**Program – 6 Installation of Selenium  
Setting up Selenium RC and Selenium Web Driver**Step 1 : Create a New Maven Project from File Menu > New.

Step 2 : To create a Maven project, you need to specify Group ID and Artifact ID, then finish.

**A screenshot of a computer

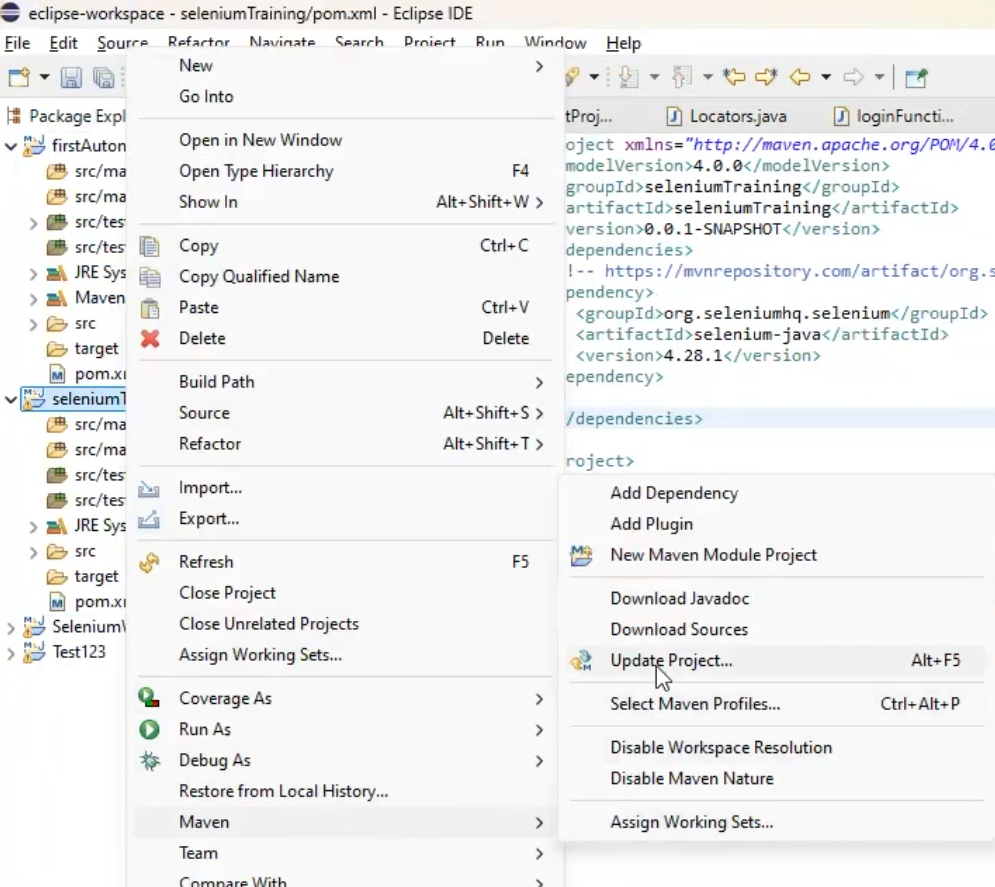
AI-generated content may be incorrect.**

Step 3 : Add the following dependency within the <dependencies> section of the pom.xml file, ensuring that it is placed before the closing </project> tag. This dependency is necessary for integrating Selenium with Java in the project.

**A screen shot of a computer

AI-generated content may be incorrect.**

Step 4 : Right click on your project > Maven > Update Project

****

Step 5: Select the following check boxes

**A screenshot of a computer

AI-generated content may be incorrect.**

Step 6 : Right click on src/test/java and create a new package and class with any name.

Select Public static void main while creating class.

Type the following code in your class.

**A screenshot of a computer

AI-generated content may be incorrect.**

**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.**

Assuming you followed the steps in 6th program, we can continue with this.

