EXPERIMENT NO. 7

Aim: To write test cases for black box testing

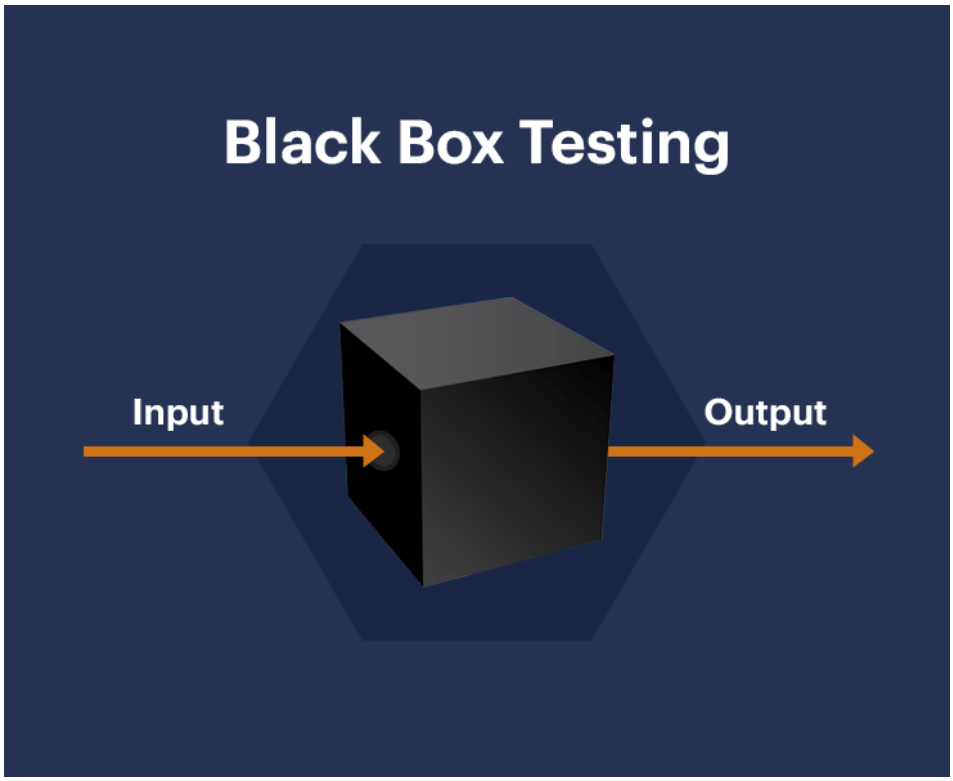
Requirements: Windows/MAC/LINUX O.S, Compatible version of JDK, Eclipse and Selenium

Problem statement: To design test cases for black box testing and implement using Eclipse and Selenium.

Theory:

What is Black Box Testing?

Black box testing is also known as opaque technique, behavioral testing, functional testing, and closed-box testing is a type of **software testing**. When we enter a topic to search on the search engine, we type out the topic and enter search. The result is obtained thereafter without looking at the internal structure or working. This is an appropriate example of black-box testing.



How is Black Box Testing done?

The steps for carrying out Black Box Testing are as follows:

- At first, the application to be tested is studied to find out the requirements and specifications. The SRS (Software Requirement Specification) document should be maintained with accuracy.
- The inputs and test scenarios are evaluated. Efficient and time-saving techniques are incorporated.

- Test cases are generated. These test cases are made in such a way that the input range is maximum.
- The test cases are then processed to obtain the output. The generated output is compared with the expected output to understand the success of the result.
- If there are unsuccessful steps, they are sent to the software development teams for fixing.
- The defects are fixed.
- Run the tests again for confirmation.

Techniques of Black Box Testing

Equivalence Partitioning:

As the name suggests, the inputs are partitioned into groups or more literally partitions. Only one input from every group is tested to find the results. The inputs are usually numeric values or a set of values or Boolean conditions. For example, if the field accepts an integer in the range 1 and 20 then:

Valid Equivalence Class Partition: 1 to 20 inclusive.

