Experiment 9

Aim: Design test cases and generate test scripts in Selenium

Learning Objective: Students will able to create unit test case.

Tools: Selenium record and playback, Microsoft Word

Theory:

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases. The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

What is Selenium?

Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages.

Selenium IDE

Next section of the Selenium Automation Testing tutorial covers everything you need to know about Selenium IDE. Shinya Kasatani developed the Selenium Integrated Development Environment (IDE) in 2006. Conventionally, it is an easy-to-use interface that records the user interactions to build automated test scripts. It is a Firefox or Chrome plugin, generally used as a prototyping tool. It was mainly developed to speed up the creation of automation scripts.

IDE ceased to exist in August 2017 when Firefox upgraded to the new Firefox 55 version, which no longer supported Selenium IDE. Applitools rewrote the old Selenium IDE and released a new version recently.

The latest version came with several advancements, such as:

- Reusability of test scripts
- Debugging test scripts
- Selenium side runner
- Provision for control flow statements
- Improved locator functionality

Record and playback in Selenium

The record function in Selenium is a feature that allows users to record their interactions with a web browser and generate a test script based on those interactions. This feature is often used by testers who want to quickly create automated test scripts without having to write code from scratch.

When the record function is activated, Selenium starts recording all user interactions with the browser, including clicks, typing, and other actions. The recorded actions are saved in a script file that can be played back later to perform the same actions automatically.

How to record a test case with Selenium IDE?

1. Download the Selenium IDE Chrome Extension
2. Create a new project: Once you open the Selenium IDE extension in the browser, a popup with various actions will appear. Click on "Create a new project" and give your project a new name. You can also easily change the name later.

3. Create a new test: After naming your project, you'll see a new window with your freshly created project. To create a new test case, you need to provide the name of the test. Once done, save the project. By default, the test case name is set to "Default". You can rename, duplicate, delete and export the test cases by clicking on the title bar link.

4. Start with Record & Playback
   All the preparation steps have been completed, and it's time to start recording your first action. Click on the "REC" button to start recording.
   Once you click on the "REC" button, you will see a popup on your screen. Enter the base URL (which is the URL of the application you want to test) and click "START RECORDING"
   A target website page will then open, where you can record the actions according to the steps of your test case. Once you are done recording the test case steps, go back to Selenium IDE and finish the recording by clicking on the Stop button. Save the project again.
5. Replay your recorded action
6. Click on "Run current test" to see the test being executed.

Once it finishes, you will see the test log with the steps executed along with their timestamps at the bottom.

Playback refers to the process of running the test script that was created during the recording phase. Once the test script is created, the user can run it multiple times to verify that the application behaves as expected. During playback, the test script is executed step-by-step, just like during the recording phase. The test script interacts with the web browser, simulating the actions that were recorded earlier. For example, if the user recorded clicking on a button, during playback, the test script would click on that button again.

The playback function in Selenium is designed to mimic the actions of a human user interacting with a web application. This means that the test script will perform actions like clicking on buttons, filling in forms, and navigating through web pages just as a human user would. This makes it possible to automate a wide variety of test cases, from simple functionality tests to more complex end-to-end tests.

