EX NO: 5 EXECUTE THE TEST CASES AGAINST A CLIENT SERVER OR DESKTOP APPLICATION AND IDENTIFY THE DEFECTS.

AIM:

Program:

1.html:

```
<html>
<head>
<title>JavascriptLoginFormValidation</title>
<!--IncludeCSSFileHere-->
<linkrel="stylesheet"href="form-style.css"/>
<!--IncludeJSFileHere -->
<scriptsrc="login.js"></script>
</head>
<body>
<divclass="container">
<divclass="main">
<h2>JavascriptLoginFormValidation</h2>
<formid="form_id"method="post"name="myform">
<label>UserName:</label>
<inputtype="text"name="username"id="username"/>
<label>Password:</label>
<inputtype="password"name="password"id="password"/>
<inputtype="button"value="Login"id="submit"onclick="validate()"/>
</form>
<span><bclass="note">Note:</b>For this demousefollowingusernameand password.
<br/><bclass="valid">UserName: Form<br/>Password:123</b></span>
</div>
</div>
</body>
```

```
</html>
```

login.js:

```
varattempt =3; //Variable tocount numberofattempts.
//BelowfunctionExecutesonclickofloginbutton.functio
nvalidate(){
varusername=document.getElementById("username").value;varp
assword=document.getElementById("password").value;
if(username=="Form"&&password=="123"){alert("
Login successfully");
window.location="success.html";//Redirectingtootherpage.returnf
alse;
}
else{
attempt --; // Decrementing by one.
Ialert("Youhaveleft"+attempt+"attempt;");
// Disabling fields after 3
attempts.if(attempt ==0){
document.getElementById("username").disabled=true;doc
ument.getElementById("password").disabled =
true;document.getElementById("submit").disabled =
true;returnfalse;
}
}
}
```

success.html:

```
<html>

<head>

<title>JavascriptLoginFormValidation</title>

</head>

<body>


<h2>SuccessfulLogin!</h2>

</body>
</html>
```

RESULT:

Ex.No:6 TEST THE PERFORMANCE OF THE E-COMMERCE APPLICATION

AIM:

PROCEDURE:

E-Commerce applications are designed in such a way that customers can easily make purchases at any time of the day, irrespective of where they are located. If an eCommerce application malfunctions during peak hours, not only will it cause customer dissatisfaction, but it will also lead to revenue loss for the business.

To avoid such situations, companies can invest in performance testing of eCommerce applications that focuses on issues related to speed, stability, and scalability. Hence, in the wake of increasing demand for online shopping, it is necessary to do performance testing of the eCommerce application and toensure that the application is stable and scalable enough to serve all customer requirements.

Goals of E-Commerce Performance Testing

Performance testing is carried out on eCommerce software to ensure that the platform is functioning as expected. Let's explore why eCommerce application performance testing is important and how it affects the app's overall quality:

- Reduce Risks: Many times, making major and considerable changes to a site can cause notable strategic changes or even trigger significant losses. However, testing these changes in a planned manner can help eliminate the chances of these uncertain revenue losses

- Increase Conversion Rates: By testing every aspect of an application and ensuring a smooth visitor experience through site optimization, the application conversion rate is bound to increase.

- Improve User Engagement: Testing tells which page element or process affects a user’s onsite journey and helps in rectifying the issues faster. The better the user experience, the greater the onsite engagement.

Benefits of E-Commerce Application Performance Testing

Performance testing, when done effectively and consistently, can greatly boost conversions while also enhancing the overall experience of site users. The following reasons explain the significance of testing and optimization:

- Coupon codes or gift vouchers are provided to increase product sales and performance tests ensure that coupon codes work well when applied to users in bulk.

- Multiple systems like email servers, social network sites, and enterprise content management systems are involved in the backend and the performance test ensures flawless functioning of various features like product images/videos, and social media in such an integrated system.

- eCommerce applications usually have more users than any other application and validation of users is a must to filter out genuine customers. Performance test helps in validating multiple sign-up and invoice email notifications in parallel.

- Augmented Reality (AR) is an advanced-level feature provided by modern eCommerce websites so that customers can feel better without seeing products in person. With augmented reality performance testing, we can make the feature more efficient and pleasing to the eye.

- A recommendation system is a program that generates online recommendations using several algorithms, artificial intelligence, and data analysis. A well-designed recommendation system helps in better customer acquisition and retention. It is developed for both new and existing customers with different behavior for different customer types. Performance testing helps in making the recommendation system a faster system, to retain online shoppers.

- Chatbots interact with online shoppers to improve the user experience, and any delay in providing the required information may switch users to a competitor's application. Performance tests speed up chatbots to answer customer queries.

- Reviews are very important to increase the sale of a product. The majority of users prefer visiting the review listing page to get instant user feedback. Performance test helps in optimizing customer review listing load time.

- eCommerce applications attract the maximum number of customers during annual sales events. Backend servers should be set with the optimal settings, keeping future users' traffic in mind. Performance tests help in identifying the servers' optimum locations to manage a higher number of users.

Common Performance Testing Issues

Some common challenges faced in implementing performance testing are as follows:

- Slow DB server: Database size increases with the increase in product items of all ages, so ensure to optimize DB queries.
- Faster Integrated Systems Communication: Multiple systems like email servers, social media accounts, and payment channels are involved in the functioning of an eCommerce application to ensure that integrated systems work well under a heavy load

- System Scaling: Brands cannot predict user load during peak hours, hence, the need of the hour is to be ready with tested scalable systems.

- Immoderate Scripts: Few eCommerce applications are associated with third-party scripts that are running in the background like Google Analytics and Ads, and these widgets can be a significant contributor to page load time.

- Slow Servers: Server resources and applications should be optimized to deliver content as quickly as possible after addressing any content-related performance issues.

- Massive Media: Oftentimes, images, videos, and other media are uploaded by non-technical employees that aren't familiar with the impact of loaded heavy files. It's important to ensure these assets are optimized to maximize performance

Main Features to be Tested in E-Commerce Applications
Best Practices for Performance Testing

Mentioned below are some of the best practices for testing eCommerce applications:

- Test all CDNs (content delivery networks) by repeating time-bound hot transactions multiple times.
- Performance testing of big data plays an important role to ensure faster decompression and compression cycles to speed up the application's speed.
- Online shopping has opened doors worldwide for businesses. Vendors shifted their servers to the cloud for better user coverage. Hence, tests from various geo-locations need to be performed to identify latency issues and to ensure that the geographical location of a user does not affect the speed of the application.
- Test individual servers before integration system testing to find bottlenecks like email servers, social accounts, DB servers, and enterprise content management systems.
- Volume testing plays a crucial role in seeing how storage systems function with or without a heavy load, hence including it in the test scope is a good practice.

Role of QASource

Testing experts at QASource have years of experience in running performance tests on a wide range of domains including eCommerce, media streaming, legal, human resources, eLearning, healthcare , enterprise content management, financial, and many more. QASource assists teams in setting up the performance test lab and identifying the best-fit performance test scenarios to ensure the best user experience. Following are our high-level performance testing services to meet the performance test requirement:

- Performance test plan
- Test data setup
- Simulation of test cases
- Generate performance test suite results
- Identify bottlenecks and follow up with the Dev team to fix them
- Generate daily status reports

Conclusion

Performance testing of eCommerce applications is a crucial part of any business's success. It helps eCommerce applications generate better user experience, safeguard user data security, and ensure mobile responsiveness, security, and a quick load time. It is vital to conduct quick and effective application performance testing from time to time to ensure that the application is offering an exceptional customer experience to maximize revenue for businesses.

RESULT:

EX NO: 7 AUTOMATE THE TESTING OF E-COMMERCE APPPLICATIONS USING SELEINIUM

AIM:

Positive Scenario

1. Test Case - Automate the User Registration process of an e-commerce website.

Steps to Automate:
1. Open this URL http://automationpractice.com/index.php
2. Click on the sign-in link.
3. Enter your email address in the 'Create and Account' section.
4. Click on Create an Account button.
5. Enter your Personal Information, Address, and Contact info.
6. Click on the Register button.
7. Validate that the user is created.

