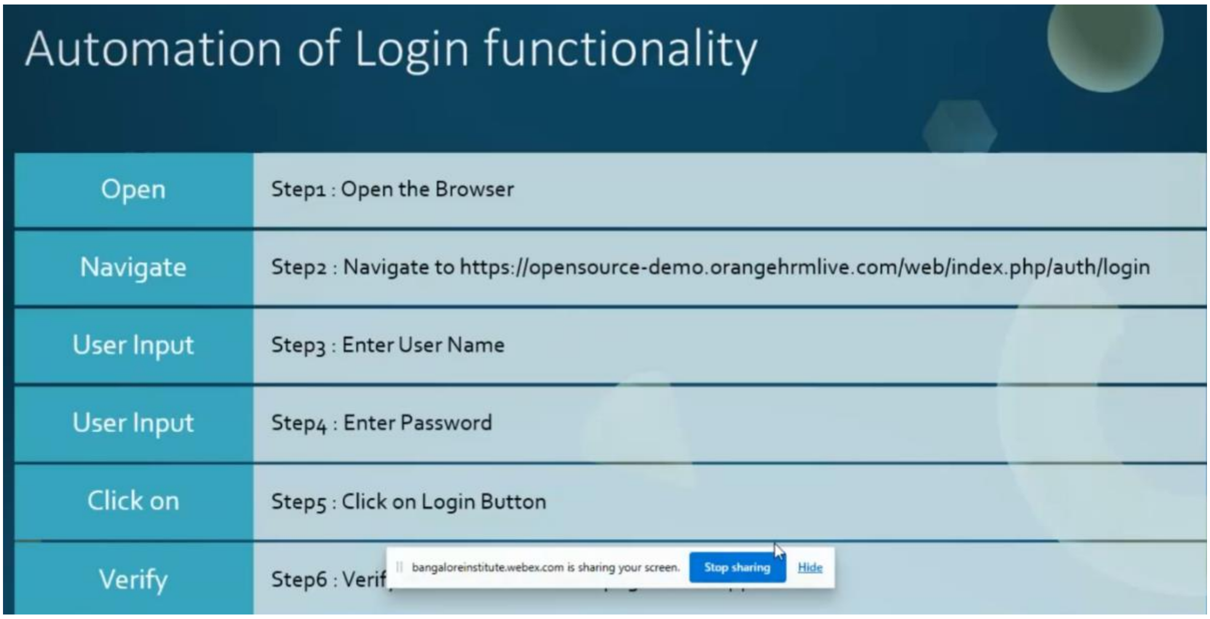
Program 7

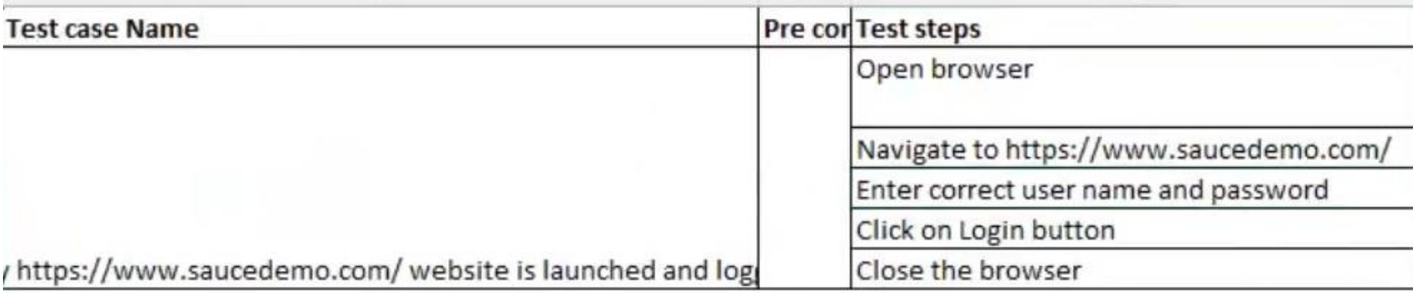
```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class aaa {
public static void main(String[] args) {
WebDriver driver = new ChromeDriver();
driver.get("https://www.saucedemo.com/");
driver.findElement(By.id("user-name")).click();  //just  for  the click function
driver.manage().window().maximize();
driver.findElement(By.id("user-name")).sendKeys("standard_user");
driver.findElement(By.id("password")).sendKeys("secret_sauce1");
 driver.findElement(By.className("submit-button")).click();
 boolean flag = driver.findElement(By.xpath("//button[text()=\"Open Menu\"]")).isDisplayed();
 if (flag == true)
  {System.out.println("Successfully logged in! hurRAY ");}
 /*driver.close(); // terminate test window only
 driver.quit(); //to Terminate all windows and test window */ }
```



