THEORETICAL QUESTIONS: - Rose Ramos

Q:What is Selenium ?
A:open-source software testing framework that is used to automate web browsers.

Q:What are the different components and versions of Selenium?
A:It provides a suite of tools and libraries for automating web browsers, including Firefox, Chrome, Safari, Internet Explorer, and others.

Q:What are Locators, different types of locators and their priorities in Selenium?
A:The different types of locators used in Selenium are:
ID- It is the most efficient and reliable way to locate web elements, as IDs should always be unique on a page.
Name-selects the web element by its name attribute.
Class Name- selects the web element by its class attribute. It is useful when the class attribute is unique or when multiple web elements share the same class attribute.
Tag Name-This locator selects the web element by its tag name, such as or <a>.
Link Text- selects the web element by its link text, which is the text displayed on a hyperlink. It is useful for links that have unique text.
Partial Link Text- selects the web element by its partial link text, which is a substring of the link text and useful when the link text is too long or when only a part of the link text is unique.
XPath- selects the web element using an XPath expression, which is a language for selecting nodes in an XML document. It useful for complex page structures or when other locators cannot select the desired web element.
CSS Selector- selects the web element using a CSS selector, which is a pattern for selecting elements based on their attributes and useful for complex page structures or when other locators cannot select the desired web element.

The priorities in Selenium when selecting a locator, it is essential to prioritize the ones that are most efficient and reliable.
The ID locator is the most efficient and reliable, followed by name and class name.
Link text and partial link text are useful for links
XPath and CSS selectors are useful for complex page structures.
Note- XPath and CSS selectors can be slow and cause performance issues. Therefore, it is important to use the most efficient locator that can select the desired web element accurately.

Q: The different types of drivers in Selenium WebDriver are:
A: WebDriver for Chrome
WebDriver for Firefox
WebDriver for Edge
WebDriver for Internet Explorer
WebDriver for Safari

WebDriver for Opera
WebDriver for Android
WebDriver for iOS

Q:How do I launch the browser using WebDriver?
A: To launch the browser using WebDriver, you can use the `get()` method
driver = new ChromeDriver();

Q: What are the different types of navigation commands in WebDriver?
A:The different types of navigation commands in WebDriver are:
navigate().to() // This method navigates to a specific URL.
navigate().back() // This method goes back to the previous page.
navigate().forward() // This method goes forward to the next page.
navigate().refresh() //This method refreshes the current page.

Q:How can you find whether an element is displayed on the screen using Selenium?
A: use the `isDisplayed()` method.
Ex: if (driver.findElementById("username").isDisplayed()) { // The element is displayed.

Q: How can you get a text on a web element using Selenium WebDriver:
A: getText() method.
Ex: String username = driver.findElementById("username").getText();

Q: How to type into a text box using Selenium?
A: sendKeys() method
Ex: driver.findElementById("username").sendKeys("John Doe");

Q: How to handle a drop-down field and select a value from it using Selenium?
A:Select class provided by the *org.openqa.selenium.support.ui* package.
The Select class provides methods to select an option from a drop-down field.
A: use select() method.
Ex: driver.findElementById("gender").selectByValue("Male");
Ex: WebDriver driver = new ChromeDriver();
driver.get("https://www.example.com");
Ex: WebElement dropdown = driver.findElement(By.id("myDropdown"));
Ex:// Create Select object and select a value
Select select = new Select(dropdown);
select.selectByVisibleText("Option 1");
Ex: Close the browser
driver.quit();

Q: The different types of waits available in WebDriver are:
A:

- Implicit wait- This is a default wait that is applied to all WebDriver operations. It is set to 0 seconds by default.
- Explicit wait- This is a wait that is explicitly specified by the user. It is used to wait for a specific condition to be met before continuing with the test.
- Fluent wait- This is a wait that combines the features of implicit and explicit waits. It is used to wait for a specific condition to be met, but it also takes into account the time it takes for the condition to be met.

