**TASK 15**

1. **The difference between IDE, Grid, RC and Web Driver.**

**Selenium IDE:-**

Selenium IDE is a Firefox plugin which is used to create and execute test cases

It records and plays back the interactions which the user had with the web browser

Using IDE, you can export the programming code in different languages: Java, Ruby, Python and so on

**Selenium Grid:-**

Selenium Grid is used for parallel testing or distributed testing. It allows us to execute test scripts parallels on different machines

**Selenium RC:-**

Selenium Remote Control (RC) is used to write test cases in different Programming languages

In Selenium IDE, we can run the recorded scripts only in Firefox browser, whereas, in Selenium RC, we can run the recorded script in any browser like IE, Chrome, Safari, Opera and so on

**Selenium web Driver:-**

Selenium web Driver is a tool used to automate testing for web application

It allows us to execute tests against different browsers like Firefox, Chrome, IE & Safari

Selenium web Driver eliminated the use of Selenium Server thus making it work faster than RC

**3. What is Selenium? How it I useful in Automation Testing?**

Selenium is an open-source, automated testing tool used to test web applications across various browsers. Selenium can only test web applications, unfortunately, so desktop and mobile apps can’t be tested. However, other tools like Appium and HP’s QTP can be used to test software and mobile applications.

* Selenium is easy to use since it’s primarily developed in JavaScript
* Selenium can test web applications against various browsers like Firefox, Chrome, Opera, and Safari
* Tests can be coded in several programming languages like Java, Python, Perl, PHP, and Ruby
* Selenium is platform-independent, meaning it can deploy on Windows, Linux, and Macintosh
* Selenium can be integrated with tools like JUnit and TestNG for test management

Selenium Software is not just a single tool but a suite of software, each piece catering to different Selenium QA testing needs of an organization. Here is the list of tools

* Selenium Integrated Development Environment (IDE)
* Selenium Remote Control (RC)
* Web Driver
* Selenium Grid

Selenium is the most popularly used freeware and open-source automation tool. The benefits of Selenium Testing with respect to Test Automation are immense. Importantly, it enables record and playback for testing web applications and can run multiple scripts across various browsers. Selenium advantages for Test Automation hold relevance across diverse business segments.

**1. Open-Source**

As mentioned earlier, the biggest strength of Selenium is that it is freeware and a portable tool. It has no upfront direct costs involved. The tool can be freely downloaded and its community-based support is freely available.

**2. Language Support**

Selenium supports various languages, including Java, Perl, Python, C#, Ruby, Groovy, JavaScript, and more. It has its own script, but it is not limited by that language. It can work with various languages – whatever the developers/testers are comfortable with.

**3. Supports Operating Systems**

Selenium can operate and support multiple Operating Systems (OS) like Windows, Mac, Linux, and UNIX. With the Selenium suite of solutions, a tailored testing suite can be created over any platform and then executed on another one. For instance, you can easily create test cases using Windows OS and run them on a Linux–based system.

**4. Support across browsers**

Selenium testing offers the advantage of automating web application testing across various browsers: Internet Explorer, Chrome, Firefox, Opera, and Safari, ensuring consistent functionality and reducing manual effort. This becomes highly resourceful while simultaneously executing and testing tests across various browsers.

**5. Support for programming languages and framework**

Selenium integrates with programming languages and various frameworks. For instance, it can integrate with ANT or Maven type of framework for source code compilation. Further, it can integrate with the TestNG framework for testing applications and reporting purposes. It can integrate with Jenkins or Hudson for Continuous Integration (CI) and even integrate with open-source tools to support other features.

**6. Tests across devices**

Selenium Test Automation can be implemented for mobile web application automation on Android, iPhone, and Blackberry. This can help in generating necessary results and address issues continuously.

**7. Constant updates**

Selenium support is community–based, which enables constant updates and upgrades. These upgrades are readily available and do not require specific training. This makes Selenium resourceful and cost-effective as well.

**8. Loaded Selenium Suites**

Selenium is not just a singular tool or utility, it is a loaded package of various testing tools and so, is referred to as a Suite. Each tool is designed to cater to different testing needs and requirements of test environments.

Selenium can also support Selenium IDE, Selenium Grid, and Selenium Remote Control (RC).

**9. Ease of implementation**

Selenium offers a user-friendly interface that helps create and execute tests easily and effectively. Its open-source features help users to script their own extensions that make them easy to develop, customize actions, and even manipulate at an advanced level.

Tests run directly across browsers; users can watch while the tests are being executed. Additionally, Selenium’s reporting capabilities are one of the reasons for choosing it, as it allows testers to extract the results and take follow-up actions.

**10. Re usability and Add-ons**

Selenium Test Automation framework uses scripts that can be tested directly across multiple browsers. Concurrently, it is possible to execute multiple tests with Selenium, as it covers almost all aspects of functional testing by implementing add-on tools that broaden the scope of testing.

Another school of thought talks about some gaps in Selenium Automation Testing. For instance, Test Automation experts also say that Selenium is not an all-inclusive tool for automating the testing of web applications, as it needs third-party frameworks and language support to get absolutely functional and show the needed results.

The advantages of Selenium Testing are immense, encompassing automated cross-browser testing, enhanced testing efficiency, and improved software quality.

**4. What is the browser drivers used in selenium?**

The browser driver is the key to receiving the HTTP requests from the JSON Wire Protocol through its HTTP server and sends the processed requests to the real browser. This is where the interaction with elements takes place in any operation.

Selenium web Driver supports many browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and Opera. Each browser has unique features and capabilities. The different types of drivers available in Selenium web Driver include:

* Chrome Driver: For executing Selenium Automation Scripts on Chrome Browser
* Firefox Driver: For Firefox Browser
* InternetExplorerDriver: For Microsoft Internet Explorer
* Safari Driver: For Apple Safari
* Opera Driver: For Opera Browser
* Edge Driver: For Edge Browser
* Event firing web Driver: For Event Firing web Driver
* Remote web Driver: For Remote web Driver

**5. What are the steps to create a simple web driver script? Explain with code**

Basic Steps in a Selenium Web Driver Script

* Create a Web Driver instance.
* Navigate to a webpage.
* Locate a web element on the webpage via locators in selenium.
* Perform one or more user actions on the element.
* Preload the expected output/browser response to the action.
* Run test.

**Step 1**: Launch Eclipse IDE and open the Selenium Java project.

**Step 2**: Create a new package by right-clicking on the src folder in the Project Explorer view and selecting “New” > “Package.”

**Step 3**: Name the package and click “Finish.”

**Step 4**: Create a new Java class in the package by right-clicking on the package and selecting “New” > “Class.”

**Step 5**: Name the class and click “Finish.”

**Step 6**: Import the necessary Selenium and Java packages by adding the following import statements at the beginning of your class:

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

**Step 7**: Set the system property for the ChromeDriver executable by adding the following code before creating an instance of the WebDriver interface:

System.setProperty(“webdriver.chrome.driver”, “path/to/chromedriver.exe”);

Replace “path/to/chromedriver.exe” with the actual path to the ChromeDriver executable on your machine.

**Step 8**: Inside the main method of the class, create an instance of the WebDriver interface using the ChromeDriver class and navigate to a URL:

public static void main(String[] args) {

System.setProperty(“webdriver.chrome.driver”, “path/to/chromedriver.exe”);

WebDriver driver = new ChromeDriver();

driver.get(“https://www.google.com”);

}

**Step 9**: Save the file and run it by right-clicking on the class and selecting “Run As” > “Java Application.”