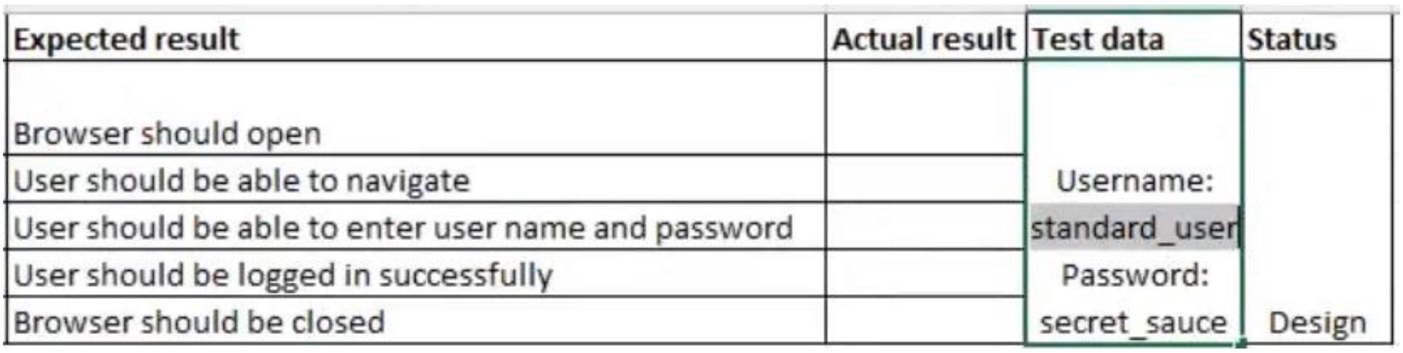
Automation of Login functionality

Open	Step1 : Open the Browser
Navigate	Step2 : Navigate to https://opensource-demo.orangehrmlive.com/web/index.php/auth/login
User Input	Step3 : Enter User Name
User Input	Step4 : Enter Password
Click on	Step5 : Click on Login Button
Verify	Step6 : Verif

bangaloreinstitute.webex.com is sharing your screen.



Test case Name	Pre cor	Test steps
/ https://www.saucedemo.com/ website is launched and log		Open browser
		Navigate to https://www.saucedemo.com/
		Enter correct user name and password
		Click on Login button
		Close the browser



Expected result	Actual result	Test data	Status
Browser should open		Username: standard_user Password: secret_sauce	Design
User should be able to navigate			
User should be able to enter user name and password			
User should be logged in successfully			
Browser should be closed			

Program – 8 Using Selenium IDE

Develop a test suite containing minimum 4 test cases.

Test Suite: **test suite** is a collection of multiple test cases grouped together for execution. It allows testers to run multiple tests sequentially or in parallel to validate different functionalities of a web application.

How to Do:

Step 1: Install Selenium IDE

1. Open **Google Chrome**.
2. Go to the **Chrome Web Store** and search for **Selenium IDE**.
3. Click **"Add to Chrome"** and install the extension.

Step 2: Create a New Project

1. Open **Selenium IDE** from your Chrome extensions.
2. Click **"Create a New Project"** and give it a suitable name.

Step 3: Record a Test Case

1. Click on the **red record button** (top right corner).
2. A new Chrome tab will open – browse normally and perform actions.
3. Close the tab when done; Selenium IDE will automatically save the recorded steps.

Step 4: Create Four Test Cases

Perform any four different actions while recording, such as:

- **Login** – Enter credentials and sign in.
- **Search** – Use a search bar to find an item.
- **Add to Cart** – Select a product and add it to the cart.
- **Logout** – Sign out of the application

Step 5: Run the Test Suite

1. Click **"Run All"** in Selenium IDE to execute all test cases sequentially.
2. Verify that each step passes without errors.