Q:What is the latest Selenium tool?
A: The latest Selenium tool is Selenium 4. It was released in November 2021. It includes a number of new features, such as:

- Support for asynchronous testing
- Improved support for mobile testing
- New WebDriver APIs
- Performance improvements

Q:What do we mean by Selenium 1, Selenium 2 and Selenium 3?
A:Selenium 1, Selenium 2, and Selenium 3 are different versions of the Selenium testing framework.

Selenium 1 was the original version of the framework, and it was based on the Selenium RC (Remote Control) library.

Selenium 2 was a major update to the framework, and it introduced the WebDriver API. Selenium 3 is the latest version of the framework, and it includes a number of new features and improvements.

Q:When should I use Selenium Grid?

A:You should use Selenium Grid when you need to run your tests on multiple browsers or machines and allows you to distribute your tests across multiple machines improving performance of your tests.

Q:What is the difference between / and // in XPath?
A:The difference between / and // in XPath is:
/ selects the element at the root of the document
// selects all elements of the specified type in the document.

Q:What is an XPath?
A:An XPath is a language that is used to locate and find web elements in an XML document.

Q:What is the difference between driver.close() and
driver.quit() commands?
A: driver.close() closes the current window
driver.quit() closes all windows that the program has opened

Q:Is WebDriver a class or interface?
A:WebDriver is an interface

Q:What is the super interface of WebDriver?
A:The super interface of WebDriver is RemoteWebDriver.

Q:How to find more than one web element in to a list?
A: use the findElements() method. //returns a list of all elements that match the specified locator.

Q:Is FirefoxDriver a class or interface?
A:FirefoxDriver is a class. It is the class that implements the FirefoxDriver interface.

Q:Explain the line of code WebDriver driver = new FirefoxDriver();?
A: WebDriver driver = new FirefoxDriver(); //creates a new instance of the FirefoxDriver class and assigns it to the variable driver.

Q:How to handle frames in WebDriver
A:use the switchTo().frame() method //takes the ID or name of the frame as a parameter and switches the focus to that frame.

Q:How to click on a hyper-link using Selenium WebDriver?
A:use the click() method //takes the web element as a parameter and clicks on that element.

Q:What are the programming languages supported by Selenium WebDriver?
A: Selenium WebDriver supports a variety of programming languages, including:

- Java
- Python
- C#
- Ruby
- JavaScript
- PHP
- Perl

Q: What are the operating systems supported by Selenium WebDriver?
A: Selenium WebDriver supports a variety of operating systems, including:

- Windows
- Mac OS X
- Linux
- Solaris
- AIX

Q:What are the browsers supported by Selenium WebDriver?
A:Selenium WebDriver supports a variety of browsers, including:

- Chrome
- Firefox
- Internet Explorer
- Edge
- Safari
- Opera

Q:What is the difference between Implicit Wait and Explicit Wait?
A:
implicit wait- is a global wait that is applied to all WebDriver operations.
explicit wait- is a specific wait that is applied to a particular WebDriver operation.

Q:How to read and verify the text on the tool tip using Selenium WebDriver?
A: getText() method takes the web element as a parameter and returns the text of the tool tip for that element.

Q:Can Selenium Automate Desktop Applications?
A:Yes, Selenium can automate desktop applications using the Selenium IDE which is a Firefox add-on. It allows you to record and playback user actions on a desktop application.

Q:What is the main component of Selenium?
A:The main component of Selenium is the WebDriver API- a set of classes and methods that allow you to interact with web browsers programmatically.

Q:What is an XPath?
A:Xpath means XML Path Language and is a query language used to navigate and select elements from an XML or HTML document.
Q:The difference between Absolute and Relative XPath with examples?
A:
Absolute XPath expressions start with the root node of the HTML document and are less flexible than Relative XPath.
Ex: By.xpath("/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input")

Relative XPath expressions start with the current node (usually the parent node of the element you want to locate) and are more flexible and recommended to use in Selenium test automation.
Ex: By.xpath("//input[@name='q']")

Q:What is the disadvantage of Absolute XPath & why is Relative XPath recommended?
A: Absolute XPath expressions start with the root node of the HTML document and are less flexible than Relative XPath.