Invalid Equivalence Class Partition: Less than 1 or more than 20, decimal numbers or alphabets and other non-numeric characters.

Boundary Value Testing:

In boundary value analysis the answers are within specific boundaries. The two ends, the inner and the outer limits are considered in this type of testing. For example, an offer is valid for customers between the ages of 18 and 30 only. Therefore other values such as 17, 18, 30 or 31 can be tested to check whether the inputs are accepted.

Test Procedure:
The test procedure of black box testing is a kind of process in which the tester has specific knowledge about the software's work, and it develops test cases to check the accuracy of the software's functionality.

It does not require programming knowledge of the software. All test cases are designed by considering the input and output of a particular function. A tester knows about the definite output of a particular input, but not about how the result is arising. There are various techniques used in black box testing for testing like decision table technique, boundary value analysis technique, state transition, All-pair testing, cause-effect graph technique, equivalence partitioning technique, error guessing technique, use case technique and user story technique.

Test Cases:
Test cases are created considering the specification of the requirements. These test cases are generally created from working descriptions of the software including requirements, design parameters, and other specifications. For the testing, the test designer selects both positive test scenario by taking valid input values and adverse test scenario by taking invalid input values

to determine the correct output. Test cases are mainly designed for functional testing but can also be used for non-functional testing. Test cases are designed by the testing team, there is not any involvement of the development team of software.

A basic example of test case design:
Title: Login to the website or app
Description: User should be able to successfully log in to their account on the website/app
Preconditions: User must already be registered and use their correct login details
Assumptions: They are using a supported device or browser to log in
Test Steps:

1. Open website or app
2. Enter the username and password in the appropriate fields
3. Click “login”

Expected Result: The user should log in successfully.
To execute following test case we need a tool which can perform the black box testing. There is variety of tool which are available online. But for our practical Selenium tool has been used.

ECLIPSE:

Eclipse is an integrated development environment that is used in computer programming.

It is the mostly widely use Java IDE and contains a base workspace and an extensible plug-in system for customising the environment.

The platform has been designed to build integrated web and application development tooling.

It is designed to not offer a huge amount of end user functionality but the value of the platform comes with its ability to encourage the rapid development of integrated features based on a plug-in model.

Eclipse provides a common user interface model for working with tools and is designed to run on multiple operating systems.

JUnit is a Java unit testing framework that is useful for creating scalable and repeatable tests. It is provided in Eclipse and can be use with Selenium web driver.

Selenium:

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others.

A browser-driver then executes these scripts on a browser-instance on your device (more on this in a moment).

Selenium WebDriver also known as Selenium 2.0, WebDriver executes test scripts through browser-specific drivers.

Selenium vs. its Counterparts

Features	HP QTP	IBM RFT	TestComplete	Selenium
License	Required	Required	Required	Open Source
Cost	High	High	High	Free
Customer support	Yes	Yes	Yes	Yes; Open source community
Coding skills	Low	Low	High	Very High
Environment support	Only Windows	Only Windows	Windows only (7, Vista, Server 2008 or later OS)	Windows, Linux, Mac
Language support	VB Script	Java and C#	VB Script, JS Script, Delphi Script, C++ & C#	Java, C#, Ruby, Python, Perl & PHP

There are several tools to test other types of applications, but testing of dynamic web applications is done best with Selenium.

Versions and Suite of Tools

*As of April – 2018, the latest release of the tool is **Selenium Version 3.12.0**.*

The Selenium suite comprised the following four components:

Selenium RC, however, was merged with WebDriver and launched anew as Selenium WebDriver with better functionalities.

Output:

How to Set-up Selenium?

The following three software are prerequisite to begin using Selenium.