Now we setup testing:

For simple setup follow these:

1.Navigate to https://mvnrepository.com/artifact/org.testng/testng/7.11.0

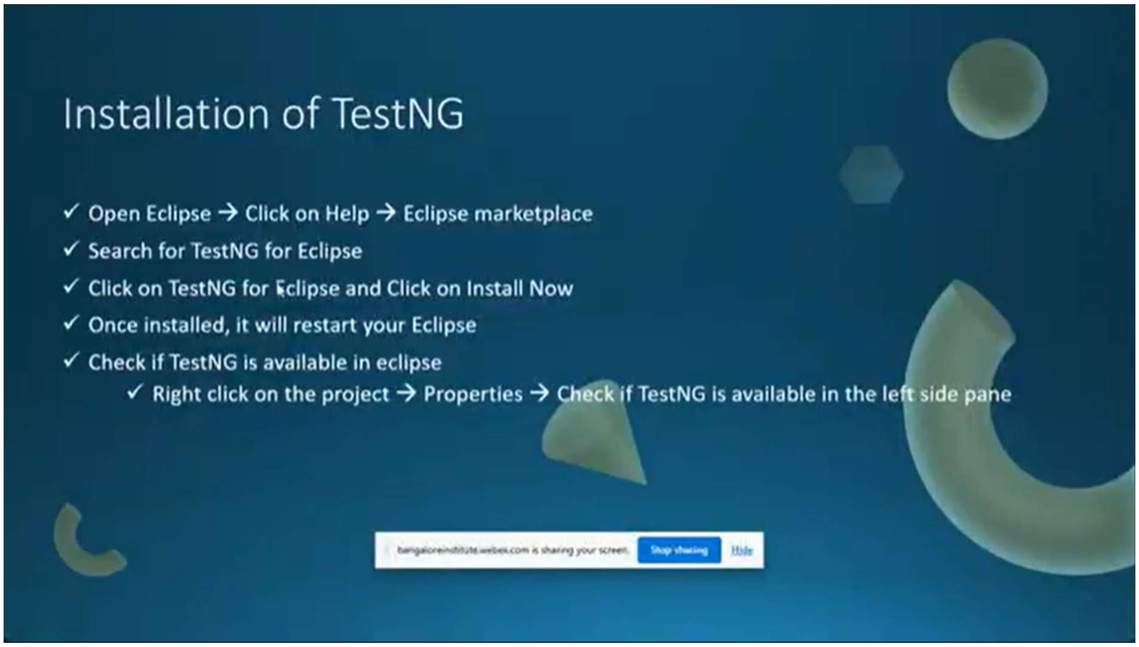
2.Copy the maven dependency

3.In eclipse, copy this under dependencies tag after selenium dependency inside pom.xml

4.Right click on the project --> Maven --> Update project --> Save

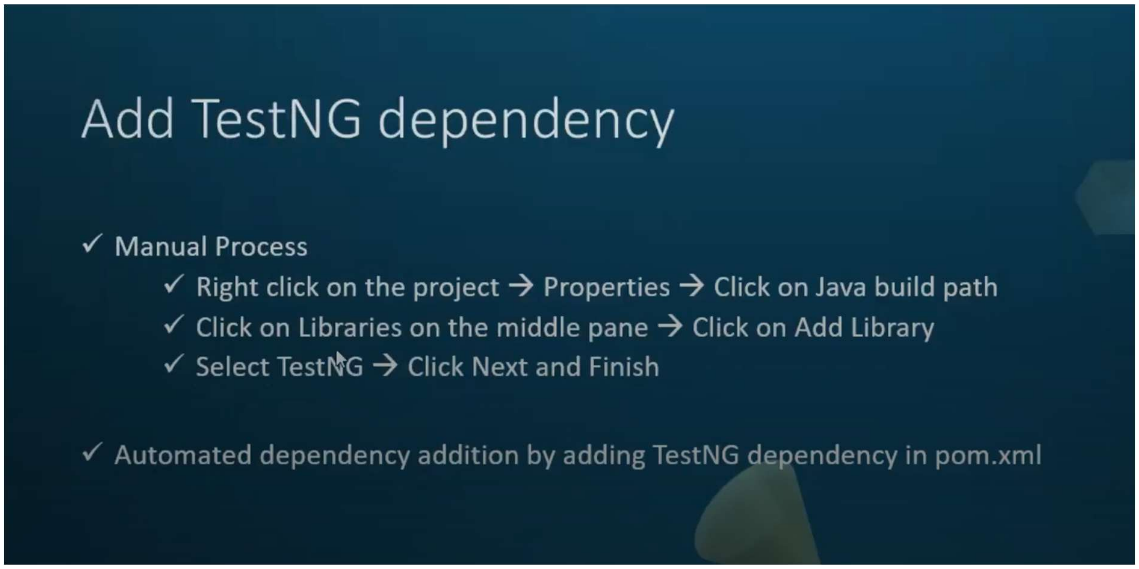
Same as how we did for adding selenium.