Q:What is an Absolute XPath? Write its syntax?
A: Absolute XPath is a direct way of finding syntax using Family Tree: ancestor, parent, preceding-sibling, following- sibling, child, descendent.
Ex:driver.findElement(By.xpath("//li[@class= 'user-li']/child::a"))

Q:What is a Relative XPath? Write its syntax?
A: Syntax of Relative Xpath :
Syntax1
//html[@attribute = ‘value oftheattribute’]

Syntax2
//html[@attibute1 = ‘value’ and @attribute2 = ‘value’]

SYNTAX 3
//html[@attibute1 = ‘value’ or @attribute2 = ‘value’]

SYNTAX 4
//html[text() = ‘valueofthetext’] // this kind of xpath only works for link

Q: How to execute JavaScript in Selenium?
A: execute_script method
Ex: driver.execute_script('alert("Hello, World!");

Q:What is the concept that makes XPath Expressions powerful ?
A:-the combination of
flexible location paths to specify the elements you want to select,
expressions that allow you to traverse the document tree in a flexible and efficient way
powerful functions and operators, predicates to further refine your selection criteria and advanced selection techniques makes XPath expressions a very powerful tool for navigating and selecting elements within XML and HTML documents.

Q:Why CSS Selectors have higher priority over XPath Expressions?
A: CSS Selectors have higher priority over XPath Expressions because of the way that the browser rendering engine processes web page styles and layouts.

Q:Names of add-ons which can auto generate the XPath Expressions and CSS Selectors?
A:

- SelectorGadget- Google Chrome and Firefox extensions
  -allows you to visually select any element on a web page and generates the corresponding CSS selector or XPath expression.
- Firebug- popular extension for Firefox
  -provides a range of web development tools:HTML, CSS editor, JavaScript debugger, network monitor.
  - feature for generating XPath expressions and CSS selectors.
- XPath Helper- Chrome extension
  - provides a simple interface for generating XPath expressions.
  - "Select" button that allows you to click on any element in the web page and generate the corresponding XPath expression.
- ChroPath- Chrome and Firefox
  - easy-to-use interface for generating and editing XPath expressions and CSS selectors
  - features for generating relative XPath expressions.

Q:Java program for printing the even numbers between 1 and 100 using for loop?
A:

```
public class EvenNumbers {
    public static void main(String[] args) {
        for (int i = 1; i <= 100; i++) {
            if (i % 2 == 0) {
                System.out.println(i);
            }
        }
    }
}
```

Q:Write a Java program to find the sum of the first 100 numbers using a for loop?
A:

```
public class SumOfNumbers {
    public static void main(String[] args) {
        int sum = 0;
        for (int i = 1; i <= 100; i++) {
            sum += i;
        }
        System.out.println("The sum of the first 100 numbers is: " + sum);
    }
}
```

Q:Prints numbers from 1 to 100. Print number and
Divisible by 5 text if divisible?
A:

```
public class DivisibleByFive {
    public static void main(String[] args) {
        for (int i = 1; i <= 100; i++) {
            System.out.print(i);
            if (i % 5 == 0) {
                System.out.print(" Divisible by 5");
            }
            System.out.println();
        }
    }
}
```

Q:Does Java support multiple inheritance? Give reasons.
A:No- can only inherit from one direct super class.
-does not support multiple inheritance through classes.
-a class can only inherit from one direct superclass and implement multiple interfaces, which allows it to inherit the method signatures and constants defined in each interface. This provides

a way for a Java class to inherit behavior from multiple sources, similar to multiple inheritance in other programming languages.
-Java does support multiple inheritance of types through interfaces.
-An interface in Java is similar to a class in that it can define methods and constants, but it cannot provide an implementation of those methods
Ex:

```
public class MyClass implements Interface1, Interface2 {
    // implementation of methods from Interface1 and Interface2
}
```

Q:What is the parent or base class of all the classes in Java?
A:Object class
Ex: equals(), hashCode(), and toString().