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)

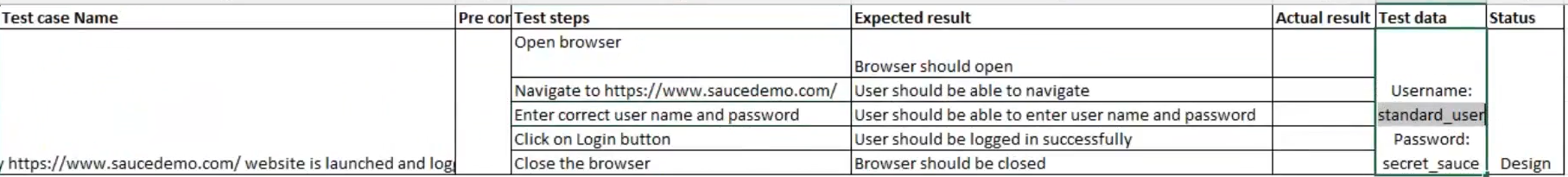
{System.out.println("Successfully logged in! hurRAY ");}

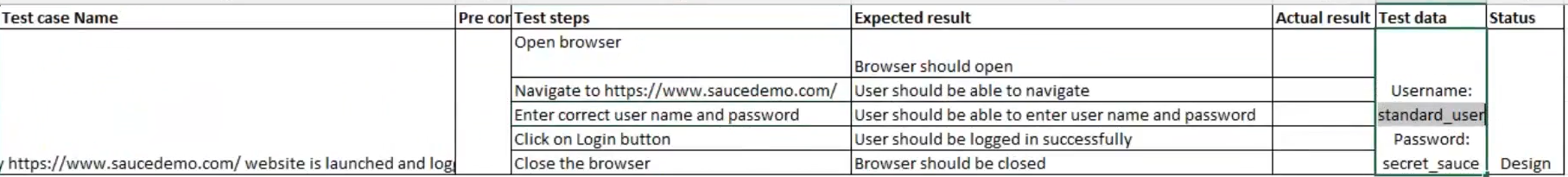
/\*driver.close(); // terminate test window only

driver.quit(); //to Terminate all windows and test window \*/ }

**A screenshot of a computer

AI-generated content may be incorrect.**

****

****

**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 **Mozilla Firefox**.
2. Select extensions and search for selenium extension.
3. Click **"Add extension"** and install the extension.

**Step 2: Create a New Project**

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

**Step 3: Record a Test Case**

A screenshot of a computer

AI-generated content may be incorrect.

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**

**A screenshot of a computer

AI-generated content may be incorrect.**

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

Then make the test suite:  
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**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.

For eg:  
goto: <https://www.automationexercise.com/test_cases>  
  
take these test cases, minimum 4 please.

If the test suites are not getting created properly, please do very simple tasks in the test cases and close the browser properly. If it is getting stuck at any test case, remake it because it can get stuck and can ruin the test suite.

Now we setup testNG:

Update pom.xml and update the project(Can be skipped since after you add the test Ng library into java build path, it should automatically add itself into pom.xml)

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

2.open eclipse marketplace

A screenshot of a computer

AI-generated content may be incorrect.Search for testng and install it:  
  
A screenshot of a computer

AI-generated content may be incorrect.

It is supposed to restart after this. To check if it is installed go here:  
A screenshot of a computer

AI-generated content may be incorrect.

Now open properties of the project and go to the java build path  
  
A screenshot of a computer

AI-generated content may be incorrect.

Click on add library and select testNG and click next. Then Apply and close.  
A computer screen shot of a program

AI-generated content may be incorrect.

Update the project(Update Maven)

**TEST CASE – Example** *( Just to check if TestNG is working or not)*

package s;

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");

}

}

**Program – 9 Test suite**

**Conduct a test suite for any two websites.**

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

TEST CASE - 1

package test2;

import org.testng.annotations.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.By;

public class test1 {

WebDriver driver = new ChromeDriver();

*@Test*(priority=1)

void openBrowser(){

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("a");

driver.findElement(By.id("loginpassword")).sendKeys("a");

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();

}

}

NOTE:

Beforetest and Aftertest not actually needed, can be replaced to test(priority = some number). Also after running the test case in demoblaze, logout please, or else you will try to login where you are already logged in. Does that make sense?

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

A screenshot of a computer program

AI-generated content may be incorrect.

For that **select the above two test cases and right click, select TestNG and select convert to TestNG.** (You can hold shift key and then click on instances to select multiple test cases at the same time)

A screenshot of a computer

AI-generated content may be incorrect.

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

A screenshot of a computer

AI-generated content may be incorrect.

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 test2;

import org.testng.annotations.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.By;

public class test1 {

WebDriver driver = new ChromeDriver();

*@Test*(priority=1)

void openBrowser(){

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.**

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();

}

}

You can do the same for any website, try for amazon if you want to.

**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.testng.annotations.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class loginWithTestNG {

WebDriver driver = new ChromeDriver();

@Test(priority=1)

void openBrowser(){

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.