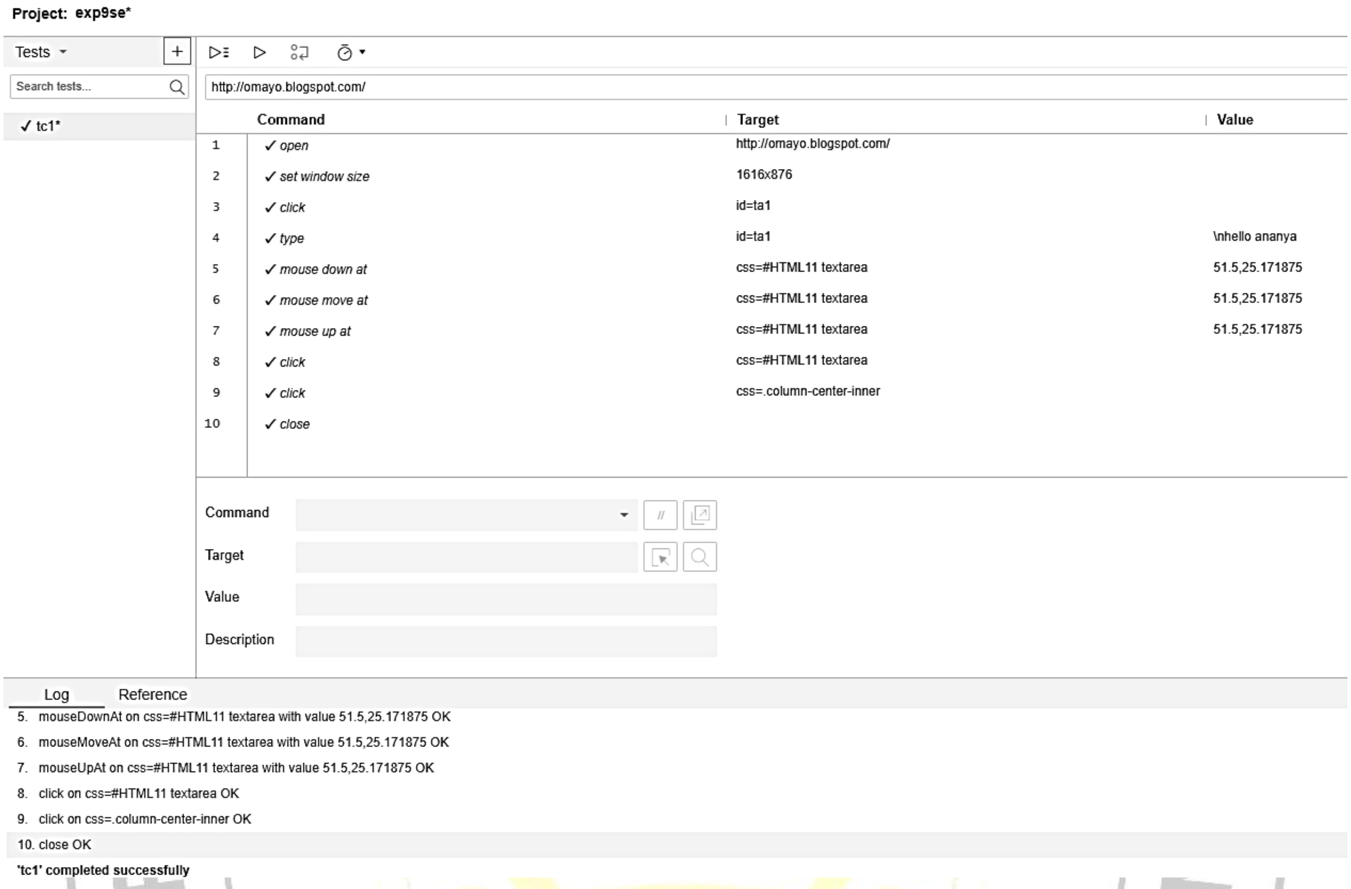
Advantages of using Selenium IDE

- Doesn't require prior code knowledge, or knowledge with other testing frameworks.
- Provides a significant level of flexibility when executing the test cases. For instance, QAs can either run a single test case or execute the entire test suite consisting of multiple test cases.
- Tests are easily exportable into Web Driver and Selenium RC.

Disadvantages of using Selenium IDE

- Only available for Chrome and Firefox browsers.
- Due to multiple architectural factors, it's almost impossible to achieve highly stable tests. One of them is that Selenium IDE relies on details of implementation, meaning that a test will break if the underlying locator will change.
- There are no easy ways to add wait times.
- Test maintenance on a large project will quickly become an issue, since there're no ways to bulk edit test cases.

Implementation: We have created various test cases using Selenium software testing tool