Selenium code for User Registration:

```
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.openqa.selenium.support.ui.Select;
import io.github.bonigarcia.wdm.WebDriverManager;
public class EcomSignUp {
 public static void main(String[] args) {
  WebDriverManager.chromedriver().setup();
  WebDriver driver=new ChromeDriver();
  String URL="http://automationpractice.com/index.php";

  driver.get(URL);
  driver.manage().timeouts().implicitlyWait(2000, TimeUnit.MILLISECONDS);
  driver.manage().window().maximize();
  //Click on Sign in
  driver.findElement(By.linkText("Sign  in")).click();
  //Enter email address
driver.findElement(By.cssSelector("[name='email_create']")).sendKeys("test1249@test.com");
  //Click on "Create an account"
  driver.findElement(By.xpath("//button[@name=\"SubmitCreate\"]")).click();
  //Select Title
  driver.findElement(By.xpath("//input[@id=\"id_gender1\"]")).click();
  driver.findElement(By.name("customer_firstname")).sendKeys("Test User");
  driver.findElement(By.name("customer_lastname")).sendKeys("Vsoft");
  driver.findElement(By.id("passwd")).sendKeys("PKR@PKR");
  // Enter your address
  driver.findElement(By.id("firstname")).sendKeys("Test User");
  driver.findElement(By.id("lastname")).sendKeys("Vsoft");
  driver.findElement(By.id("company")).sendKeys("Vsoft");
  driver.findElement(By.id("address1")).sendKeys("Test 81/1,2nd cross");
  driver.findElement(By.id("city")).sendKeys("XYZ");
  // Select State
  WebElement statedropdown=driver.findElement(By.name("id_state"));
  Select oSelect=new Select(statedropdown);
  oSelect.selectByValue("4");
  driver.findElement(By.name("postcode")).sendKeys("51838");
  // Select Country
  WebElement countrydropDown=driver.findElement(By.name("id_country"));
  Select oSelectC=new Select(countrydropDown);
  oSelectC.selectByVisibleText("United States");
  //Enter Mobile Number
  driver.findElement(By.id("phone_mobile")).sendKeys("234567890");
  driver.findElement(By.xpath("//input[@name=\"alias\"]")).clear();
  driver.findElement(By.xpath("//input[@name=\"alias\"]")).sendKeys("Office");
  driver.findElement(By.name("submitAccount")).click();
  String userText=driver.findElement(By.xpath("//*[@id=\"header\"]/div[2]/div/div/nav/div[1]/a")).getText();
  // Validate that user has created
  if(userText.contains("Vsoft")) {
  System.out.println("User Verified,Test case Passed");
  }
  else {
   System.out.println("User Verification Failed,Test case Failed");
  }
 }}
```

<u>Negative Scenarios</u>

2. Test Case - Verify invalid email address error.

Steps to Automate:
1. Open this URL http://automationpractice.com/index.php
2. Click on the sign-in link.
3. Enter an invalid email address in the email box and click enter.
4. Validate that an error message is displaying saying "Invalid email address."

3. Test Case - Verify error messages for mandatory fields.

Steps to Automate:
1. Open this URL http://automationpractice.com/index.php
2. Click on the sign-in link.
3. Enter your email address and click the Register button.
4. Leave the mandatory fields (marked with *) blank and click the Register button.
5. Verify that an error has been displayed for the mandatory fields.

4. Test Case - Verify error messages for entering incorrect values in fields.

Steps to Automate:
1. Open this URL http://automationpractice.com/index.php
2. Click on the sign-in link.
3. Enter your email address and click the Register button.
4. Enter incorrect values in fields like., enter numbers in first and last name, city field, etc., and enter alphabets in Mobile no, Zip postal code, etc., and click on the 'Register' button.
5. Verify that error messages for respective fields are displaying.

Try automating the above scenarios using Selenium commands, if you face any difficulty please refer to the Selenium Tutorial series.

5. Automate the 'Search Product' feature of Amazon like e-commerce website with Selenium

1. Test Case - Automate the 'Search Product' feature of the e-commerce website with Selenium.

Steps to Automate:
1. Open link http://automationpractice.com/index.php
2. Move your cursor over the Women's link.
3. Click on the sub-menu 'T-shirts'
4. Get the Name/Text of the first product displayed on the page.
5. Now enter the same product name in the search bar present at the top of the page and click the search button.
6. Validate that the same product is displayed on the searched page with the same details which were displayed on T-Shirt's page.

Automation Code for Product Search:

```
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.openqa.selenium.interactions.Actions;
import io.github.bonigarcia.wdm.WebDriverManager;
public class EcomPractice2 {
 public static void main(String[] args) throws InterruptedException{
  WebDriverManager.chromedriver().setup();
  WebDriver driver=new ChromeDriver();
  String URL="http://automationpractice.com/index.php";
  driver.get(URL);
  driver.manage().window().maximize();
  // Initialise Actions class object
  Actions actions=new Actions(driver);
  driver.manage().timeouts().implicitlyWait(2000, TimeUnit.MILLISECONDS);
  WebElement womenTab=driver.findElement(By.linkText("WOMEN"));
  WebElement
TshirtTab=driver.findElement(By.xpath("//div[@id='block_top_menu']/ul/li[1]/ul/li[1]/ul//a[@title='T-
shirts']"));
  actions.moveToElement(womenTab).moveToElement(TshirtTab).click().perform();
  Thread.sleep(2000);
  // Get Product Name
  String
ProductName=driver.findElement(By.xpath("/html[1]/body[1]/div[1]/div[2]/div[1]/div[3]/div[2]/ul[1]/li[1]/di
v[1]/div[2]/h5[1]/a[1]")).getText();
  System.out.println(ProductName);
  driver.findElement(By.id("search_query_top")).sendKeys(ProductName);
  driver.findElement(By.name("submit_search")).click();
  // Get Name of Searched Product
  String
SearchResultProductname=driver.findElement(By.xpath("/html[1]/body[1]/div[1]/div[2]/div[1]/div[3]/div[2]/
ul[1]/li[1]/div[1]/div[2]/h5[1]/a[1]")).getText();
  // Verify that correct Product is displaying after search
  if(ProductName.equalsIgnoreCase(SearchResultProductname)) {
  System.out.println("Results Matched;Test Case Passed");
  }else{
   System.out.println("Results NotMatched;Test Case Failed");
  }
  // Close the browser
  driver.close();
 }
}
```

6. Automate the 'Buy Product' feature of Amazon like an e-commerce website with Selenium

The most important function of an e-commerce website is buying a product, which includes various steps like selecting a product, selecting size/color, adding to the cart, checkout, etc. You will find every test scenario along with the automation code in the following section.

1. Test Case - Automate the end-to-end "Buy Product" feature of the e-commerce website.

Steps to Automate:
1. Open link http://automationpractice.com/index.php
2. log in to the website.
3. Move your cursor over the Women's link.
4. Click on the sub-menu 'T-shirts'.
5. Mouse hover on the second product displayed.
6. 'More' button will be displayed, click on the 'More' button.
7. Increase quantity to 2.
8. Select size 'L'
9. Select color.
10. Click the 'Add to Cart' button.
11. Click the 'Proceed to checkout' button.
12. Complete the buy order process till payment.
13. Make sure that the Product is ordered.

Automation code for Buy product

```
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.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.Select;
import io.github.bonigarcia.wdm.WebDriverManager;
public class EcomExpert {
 public static void main(String[] args){
  WebDriverManager.chromedriver().setup();
  WebDriver driver=new ChromeDriver();
  String URL="http://automationpractice.com/index.php";

  // Open URL and Maximize browser window
  driver.get(URL);
  driver.manage().window().maximize();
  driver.manage().timeouts().implicitlyWait(3000,  TimeUnit.MILLISECONDS);
  //Click on Sign in
  driver.findElement(By.linkText("Sign in")).click();
  //Login
  driver.findElement(By.id("email")).sendKeys("test1249@test.com");
  driver.findElement(By.id("passwd")).sendKeys("PKR@PKR");
  driver.findElement(By.id("SubmitLogin")).click();
  //Click on Women
  driver.findElement(By.linkText("WOMEN")).click();
  WebElement
SecondImg=driver.findElement(By.xpath("/html/body/div[1]/div[2]/div/div[3]/div[2]/ul/li[2]/div/div[1]
/div/a[ 1]/img"));
  WebElement
MoreBtn=driver.findElement(By.xpath("/html/body[1]/div[1]/div[2]/div[1]/div[3]/div[2]/ul/li[2]/div[1]/
  div[2]
/div[2]/a[2]"));
  Actions actions=new Actions(driver);
  actions.moveToElement(SecondImg).moveToElement(MoreBtn).click().perform();
  //Change quantity by 2
  driver.findElement(By.id( "quantity_wanted" )).clear();
  driver.findElement(By.id( "quantity_wanted" )).sendKeys("2");
  //Select size as L

  WebElement Sizedrpdwn=driver.findElement(By.xpath("//*[@id='group_1']"));
  Select oSelect=new Select(Sizedrpdwn);
  oSelect.selectByVisibleText("M");
  //Select Color
  driver.findElement(By.id("color_11")).click();
  //Click on add to cart
  driver.findElement(By.xpath("//p[@id='add_to_cart']//span[.='Add to cart']")).click();
  //Click on proceed
  driver.findElement(By.xpath("/html//div[@id='layer_cart']//a[@title='Proceed  to  checkout']/span")).click();
  //Checkout page Proceed
  driver.findElement(By.xpath("/html/body/div[1]/div[2]/div/div[3]/div/p[2]/a[1]/span")).click();
  driver.findElement(By.xpath("/html/body/div[1]/div[2]/div/div[3]/div/form/p/button/span")).click();
  //Agree terms&Conditions
  driver.findElement(By.xpath("//*[@id=\"cgv\"]")).click();
  driver.findElement(By.xpath("/html/body/div[1]/div[2]/div/div[3]/div/div/form/p/button/span")).click();
  //Click on Payby Check
  driver.findElement(By.xpath("/html/body/div[1]/div[2]/div/div[3]/div/div/div[3]/div[2]/div/p/a")).click();
  //Confirm the order
  driver.findElement(By.xpath("/html/body/div[1]/div[2]/div/div[3]/div/form/p/button/span")).click();
  //Get Text
  String ConfirmationText=driver.findElement(By.xpath("//div[@id='center_column']/p[@class='alert alert-
success']")).getText();
  // Verify that Product is ordered
  if(ConfirmationText.contains("complete")) {
  System.out.println("Order Completed: Test Case Passed");
  }
  else {
   System.out.println("Order Not Successfull: Test Case Failed");
  }

 }
}
```

2. Test Case - Verify that 'Add to Wishlist' only works after login.

Steps to Automate:

1. Open link http://automationpractice.com/index.php

2. Move your cursor over the Women's link.

3. Click on the sub-menu 'T-shirts'.

4. Mouse hover on the second product displayed.

5. 'Add to Wishlist' will appear on the bottom of that product, click on it.