I. Java → *Programming Language to write scripts*

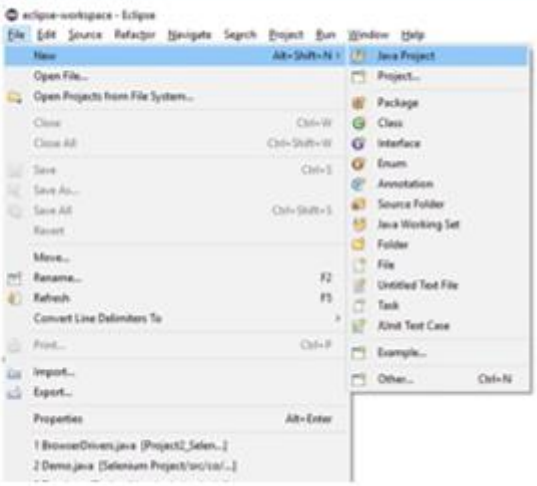
II. Eclipse → *Environment to compile and run scripts*

III. Selenium → *Framework for testing web applications*

How to Set-up Selenium?

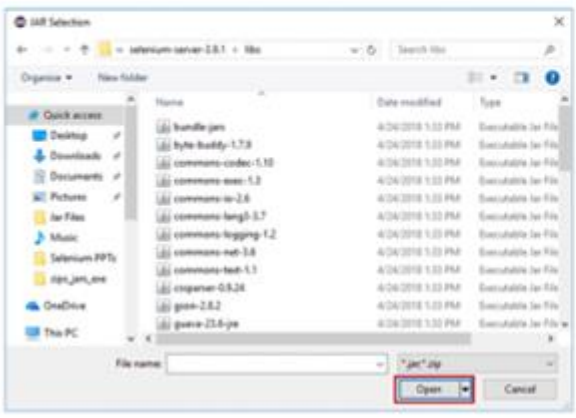
- ✓ After Eclipse is launched, go to

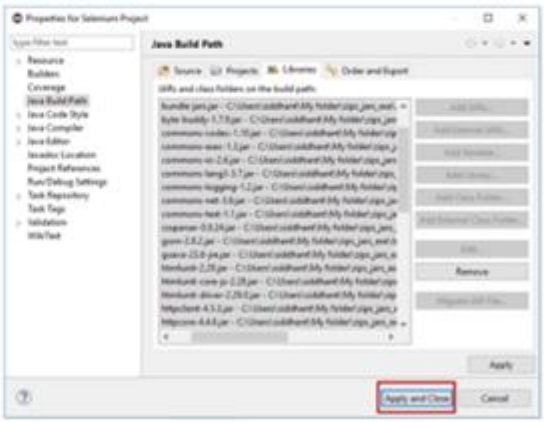
  File → New → Java Project.