Q:What is the difference between instance variable and local variable?
A:Differences of instances variables and local variables:
Instance variables:
- is declared inside a class but outside any method, constructor, or block.
-are associated with instances of the class, meaning that each instance of the class has its own copy of the variable
-are created when an object of the class is created
-are destroyed when the object is garbage collected.
- can be accessed and modified by any method or constructor in the class.

local variable:
- is declared inside a method, constructor, or block.
- only accessible within the method, constructor, or block where they are declared.
- are created when the method, constructor, or block is entered.
- are destroyed when it is exited. Local variables must be initialized before they can be used.

Q:Is Java a pure 100% Object Oriented Programming language?
A: Java is not a pure OOP language as it allows for primitive data types such as int, float, and boolean.

]

Q:What is the difference between Primitive & Non- Primitive Data types in Java?
A:Data types are categorized as primitive and non-primitive- reference types or objects.

Primitive data types:

pre-defined data types that are built into the Java language
Ex: int, char, boolean, float, double
-are stored directly in memory
-have a default value assigned to them
-are operated using operators
-are assigned by value, meaning that a copy of the data is made and assigned to the variable.
-have fixed size and they do not change

Non-primitive data types:

- are created by the programmer using classes or interfaces.
-stored as references to memory locations where the actual data is stored
- do not have any default value assigned to them.
Ex: the default value of an int is 0, and the default value of a boolean is false.

- are operated using methods.
- are assigned by reference, meaning that a reference to the memory location where the data is stored is assigned to the variable.
- can grow or shrink dynamically based on the data.

Q:Why Strings are immutable in Java?
A: ensures security, efficiency, thread safety, and allows for caching, making Java more robust and performant.

Q:What is the difference between String and StringBuffer?
A:Both String and StringBuffer are used to represent a sequence of characters in Java main difference is:
String:
- are immutable
- once a String object is created, it cannot be modified.
StringBuffer:
- objects are mutable
- the contents of the object can be modified after it is created.

—--------

Q:Accessing all elements inside int[][] a = {{5,2,9},{4,6,8}};
using for loop?
A:

```
int[][] a = {{5, 2, 9}, {4, 6, 8}};

for (int i = 0; i < a.length; i++) {
    for (int j = 0; j < a[i].length; j++) {
        System.out.print(a[i][j] + " ");
    }
}
```

Q: Assign different values say integer, character, string etc
into a single array?
A:

```
Object[] arr = new Object[5];
arr[0] = 42;
arr[1] = 'A';
arr[2] = "hello";
arr[3] = 3.14;
arr[4] = true;

for (int i = 0; i < arr.length; i++) {
    System.out.println(arr[i]);
}
```

Q:What is the disadvantage of array?
A:
-Once an array is created, its size is fixed and cannot be changed, very limiting in situations where you need to add or remove elements dynamically.
- Arrays do not have built-in methods for common operations such as sorting, searching, or filtering.
-You need to write your own algorithms or use external libraries to perform these operations.
-homogeneous which means that all elements in the array must be of the same type.
-lacks flexibility
- Inefficient insertions and deletions: Because the size of an array is fixed, inserting or deleting elements can be inefficient, especially if the operation needs to be performed frequently. This is because all elements after the insertion or deletion point need to be shifted, which is time-consuming.

Q:What is the difference between equals() and == Operator?

A: Difference between equals() and == Operator are:

== operator:

- compares the references of two objects and returns true if they refer to the same object in memory
- if the two objects are physically the same in terms of their memory location.

equals() method:

- is used to compare the contents of two objects and returns true if the contents are equal.
- contents of an object refer to its state, which is defined by the values of its fields.
- is usually overridden in Java classes to compare objects based on their state rather than their memory location.

```
String s1 = "hello";
String s2 = "hello";
String s3 = new String("hello");

System.out.println(s1 == s2);  // true, because s1 and s2 reference the same object in memory
System.out.println(s1 == s3);  // false, because s1 and s3 are two different objects in memory
System.out.println(s1.equals(s2));  // true, because the contents of s1 and s2 are the same
System.out.println(s1.equals(s3));  // true, because the contents of s1 and s3 are the same
```

Q:What is the purpose of using Wrapper classes in Java?