6. Verify that the error message is displayed 'You must be logged in to manage your wish list.'

3. Test Case - Verify that Total Price is reflecting correctly if the user changes quantity on the 'Shopping Cart Summary' Page.

Steps to Automate:

1. Open link http://automationpractice.com/index.php

2. Log in to the website.

3. Move your cursor over the Women's link.

4. Click on the sub-menu 'T-shirts'.

5. Mouse hover on the second product displayed.

6. 'More' button will be displayed, click on the 'More' button.

7. Make sure the quantity is set to 1.

8. Select size 'M'

9. Select the color of your choice.

10. Click the 'Add to Cart' button.

11. Click the 'Proceed to checkout' button.

12. Change the quantity to 2.

13. Verify that the Total price is changing and reflecting the correct price.

RESULT:

EX.NO:8 INTEGRATE TestNG WITH THE ABOVE TEST AUTOMATION

AIM:

PROCEDURE:

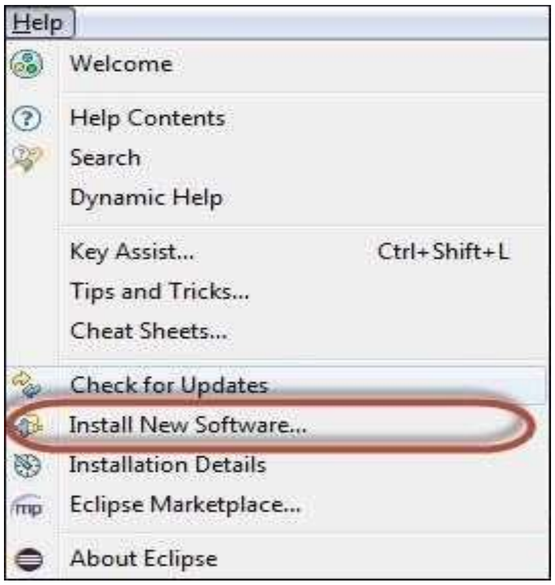
TestNG is a powerful testing framework, an enhanced version of JUnit which was in use for a long time before TestNG came into existence. NG stands for 'Next Generation'.

TestNG framework provides the following features −

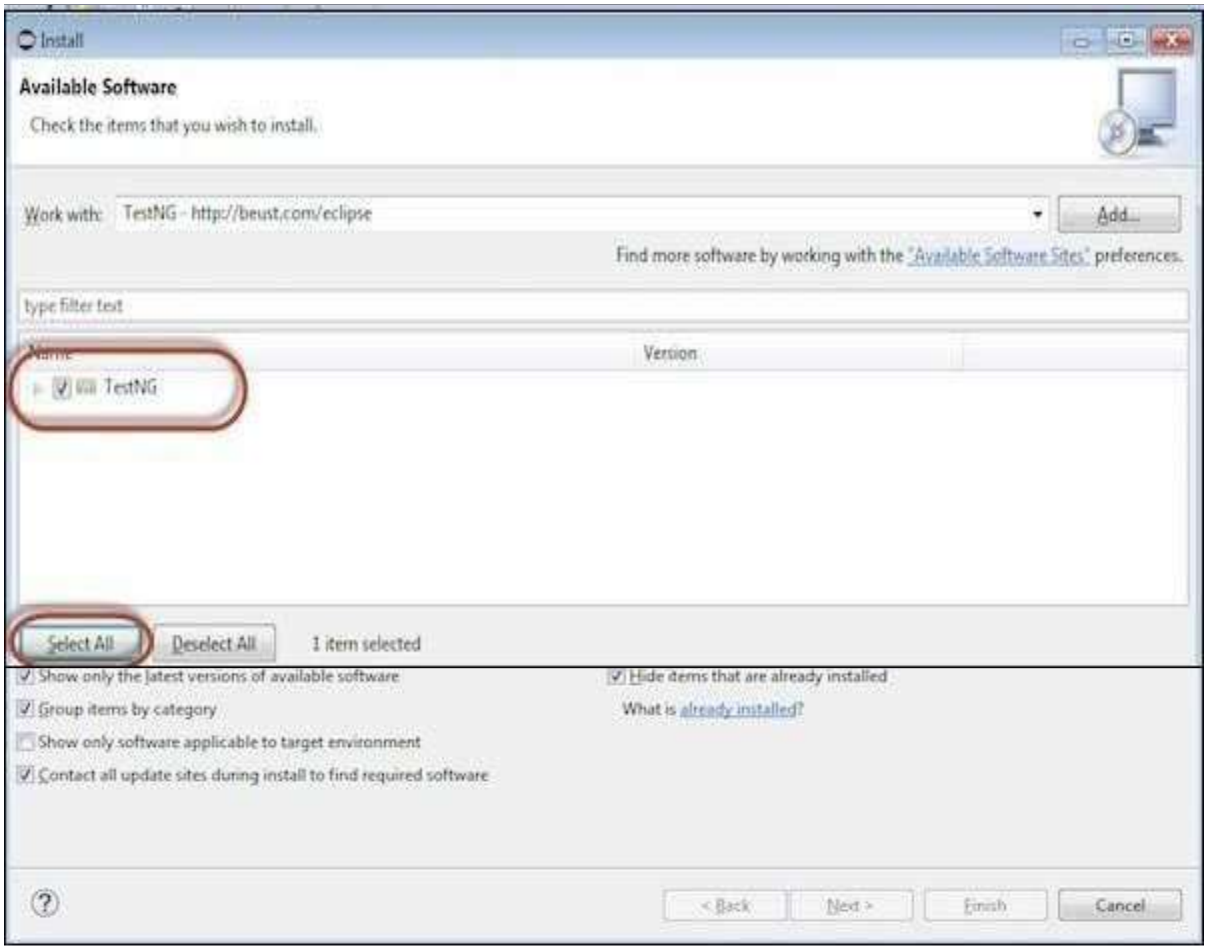
- Annotations help us organize the tests easily.
- Flexible test configuration.
- Test cases can be grouped more easily.
- Parallelization of tests can be achieved using TestNG.
- Support for data-driven testing.
- Inbuilt reporting.

Installing TestNG for Eclipse

Step 1 − Launch Eclipse and select 'Install New Software'.



Step 2 − Enter the URL as 'http://beust.com/eclipse' and click 'Add'.


Install

Available Software

Check the items that you wish to install.

Work with: TestNG - http://beust.com/eclipse

Find more software by working with the "Available Software Sites" preferences.

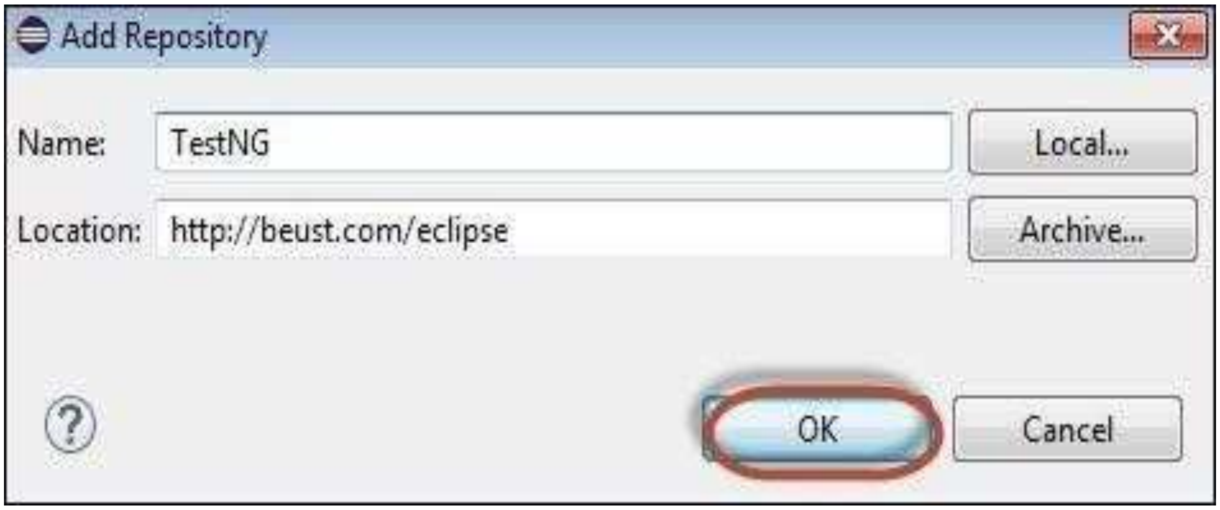
type filter text

Name	Version
☑ TestNG	

1 item selected

☑ Show only the latest versions of available software
☑ Group items by category
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software
☑ Hide items that are already installed
What is already installed?