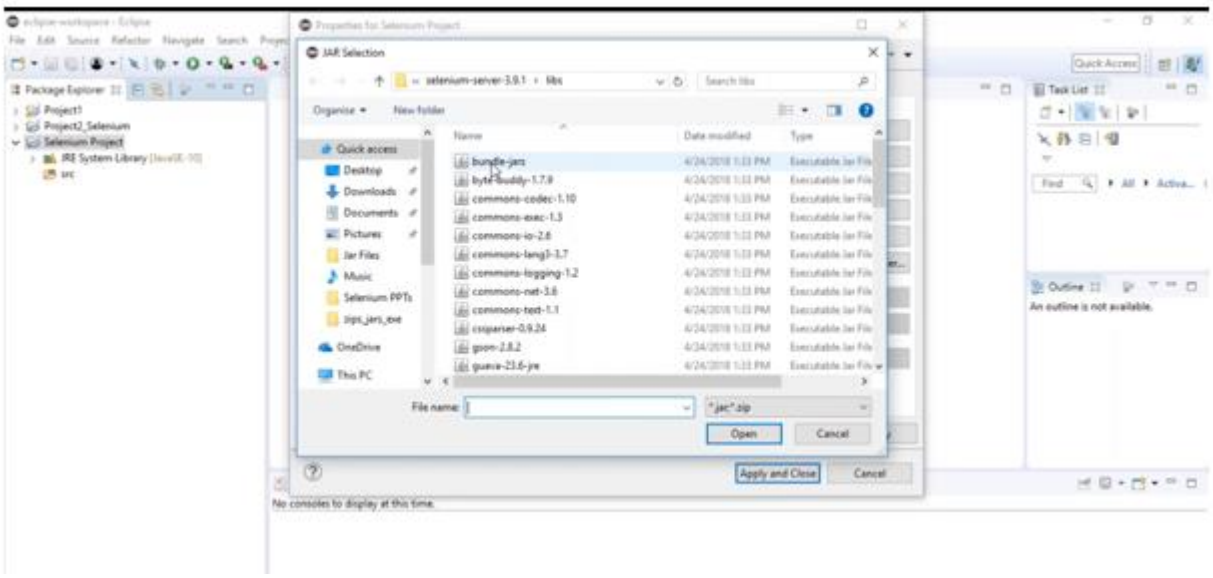


How to Set-up Selenium?

- ✓ Find all the JAR files that we downloaded, select them and click on Open.
- ✓ After you're done adding all the JAR files, Click on Apply and Close.