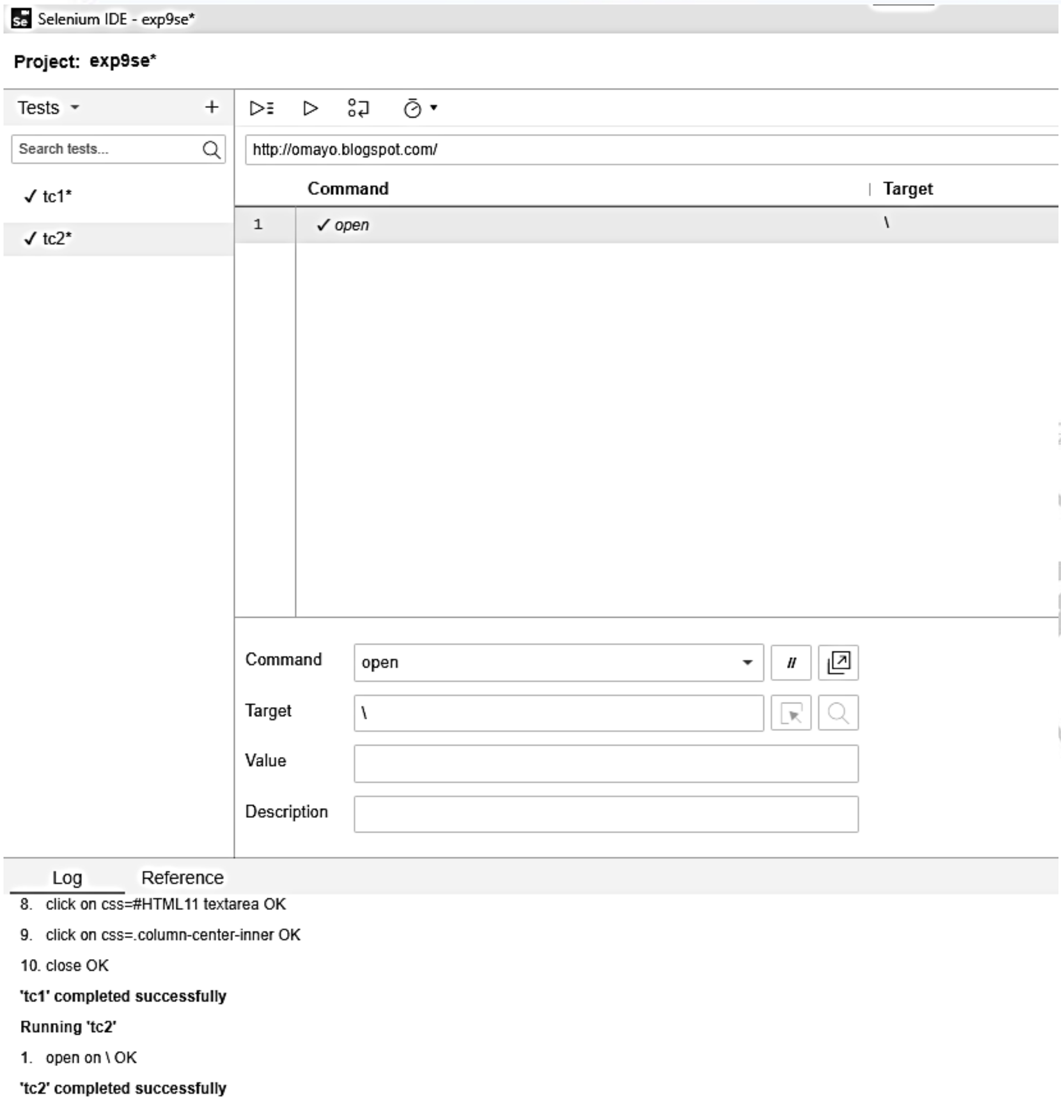


Project: exp9se*

http://omayo.blogspot.com/

	Command	Target	Value
1	✓ open	http://omayo.blogspot.com/	
2	✓ set window size	1616x876	
3	✓ click	id=ta1	
4	✓ type	id=ta1	\nhello ananya
5	✓ mouse down at	css=#HTML11 textarea	51.5,25.171875
6	✓ mouse move at	css=#HTML11 textarea	51.5,25.171875
7	✓ mouse up at	css=#HTML11 textarea	51.5,25.171875
8	✓ click	css=#HTML11 textarea	
9	✓ click	css=.column-center-inner	
10	✓ close		

```
5.  mouseDownAt on css=#HTML11 textarea with value 51.5,25.171875 OK
6.  mouseMoveAt on css=#HTML11 textarea with value 51.5,25.171875 OK
7.  mouseUpAt on css=#HTML11 textarea with value 51.5,25.171875 OK
8.  click on css=#HTML11 textarea OK
9.  click on css=.column-center-inner OK
10. close OK
'tc1' completed successfully
```