Step 3 − The dialog box 'Add Repository' opens. Enter the name as 'TestNG' and click 'OK'

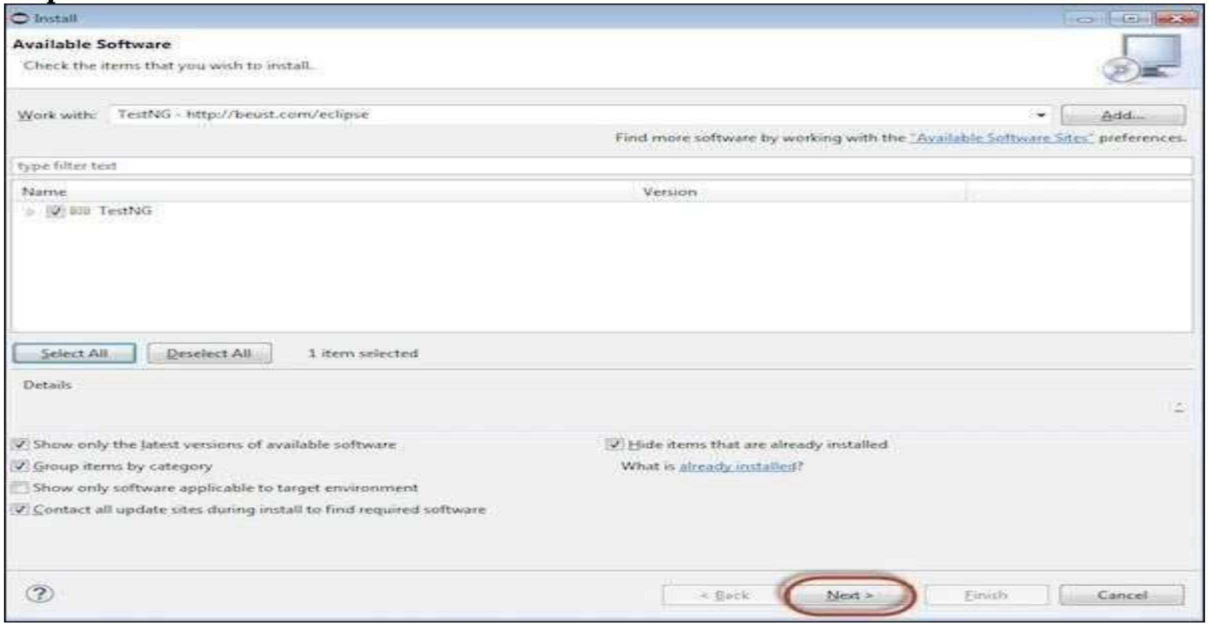

Add Repository

Name: TestNG

Location: http://beust.com/eclipse

Step 4 − Click 'Select All' and 'TestNG' would be selected as shown in the figure.

Step 5 − Click 'Next' to continue.



Install

Available Software

Check the items that you wish to install.

Work with: TestNG - http://beust.com/eclipse

Find more software by working with the "Available Software Sites" preferences.

type filter text

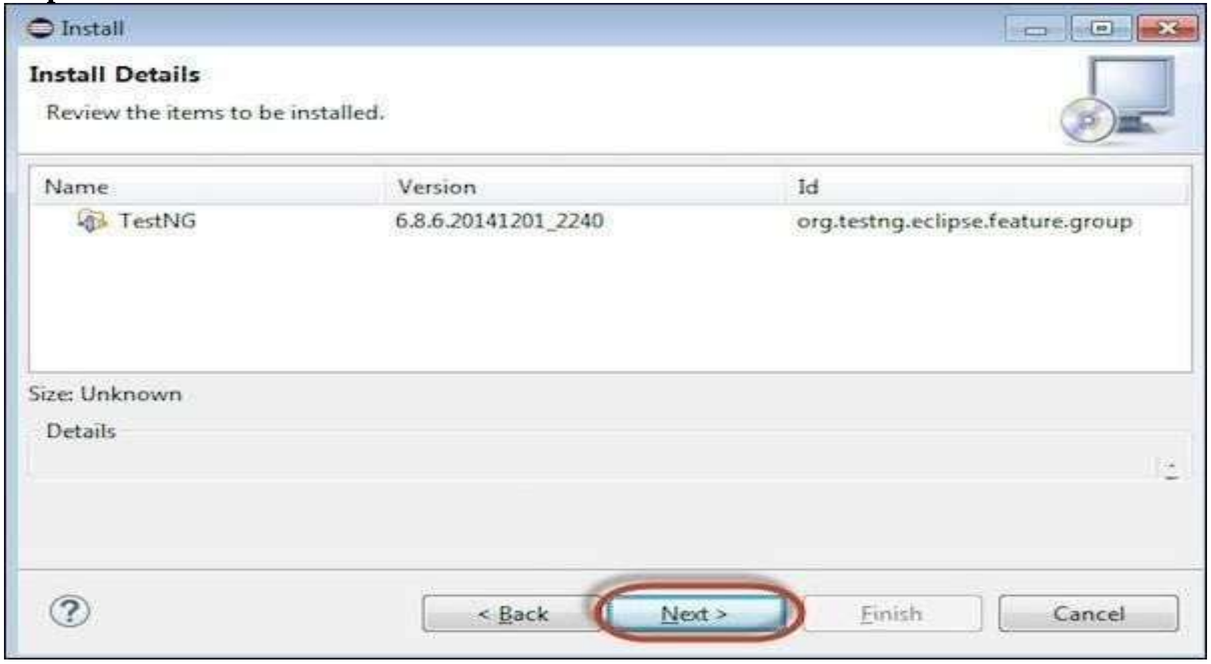
Name	Version
☑ TestNG	

1 item selected

Details

☑ Show only the latest versions of available software
☑ Group items by category
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software
☑ Hide items that are already installed
What is already installed?

Step 6 − Review the items that are selected and click 'Next'.



Install

Install Details

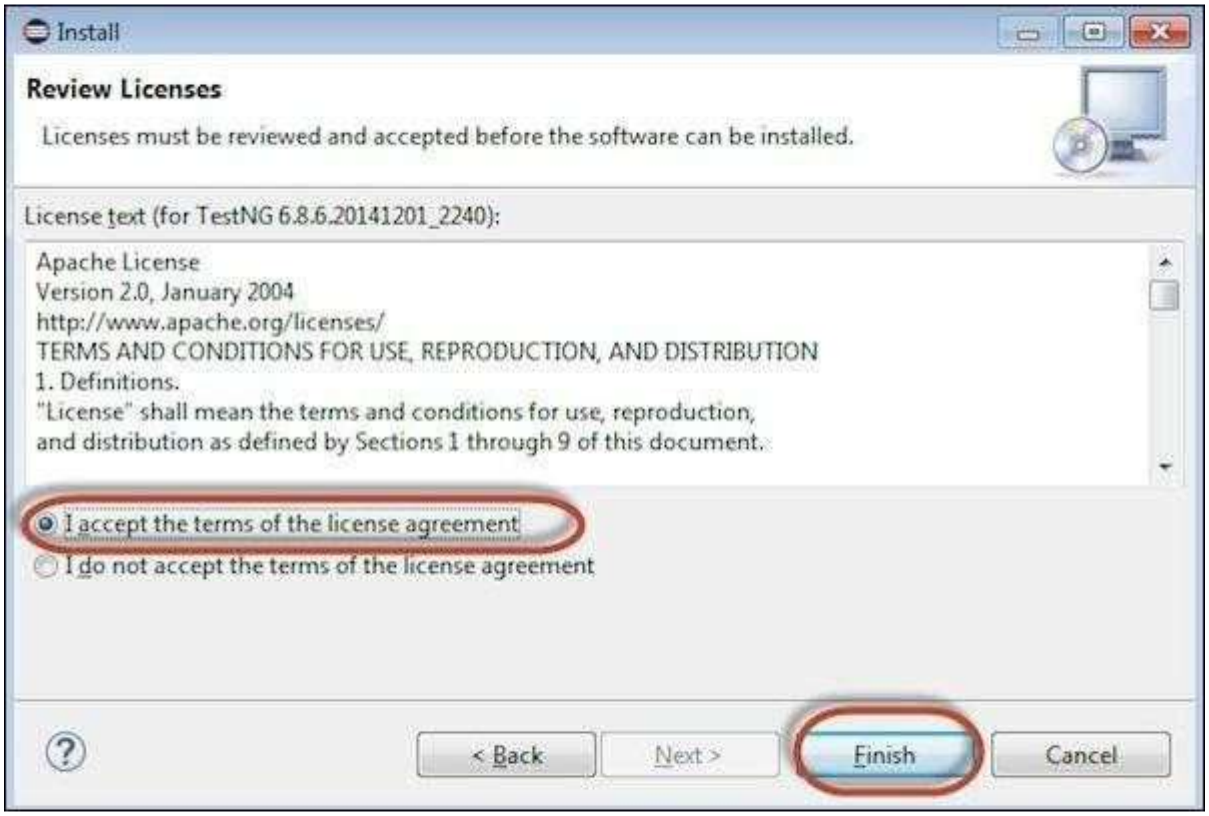
Review the items to be installed.

