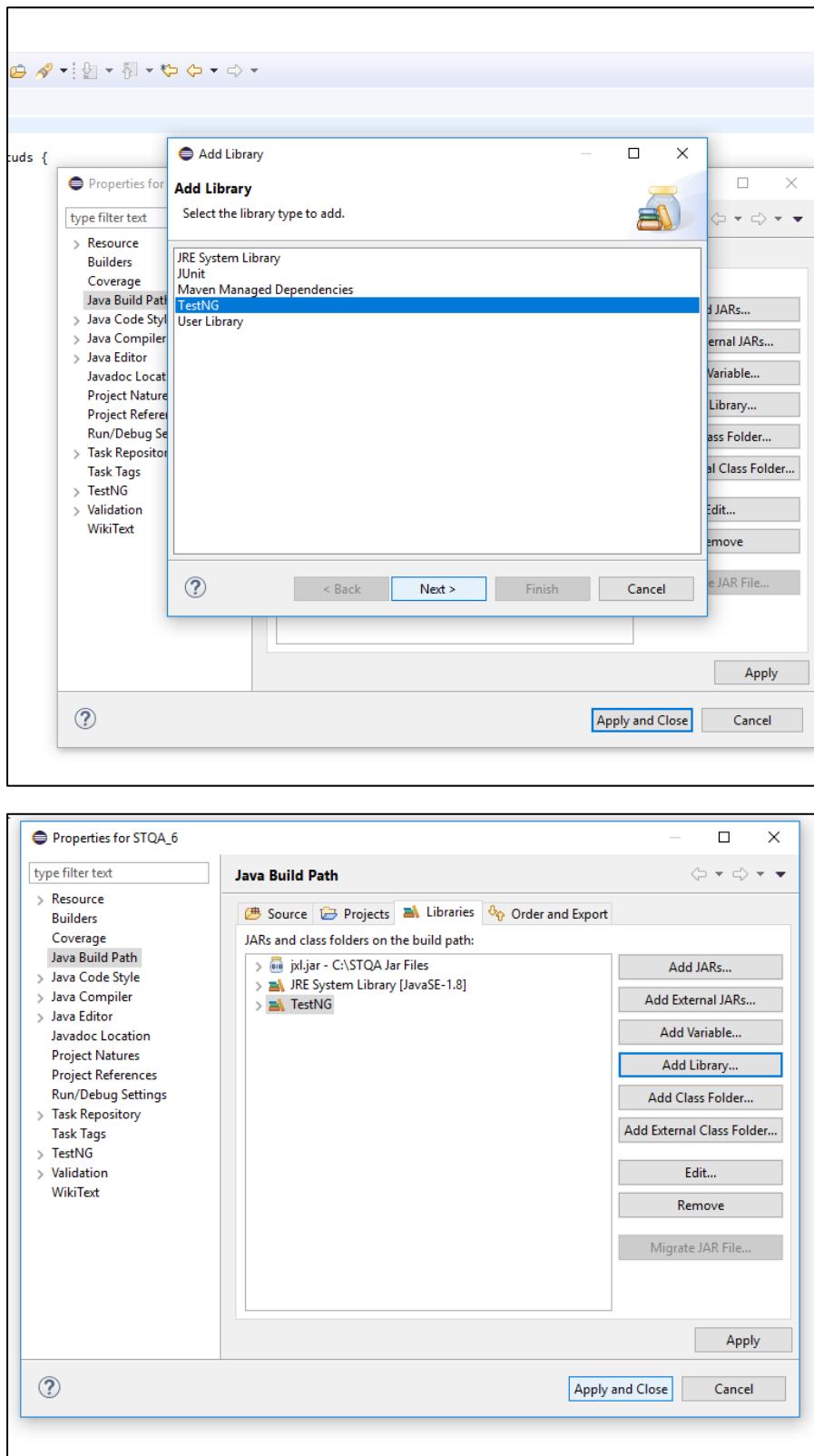
## 7. Add Configuration Files in Build Path



### 8. Add External Jar files jxl.exe



### 9. Add New Library TestNG



### 10. Write the following code

```
package p5;

import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;
import jxl.*;
import jxl.read.*;
import jxl.write.*;
import java.io.*;

public class countstuds{

    @BeforeClass
    public void f1()
    {}

    @Test
    public void testImportexport1() throws Exception {
        FileInputStream fi = new FileInputStream("D:\\pract5\\sourcedata.xls");
        Workbook w = Workbook.getWorkbook(fi);
        Sheet s = w.getSheet(0);
        String a[][] = new String[s.getRows()][s.getColumns()];
        FileOutputStream fo = new FileOutputStream("D:\\pract5\\result.xls");
        WritableWorkbook wwb = Workbook.createWorkbook(fo);
        WritableSheet ws = wwb.createSheet("result1", 0);
        int c=0;
        for (int i = 0; i < s.getRows(); i++)
        {
            for (int j = 0; j < s.getColumns(); j++)
            {
                if(i>=1)
                {
                    String b=new String();
                    b=s.getCell(3,i).getContents();

```

## **Software Testing And Quality Assurance**

```
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();
}
}
}
```

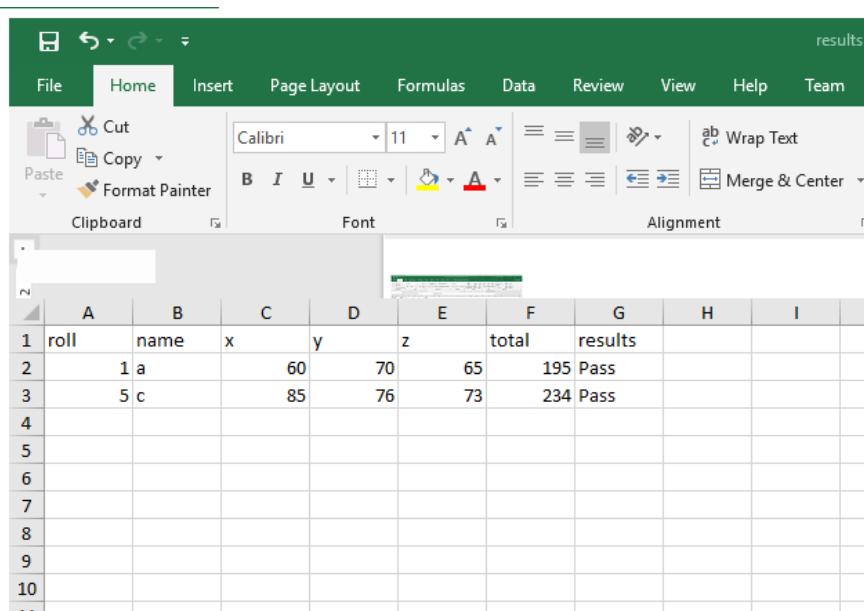
## Software Testing And Quality Assurance

11. Now run the program u can see all the data from sourcedata in results including results pass and fail

'sourcedata' file

	A	B	C	D	E	F	G	H
1	roll	name	x	y	z	total		
2	1	a		60	70	65	195	
3	2	s		25	23	54	102	
4	3	d		3	43	4	50	
5	4	f		2	54	23	79	
6	5	c		85	76	73	234	
7	6	d		3	43	21	67	
8	7	s		5	23	21	49	
9								
10								
11								
12								

'results' file



	A	B	C	D	E	F	G	H	I
1	roll	name	x	y	z	total			results
2	1	a		60	70	65	195		Pass
3	5	c		85	76	73	234		Pass
4									
5									
6									
7									
8									
9									
10									



```

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

```

**Input:-**

	A	B	C	D	E	F	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							