http://omayo.blogspot.com/

	Command	Target
1	✓ open	\

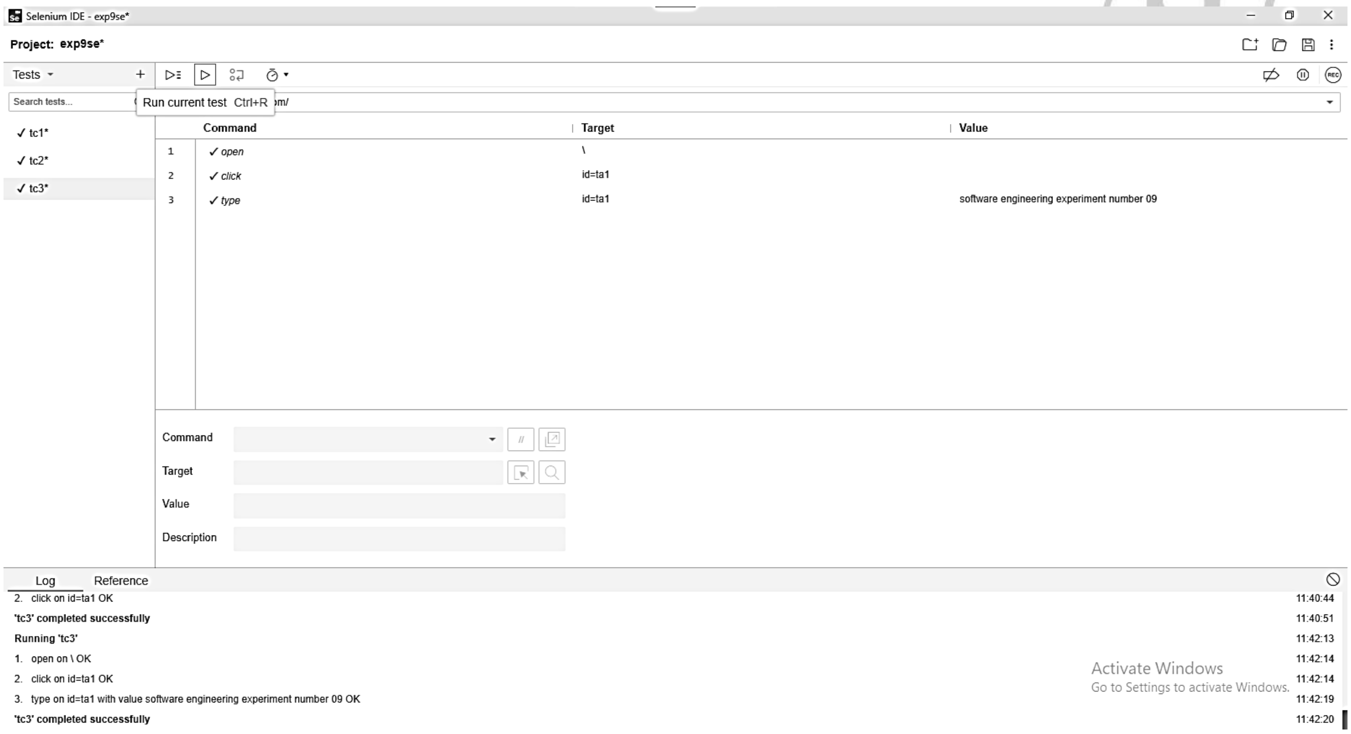
Command: open

Target: \

Value:

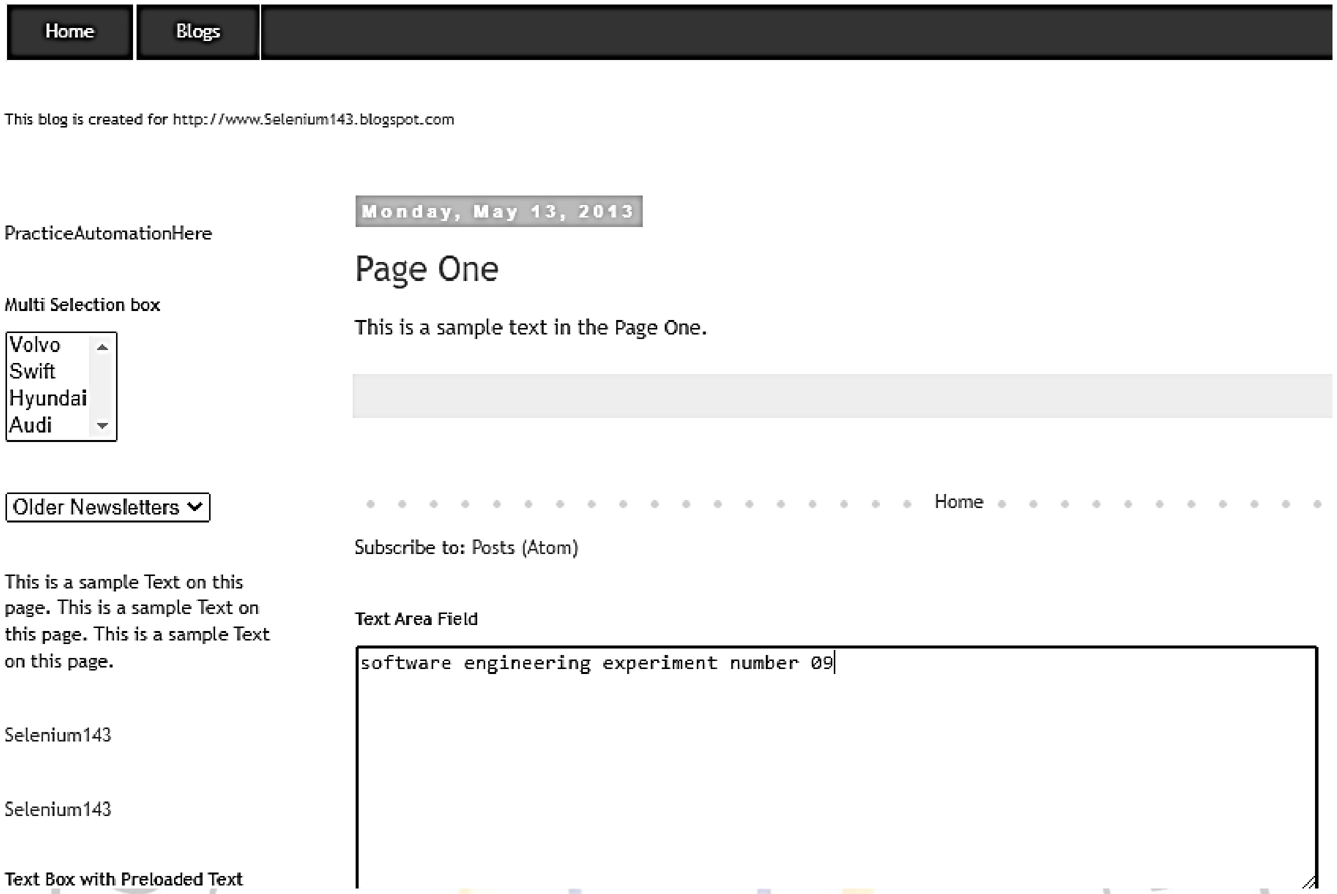
Description:

```
8.  click on css=#HTML11 textarea OK
9.  click on css=.column-center-inner OK
10. close OK
'tc1' completed successfully
Running 'tc2'
1.  open on \ OK
'tc2' completed successfully
```



	Command	Target	Value
1	✓ open	\	
2	✓ click	id=ta1	
3	✓ type	id=ta1	software engineering experiment number 09

```
2.  click on id=ta1 OK                                                      11:40:44
'tc3' completed successfully                                                11:40:51
Running 'tc3'                                                               11:42:13
1.  open on \ OK                                                            11:42:14
2.  click on id=ta1 OK                                                      11:42:14
3.  type on id=ta1 with value software engineering experiment number 09 OK  11:42:19
'tc3' completed successfully                                                11:42:20
```



This blog is created for http://www.Selenium143.blogspot.com

PracticeAutomationHere

Multi Selection box

Volvo
Swift
Hyundai
Audi

Older Newsletters

This is a sample Text on this page. This is a sample Text on this page. This is a sample Text on this page.