Name	Version	Id
TestNG	6.8.6.20141201_2240	org.testng.eclipse.feature.group

Size: Unknown

Details

Step 7 − "Accept the License Agreement" and click 'Finish'.


Install

Review Licenses

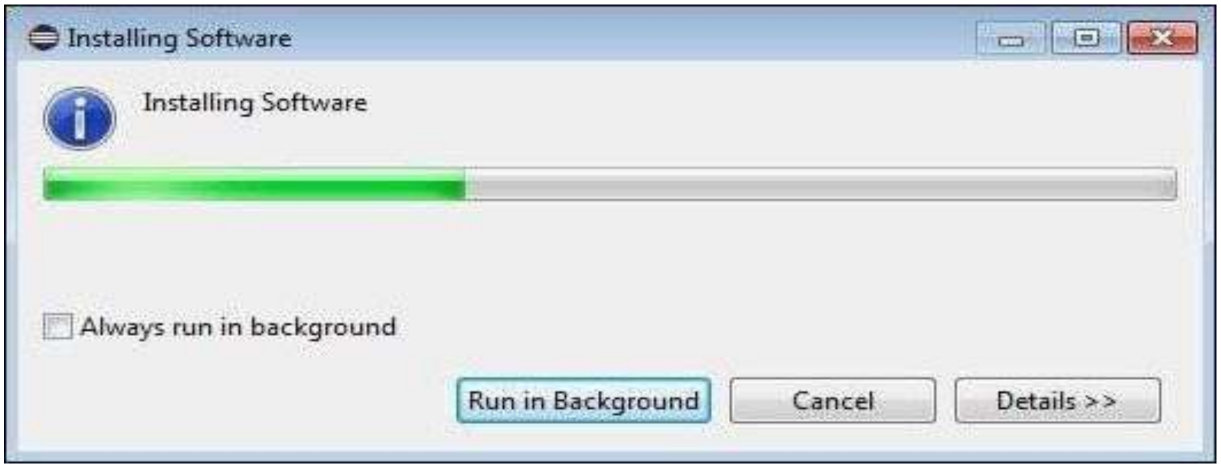
Licenses must be reviewed and accepted before the software can be installed.

License text (for TestNG 6.8.6.20141201_2240):

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

[✓ I accept the terms of the license agreement]
I do not accept the terms of the license agreement

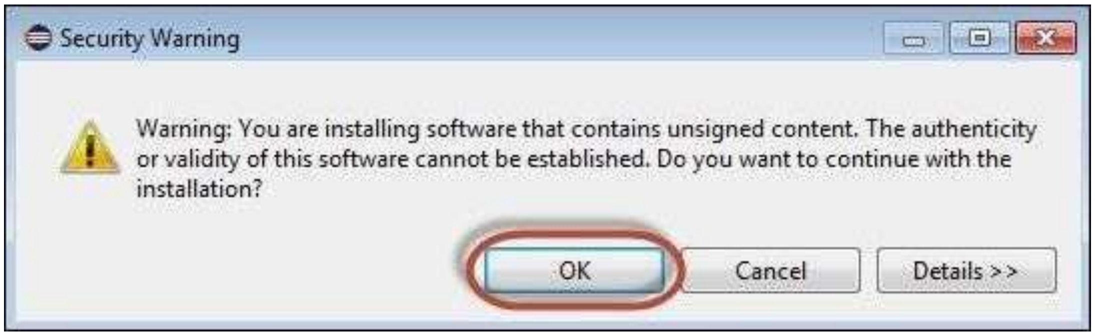
Step 8 − TestNG starts installing and the progress would be shown follows.


Installing Software

Installing Software

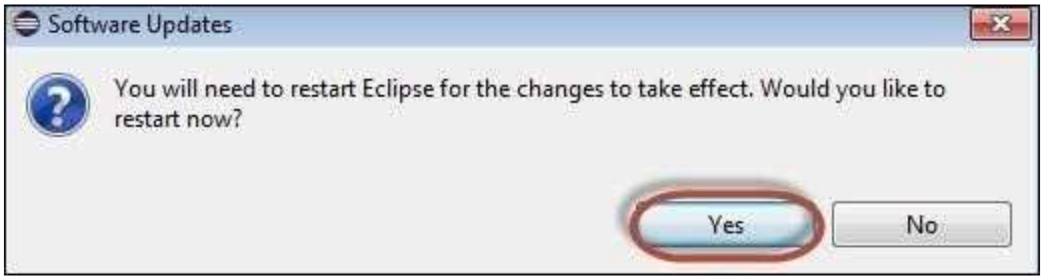
Always run in background

Step 9 − Security Warning pops up as the validity of the software cannot be established. Click 'Ok'.



Warning: You are installing software that contains unsigned content. The authenticity or validity of this software cannot be established. Do you want to continue with the installation?

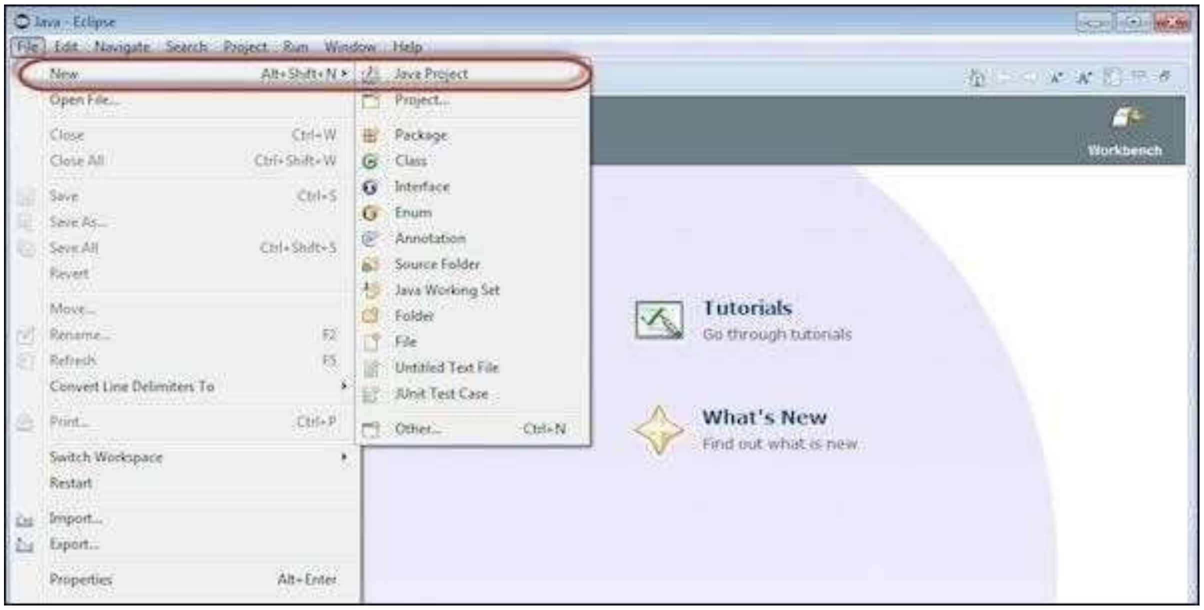
Step 10 − The Installer prompts to restart Eclipse for the changes to take effect. Click 'Yes'.



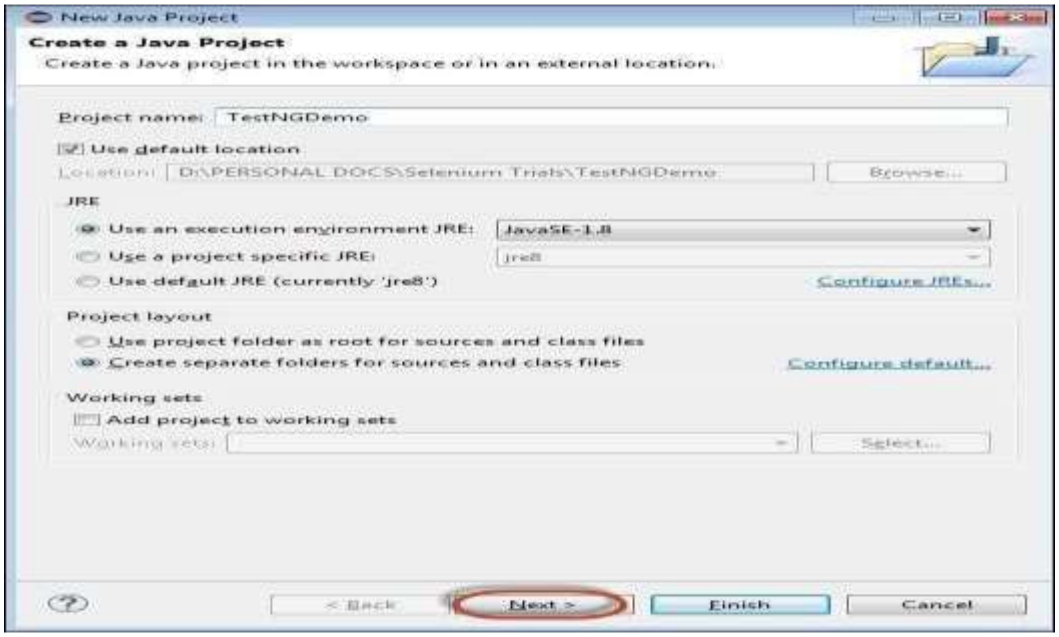
You will need to restart Eclipse for the changes to take effect. Would you like to restart now?