Below is the hard way (Manual way)



Installation of TestNG

- ✓ Open Eclipse → Click on Help → Eclipse marketplace
- ✓ Search for TestNG for Eclipse
- ✓ Click on TestNG for Eclipse and Click on Install Now
- ✓ Once installed, it will restart your Eclipse
- ✓ Check if TestNG is available in eclipse
  - ✓ Right click on the project → Properties → Check if TestNG is available in the left side pane

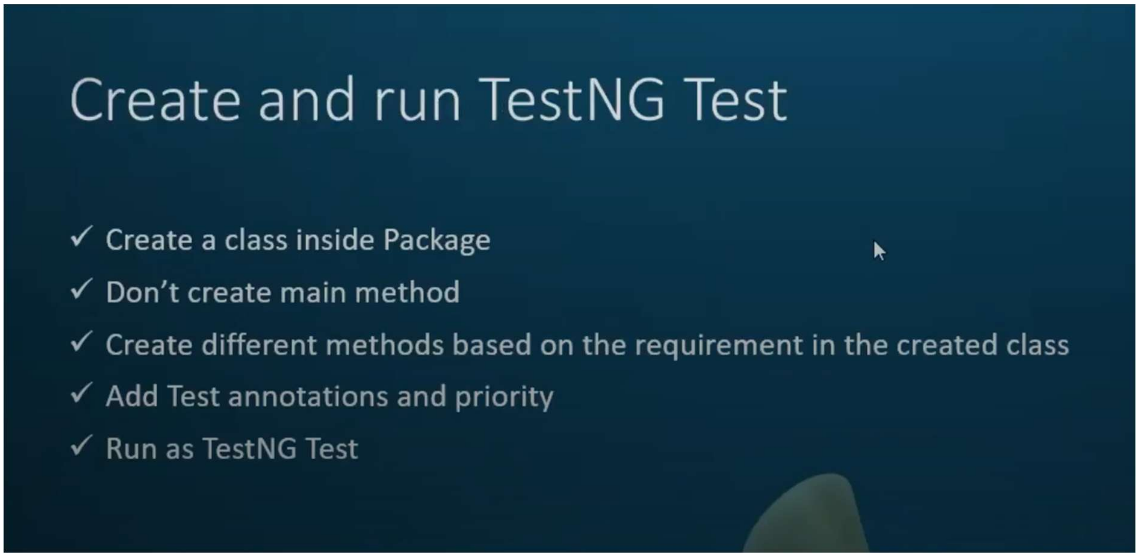


Add TestNG dependency

- ✓ Manual Process
  - ✓ Right click on the project → Properties → Click on Java build path
  - ✓ Click on Libraries on the middle pane → Click on Add Library
  - ✓ Select TestNG → Click Next and Finish

- ✓ Automated dependency addition by adding TestNG dependency in pom.xml

Finally



Create and run TestNG Test

- ✓ Create a class inside Package
- ✓ Don’t create main method
- ✓ Create different methods based on the requirement in the created class
- ✓ Add Test annotations and priority
- ✓ Run as TestNG Test

Just a Side note: Always create a java class for each test case.

TEST CASE – Example *(Just to check)*

```
package seleniumDay3;

import org.testing.annotations.Test;

public class testNGFramework {

        @Test(priority=1)

        void openBrowser(){

                System.out.println("Open browser");

        }


        @Test(priority=2)

        void login(){

                System.out.println("Login");

        }


        @Test(priority=3)

        void close(){

                System.out.println("Close browser");

        }

}
```

Program – 9 Test suite

Conduct a test suite for any two websites.

All the rest of the programs will be done in testing environment.

TEST CASE - 1

```
package seleniumDay3;

import org.testing.annotations.Test;
import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;


public class loginWithTestNG {

    WebDriver driver;

    @Test(priority=1)

    void openBrowser(){

        WebDriver driver = new ChromeDriver();

        driver.get("https://www.saucedemo.com");

        driver.manage().window().maximize();

    }


    @Test(priority=2)

    void login(){


        driver.findElement(By.id("user-name")).click();

        driver.findElement(By.id("user-name")).sendKeys("standard_user");

        driver.findElement(By.name("password")).sendKeys("secret_sauce");


        driver.findElement(By.name("login-button")).click();

        boolean flag driver.findElement(By.xpath("//button[text()- \"Open
Menu\"]")).isDisplayed(); if(flag==true){

            System.out.println("User is successfully logged in");

        }

        else{

            System.out.println("User is not logged in successfully");

        }

    }
```

```
        @Test(priority=3)
        void close(){
                driver.close();
        }
}
```