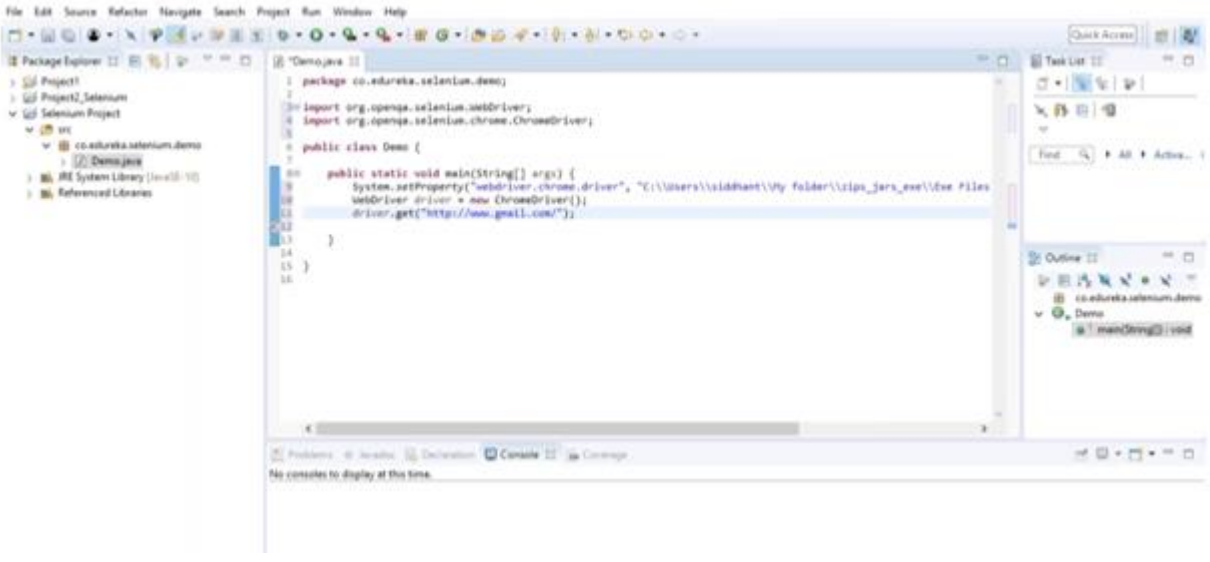




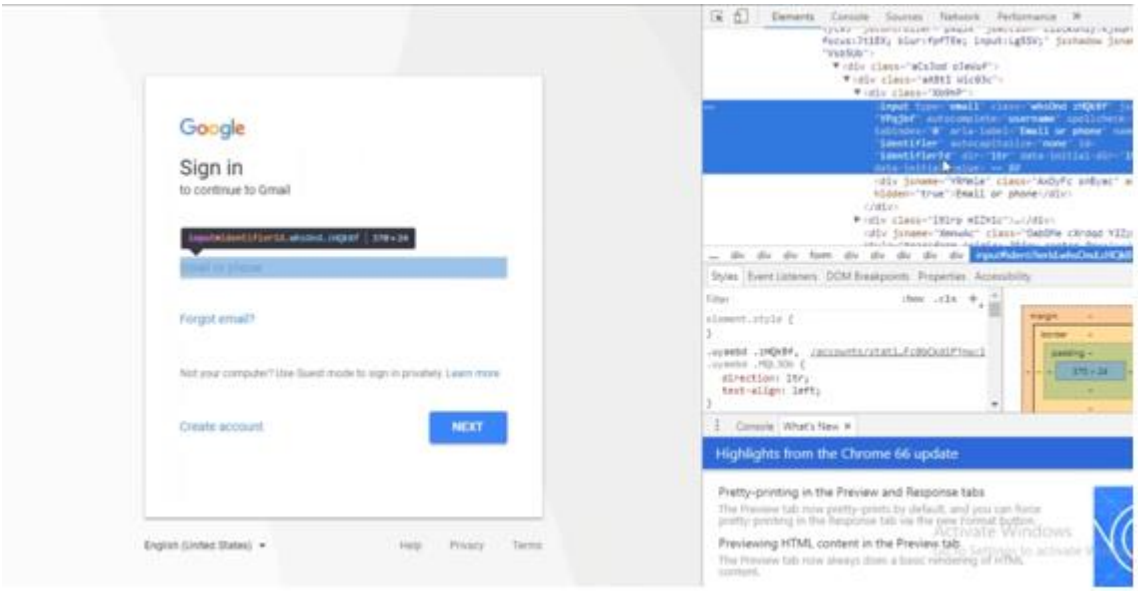
Hands-on in Selenium

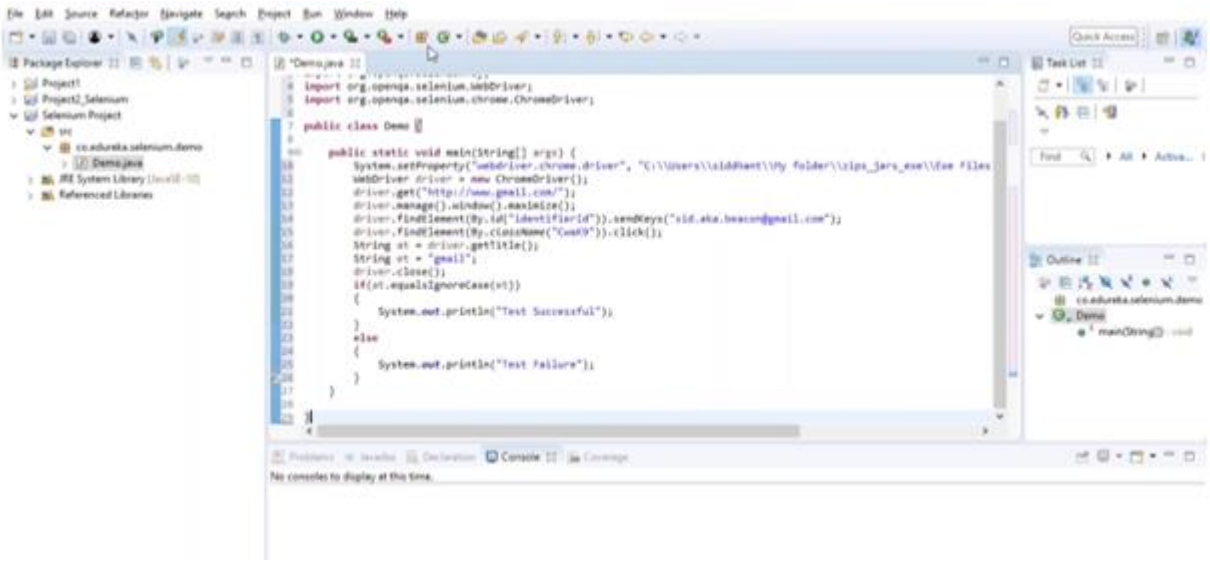
Now that the prerequisites have all been installed and Selenium has been configured for the Eclipse IDE, how about we automate and test Gmail using Google Chrome?