TestNG-Eclipse Setup

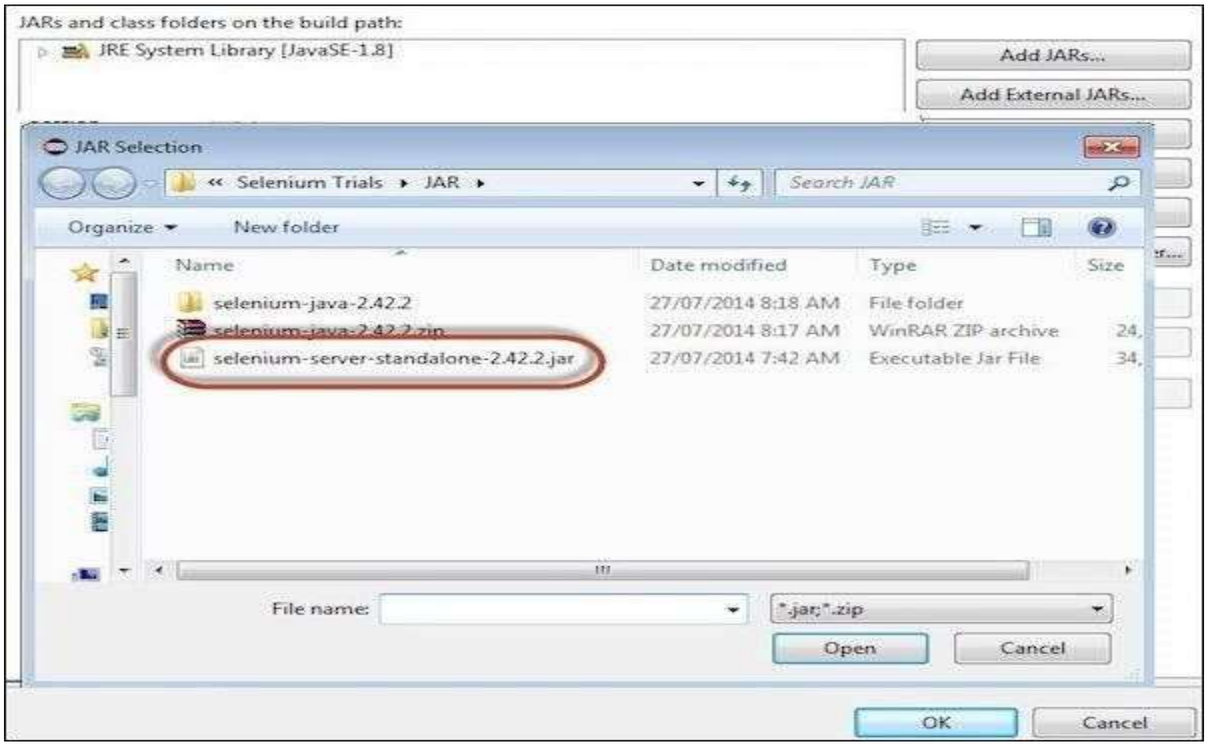
Step 1 − Launch Eclipse and create a 'New Java Project' as shown below.



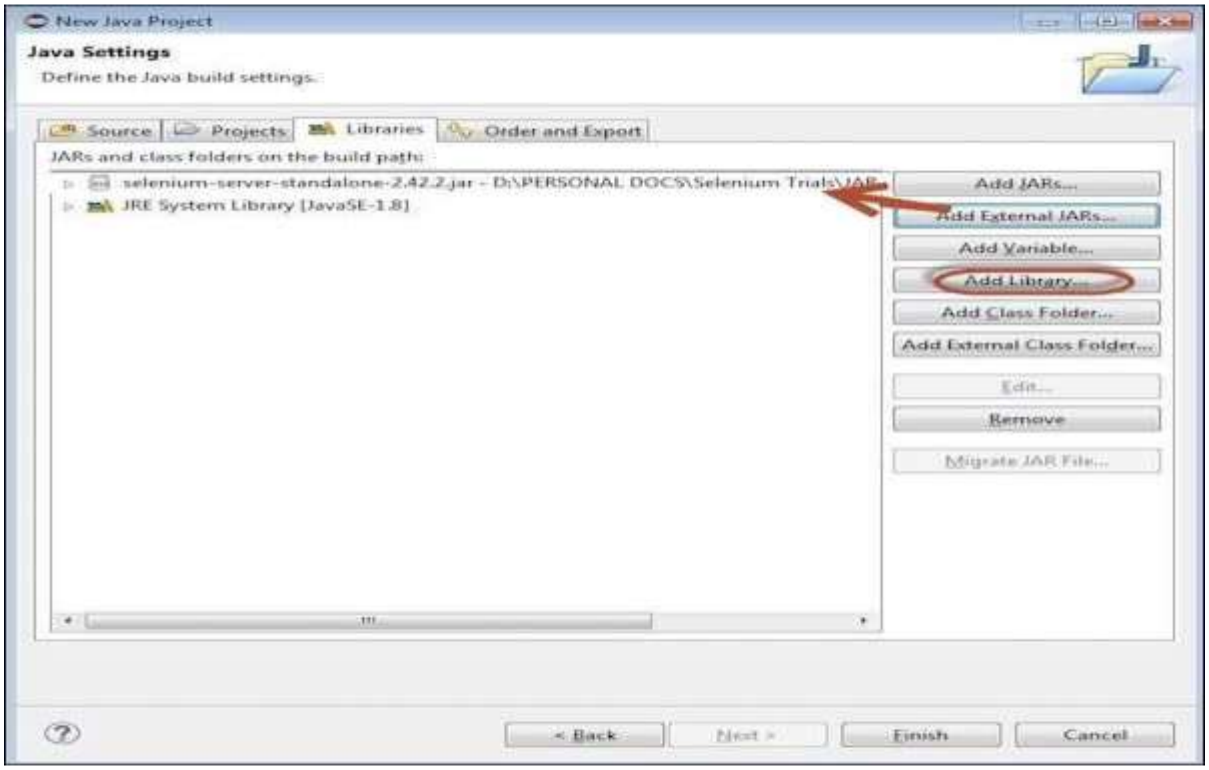
Step 2 − Enter the project name and click 'Next'.


New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
Project name: TestNGDemo
Use default location
Location: D:\PERSONAL DOCS\Selenium Trials\TestNGDemo
Browse...
JRE
Use an execution environment JRE: JavaSE-1.8
Use a project specific JRE: jre8
Use default JRE (currently 'jre8')
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel

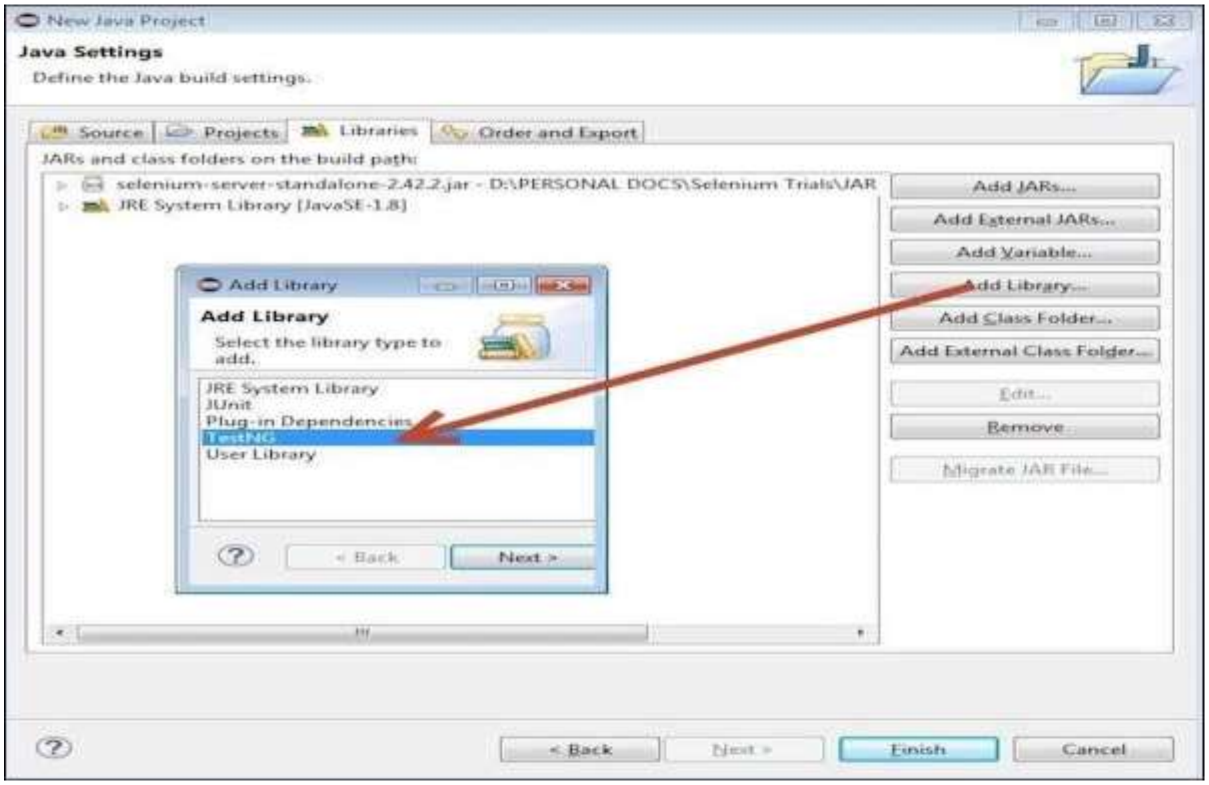
Step 3 − Navigate to "Libraries" Tab and Add the Selenium Remote Control Server JAR file by clicking on "Add External JAR's" as shown below.