TEST CASE - 2

```
package seleniumDay3;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

public class LoginDemoBlaze {
    WebDriver driver;

    @BeforeTest
    void openBrowser() {
        driver = new ChromeDriver();
        driver.get("https://www.demoblaze.com/");
        driver.manage().window().maximize();
    }
    @Test(priority = 1)
    void login() throws InterruptedException {
        driver.findElement(By.id("login2")).click();
        Thread.sleep(2000);  // Wait for the login modal to appear

        driver.findElement(By.id("loginusername")).sendKeys("testuser");
        driver.findElement(By.id("loginpassword")).sendKeys("testpassword");
        driver.findElement(By.xpath("//button[text()='Log in']")).click();

        Thread.sleep(3000);  // Wait for login to process

        WebElement logoutButton = driver.findElement(By.id("logout2"));
        if (logoutButton.isDisplayed()) {
            System.out.println("User is successfully logged in");
        } else {
            System.out.println("User login failed");
        }
    }

    @AfterTest
    void closeBrowser() {
        driver.quit();
    }
}
```

Beforetest and Aftertest not actually needed, can be replaced to test(priority = some number)

Now you can run the test cases individually or select multiple test cases to run as a suite.

For that **select the above two test cases, select TestNG and select convert to TestNG.**

It creates a xml file for the suite **automatically**.

You can use test case-1 for another website and that would make another test suite for program-9, the same can be done using program-8 using selenium ide, just add two proper testcases for different websites and that would make a test suite but the difference will be that it will be in selenium IDE.

Program -10 Test Scripts

Develop and test a program to login a specific web pade using selenium test scripts

Use the test case-1 from previous prog to check for login or just use prog 7's code. All do same thing. Still the code is below.

```
package seleniumDay3;
import org.testing.annotations.Test;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class loginWithTestNG {
    WebDriver driver;
    @Test(priority=1)
    void openBrowser(){
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.saucedemo.com");
        driver.manage().window().maximize();
    }
    @Test(priority=2)
    void login(){

        driver.findElement(By.id("user-name")).click();
        driver.findElement(By.id("user-name")).sendKeys("standard_user");
        driver.findElement(By.name("password")).sendKeys("secret_sauce");

        driver.findElement(By.name("login-button")).click();
        boolean flag driver.findElement(By.xpath("//button[text()- \"Open
Menu\"]")).isDisplayed(); if(flag==true){
            System.out.println("User is successfully logged in");
        }
        else{
            System.out.println("User is not logged in successfully");
        }
    }

    @Test(priority=3)
    void close(){
        driver.close();
    }
}
```

Program -11 Test Scripts

Develop and test a program to provide total number of objects present available on the web page using selenium test scripts.

```
package seleniumDay3;


import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

import java.util.List;


public class CountObjects {

  WebDriver driver;


  @Test(priority = 1)

  void openBrowser() {

    driver = new ChromeDriver();

    driver.get("https://www.demoblaze.com/");

    driver.manage().window().maximize();

  }


  @Test(priority = 2)

  void countObjects() {

    List<WebElement> elements = driver.findElements(By.xpath("//*"));

    System.out.println("Total number of elements on the page: " + elements.size());

  }
```

```
    @Test(Priority = 3)
    void closeBrowser() {
        driver.quit();
    }
}
```

Program -12 Practical exercise and wrap-up

Build test suite with suitable application and complete end to end automation process, discussion on best practices and Q&A.

Just use the same program as prog 10

```
package seleniumDay3;

import org.testing.annotations.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;


public class loginWithTestNG {
	WebDriver driver;
	@Test(priority=1)
	void openBrowser(){
		WebDriver driver = new ChromeDriver();
		driver.get("https://www.saucedemo.com");
		driver.manage().window().maximize();
	}
```

```
    @Test(priority=2)
    void login(){

        driver.findElement(By.id("user-name")).click();
        driver.findElement(By.id("user-name")).sendKeys("standard_user");
        driver.findElement(By.name("password")).sendKeys("secret_sauce");

        driver.findElement(By.name("login-button")).click();
        boolean flag driver.findElement(By.xpath("//button[text()- \"Open Menu\"]")).isDisplayed(); if(flag==true){
            System.out.println("User is successfully logged in");
        }
        else{
            System.out.println("User is not logged in successfully");
        }
    }

    @Test(priority=3)
    void close(){
        driver.close();
    }
}
```

If you want to add log out functionality, you can do so. Nearly all prog 9, 10, 11, 12 can be done using simple open browser, login and close browser in saucedemo website or demoblaze.