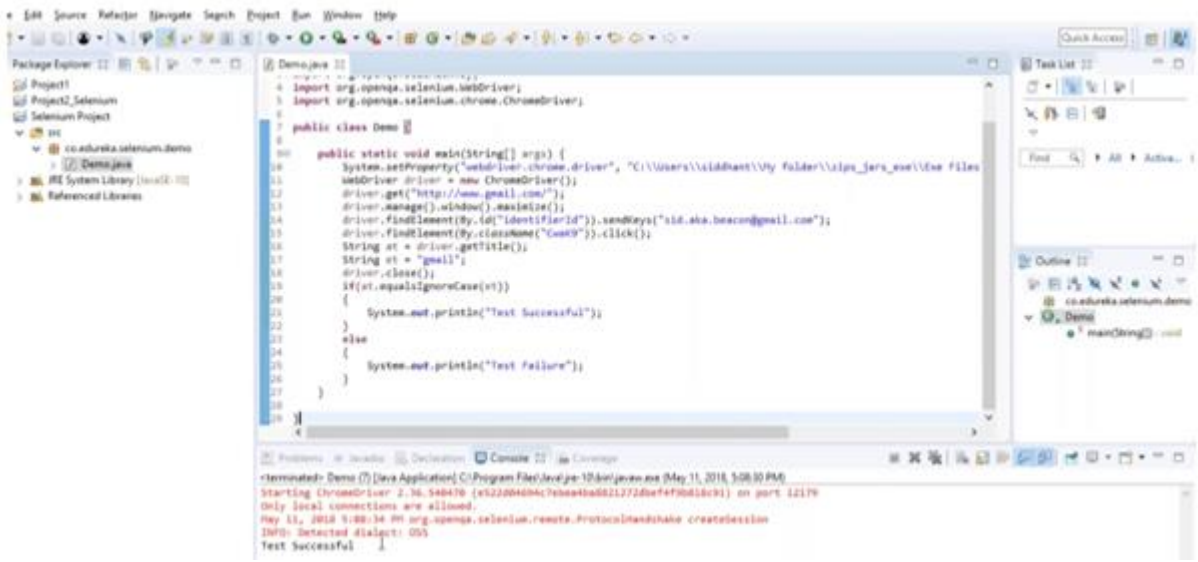
```
package co.edureka.selenium.demo;

import org.openqa.selenium.By;

public class Demo {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\siddhant\\My folder\\zips_jars_exe\\Exe Files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.gmail.com/");
        driver.manage().window().maximize();
        driver.findElement(By.id("identifierId")).sendKeys("xld.eka.beacon@gmail.com");
        Thread.sleep(3000);
        driver.findElement(By.className("CwaK9")).click();
        Thread.sleep(1000);
        String et = driver.getTitle();
        String rt = "gmail";
        System.out.println(et);
        driver.close();
        if(et.equalsIgnoreCase(rt))
        {
            System.out.println("Test Successful");
        }
        else
        {
            System.out.println("Test Failure");
        }
    }

}
```



The following code checks whether Gmail account recognises the registered user id and password. If the registered user id and password is typed then result gives the test is successful otherwise it is unsuccessful.







Conclusion: We have successfully design test cases for black box testing and tested them using Eclipse and Selenium in Java.