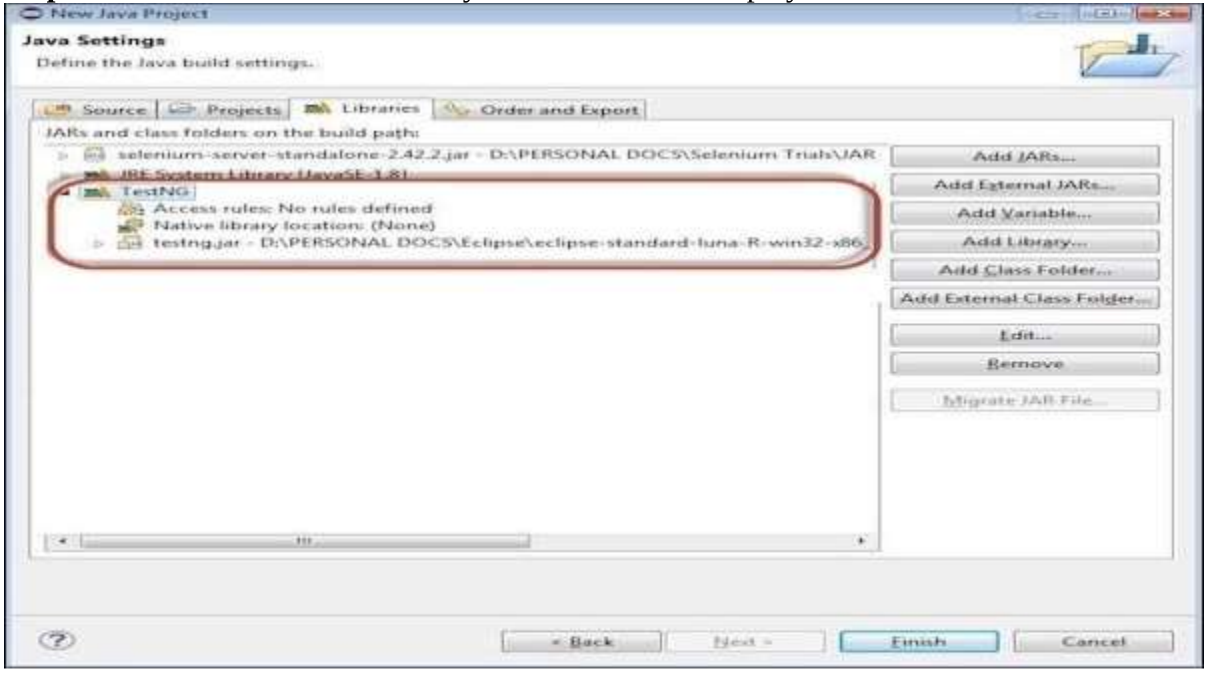
Step 4 − The added JAR file is shown here. Click 'Add Library'.


New Java Project
Java Settings
Define the Java build settings.
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
selenium-server-standalone-2.42.2.jar - D:\PERSONAL DOCS\Selenium Trials\JAR
JRE System Library [JavaSE-1.8]
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
< Back
Next >
Finish
Cancel

Step 5 − The 'Add Library' dialog opens. Select 'TestNG' and click 'Next' in the 'Add Library' dialog box.



Step 6 − The added 'TestNG' Library is added and it is displayed as shown below.


New Java Project

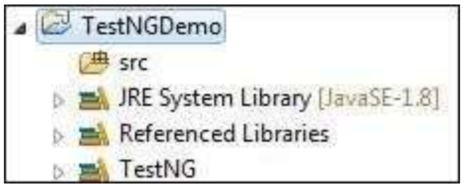
Java Settings

Define the Java build settings.

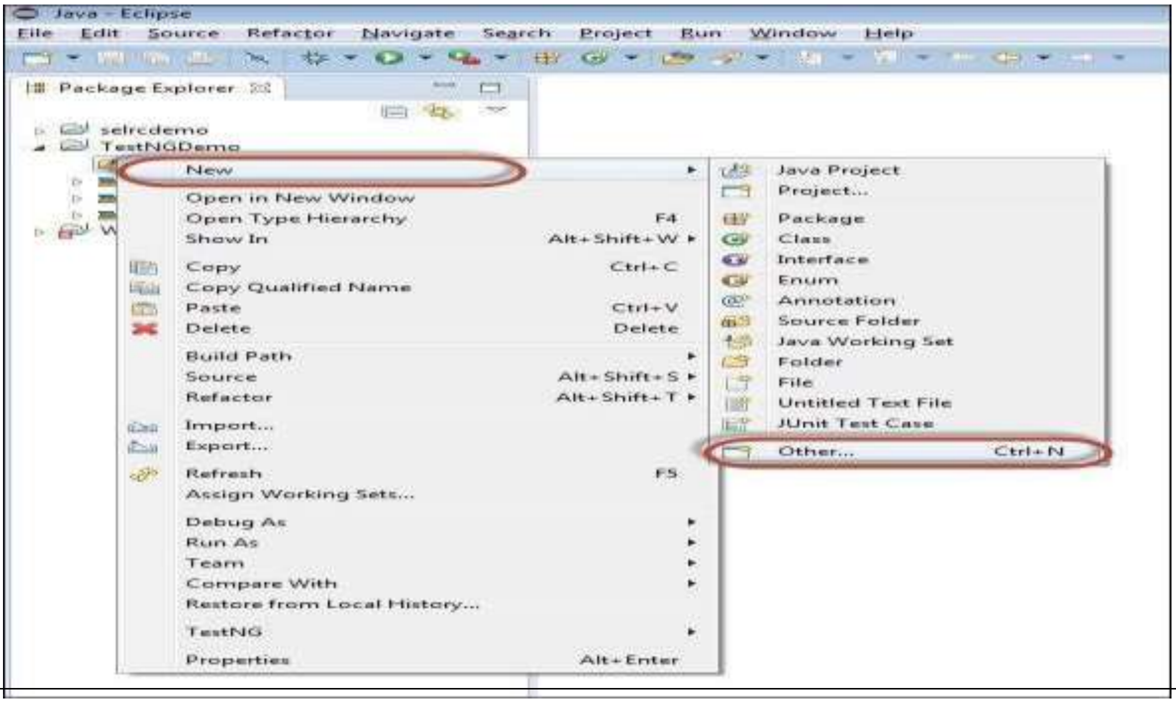
JARs and class folders on the build path:

```
▷ selenium-server-standalone-2.42.2.jar - D:\PERSONAL DOCS\Selenium Trials\JAR
▷ JRE System Library [JavaSE-1.8]
◢ TestNG
    Access rules: No rules defined
    Native library location: (None)
  ▷ testng.jar - D:\PERSONAL DOCS\Eclipse\eclipse-standard-luna-R-win32-x86
```