Selenium143

Selenium143

Text Box with Preloaded Text

Monday, May 13, 2013

Page One

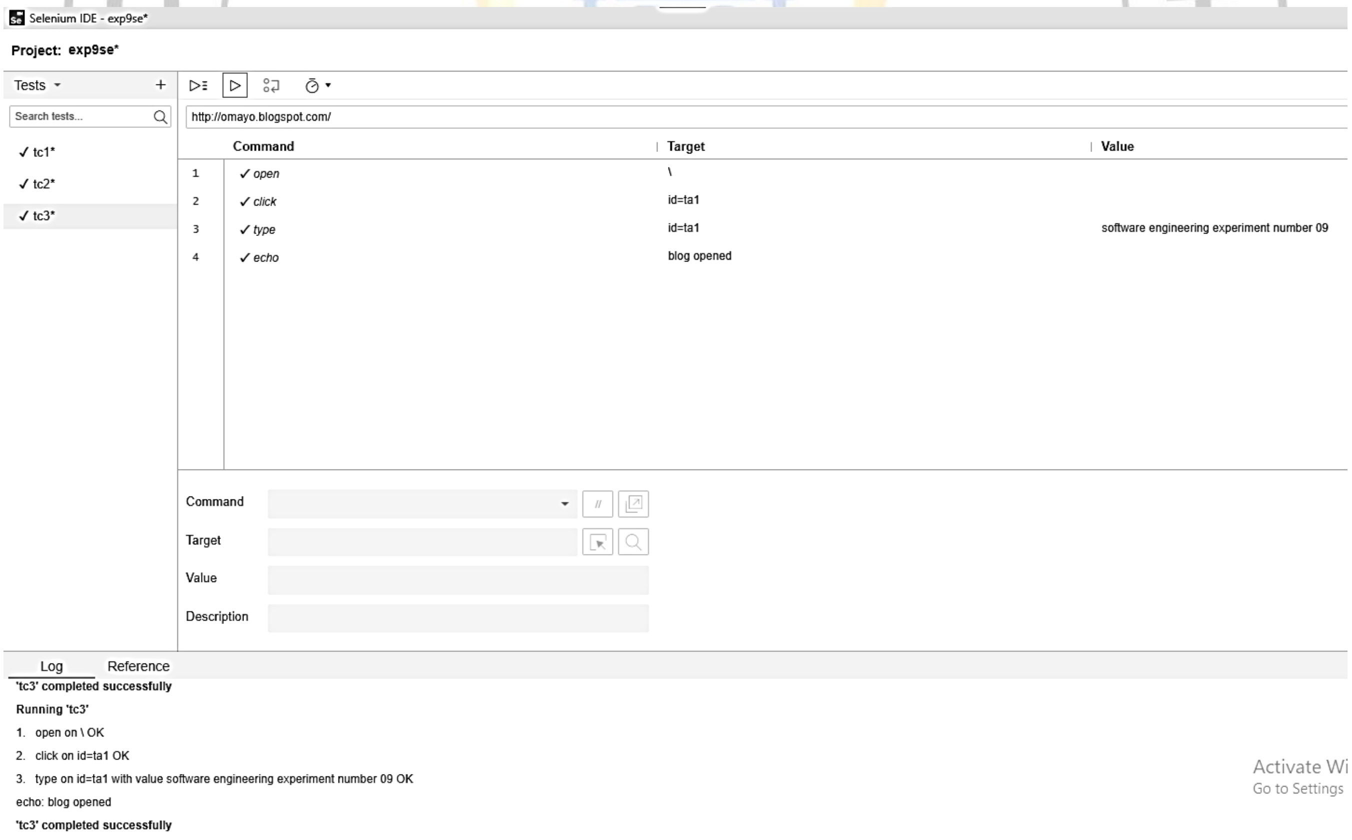
This is a sample text in the Page One.

Home

Subscribe to: Posts (Atom)

Text Area Field

software engineering experiment number 09



Project: exp9se*

✓ tc1*
✓ tc2*
✓ tc3*

http://omayo.blogspot.com/

	Command	Target	Value
1	✓ open	\	
2	✓ click	id=ta1	
3	✓ type	id=ta1	software engineering experiment number 09
4	✓ echo	blog opened	

Command
Target
Value
Description

```
'tc3' completed successfully
Running 'tc3'
1.  open on \ OK
2.  click on id=ta1 OK
3.  type on id=ta1 with value software engineering experiment number 09 OK
echo: blog opened
'tc3' completed successfully
```

Activate Wi
Go to Settings

Learning Outcomes: Students should have the ability to
LO1: Students will be able to understand Software Testing Concepts and the various Software standards.
LO2: to test a software with the help of Selenium
LO3: create test cases
LO4: To understand different tools for testing

Outcomes: Upon completion of the course students will be able to write test cases for the project.

Conclusion:

In this experiment, we have learned the concepts of software testing and created various test cases using Selenium tool.

For Faculty Use

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				