

Program 7:

Designing Test Cases: Test case design and Conversion of Mapping and Templates using selenium tool and Manual test cases mapping with Selenium test cases

```
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_sau ce1");
        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

Test case Name	Pre condition	Test steps
		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			
User should be able to navigate		Username:	
User should be able to enter user name and password		standard_user	
User should be logged in successfully		Password:	
Browser should be closed		secret_sauce	Design

Program 8:

Using Selenium IDE, Develop a test suite containing minimum 4 test cases.

Test Suite: It 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

bengaloreinstituteofweb.com is sharing your screen.

Stop sharing

Hide

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

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

TEST CASE –Example

```
package seleniumDay3;
import org.testng.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");  
}  
}
```