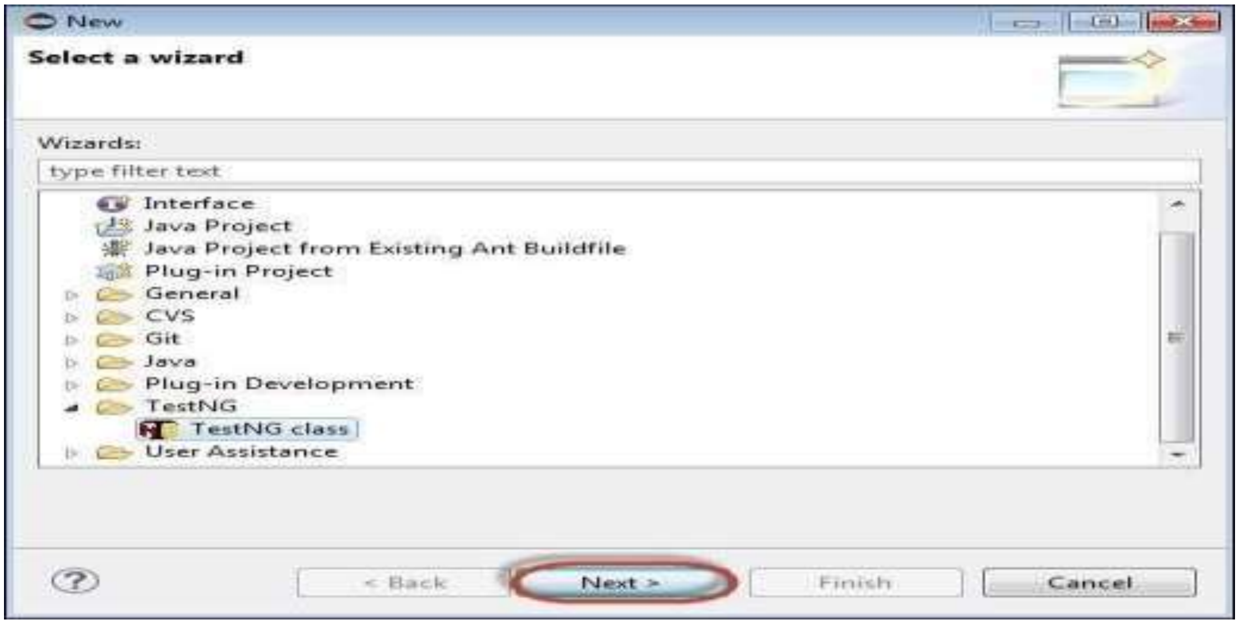
Step 7 − Upon creating the project, the structure of the project would be as shown below.


```
◢ TestNGDemo
    src
  ▷ JRE System Library [JavaSE-1.8]
  ▷ Referenced Libraries
  ▷ TestNG
```

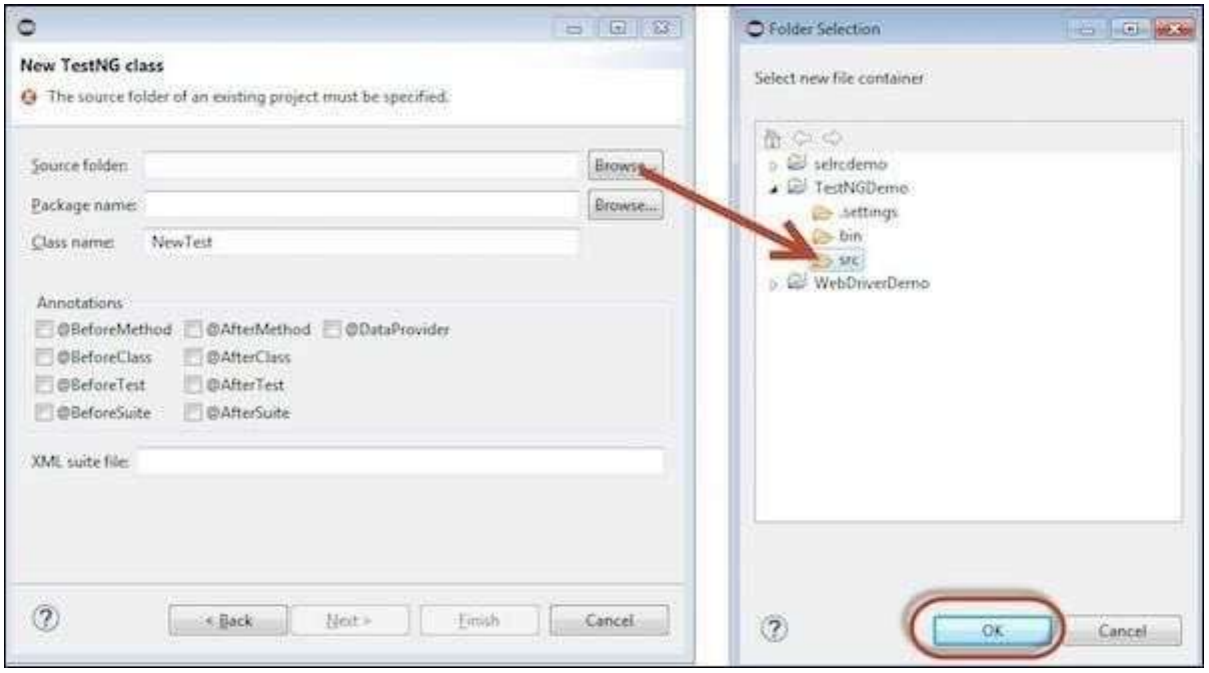
Step 8 − Right-click on 'src' folder and select New >> Other.



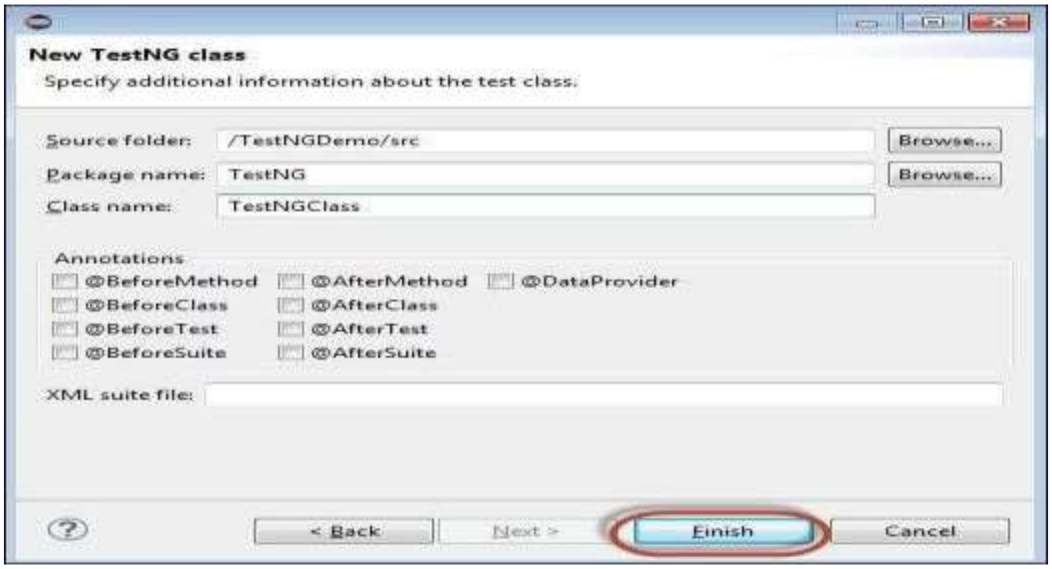
Step 9 − Select 'TestNG' and click 'Next'.



Step 10 − Select the 'Source Folder' name and click 'Ok'.



Step 11 − Select the 'Package name', the 'class name', and click 'Finish'.



New TestNG class

Specify additional information about the test class.

Source folder: /TestNGDemo/src

Package name: TestNG

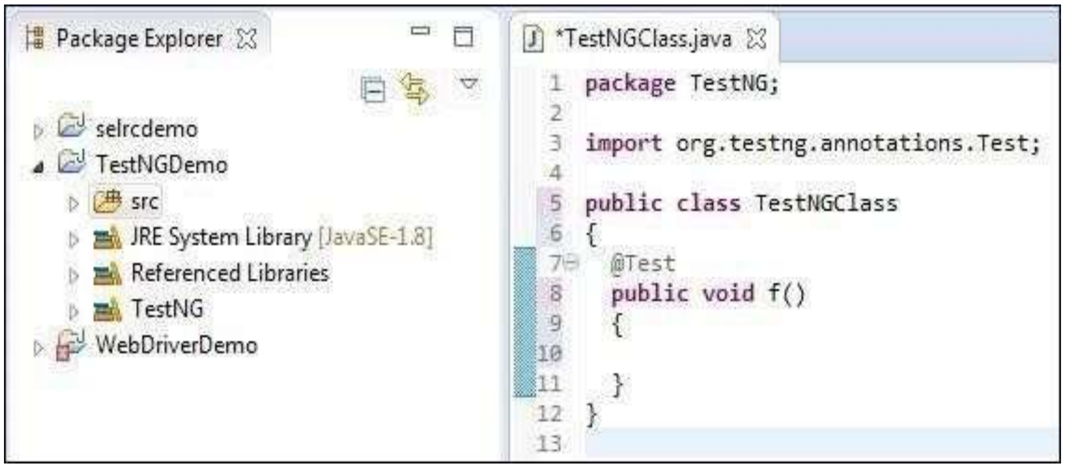
Class name: TestNGClass

Annotations

- [ ] @BeforeMethod
- [ ] @AfterMethod
- [ ] @DataProvider
- [ ] @BeforeClass
- [ ] @AfterClass
- [ ] @BeforeTest
- [ ] @AfterTest
- [ ] @BeforeSuite
- [ ] @AfterSuite

XML suite file:

Step 12 − The Package explorer and the created class would be displayed.



```
package TestNG;

import org.testng.annotations.Test;

public class TestNGClass
{
  @Test
  public void f()
  {

  }
}
```

RESULT: