Lab-12: Study of Selenium

What are the advantages of automated testing?

- Increased test coverage
- Improved accuracy and reliability
- Time and cost savings
- Reusability of test cases
- Support for parallel execution

What is Selenium?

- Open-source suite of tools for automating web browsers
- Supports multiple programming languages and platforms

What are the benefits of using Selenium as an automation tool?

- Cross-browser compatibility
- Robust and flexible framework
- Large community support
- Integration with other testing tools

What are the limitations of Selenium as a tool?

- Limited support for desktop applications
- No built-in support for image and file uploads
- Requires additional tools for reporting and test management

How to install selenium? Describe the steps.

- Download the Selenium server standalone JAR file
- Set up the desired programming language bindings (e.g., Python, Java, C#)
- Configure system PATH variables
- Install required browser drivers

Which testing types are supported by Selenium?

Functional testing
- Regression testing
- Cross-browser testing
- Data-driven testing

What is Selenium WebDriver?

Core component of Selenium suite
- API for automating web browsers
- Supports multiple programming languages

Which browsers provide support for Selenium WebDriver?

Google Chrome
- Mozilla Firefox
- Internet Explorer
- Microsoft Edge
- Safari
- Opera

Which are the ways to locate an element of a webpage using Selenium?

- By ID
- By name
- By XPath
- By CSS selector
- By link text
- By partial link text

Demonstrate the use of selenium to test web page using 1. Selenium Chrome Extension 2. Python Code.

1. Selenium Chrome Extension:

- Install the Selenium IDE extension in Chrome
- Record and playback test cases
- Export tests as programming language scripts

2. Python Code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Firefox()
driver.get("https://www.selenium.dev/selenium/web/inputs.html")

# Click on the element
driver.find_element(By.NAME, "color_input").click()
```