A:Wrapper classes are used to represent primitive data types as objects
and provide additional functionality that is not available with primitive data types.

Ex:

```
int i = 42;
Integer integerObj = Integer.valueOf(i); // using wrapper class to represent int as an object
int j = integerObj.intValue(); // extracting the int value from the wrapper object

System.out.println("i = " + i);
System.out.println("integerObj = " + integerObj);
System.out.println("j = " + j);
```

Q:How to capture screen-shot in Selenium WebDriver?
A:Great for Capturing a screenshot of a web page and useful for debugging or error reporting.

// Create a WebDriver instance
Ex: WebDriver driver = new ChromeDriver();

// Navigate to a web page
Ex: driver.get("http://www.example.com");

// Take a screenshot and store it in a file
Ex: File screenshotFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

// Copy the file to a location of your choice
Ex: FileUtils.copyFile(screenshotFile, new File("path/to/screenshot.png"));

Q:What is Automation Testing?
A:Automation testing:
- is a software testing technique that involves using software tools to automate the execution of test cases, comparing actual results against expected results, and generating detailed test reports.
-test various types of software applications like desktop applications, web applications, mobile applications, and APIs.
-aspects of the software, including functional testing, regression testing, performance testing, security testing,etc.

Q:What is the difference between Manual and Automation Testing?
A: Difference between Manual and Automation Testing:

Manual testing- is the process of manually executing test cases without using any software tools or scripts.

Automation testing-use software tools and scripts to automate the execution of test cases.

Q:What are primitive data types in Java?
A: There are eight primitive data types in Java and are built into the Java language and do not require any additional libraries or frameworks to use:

- byte: used to store whole numbers from -128 to 127
- short: used to store whole numbers from -32,768 to 32,767
- int: used to store whole numbers from -2,147,483,648 to 2,147,483,647
- long: used to store whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
- float: used to store floating-point numbers with single precision
- double: used to store floating-point numbers with double precision
- boolean: used to store true or false values
- char: used to store a single character or letter

Q:Can we create an object for an interface?
A: We cannot create an object for an interface since an interface is a blueprint for a class and does not provide any implementation for its methods.

Q:Can we create an object in an abstract class?
A:No, we cannot create an object of an abstract class in Java because abstract classes are incomplete and are meant to be extended by other classes to provide their implementation.

Q:Can we write Webdriver driver = new Webdriver()
A:No because Webdriver is an abstract class and abstract classes cannot be instantiated.

Q:What is the purpose of using constructors in Java?
A:The purpose of using constructors in Java is to initialize the state of an object. When an object is created, the constructor is called to initialize the object's state.

Q:What is the purpose of using constructors in Java?
A: Constructors have the same name as the class and are called when an object is created. They don't have a return type, not even void. Methods can have any name, can have a return type, and are called explicitly by the code.

Q: What is the purpose of using the 'this' keyword in Java?
A: The purpose of using the 'this' keyword in Java is to refer to the current object. It is often used to disambiguate between instance variables and local variables with the same name.

Q:What is Overloading in Java?
A: Overloading in Java is the ability to have multiple methods with the same name but different parameters in a class. It allows a method to perform different operations based on the parameters passed.

Q:What is the purpose of using packages in Java?
A:The purpose of using packages in Java is to organize classes and interfaces into a hierarchy, making it easier to manage and reuse code. It also helps to avoid naming conflicts.

Q:Keyword used by a Java class to inherit the properties of another class?
A: extends

Q:How to access the variables and methods of another class in Java
A: Create an object of that class and use dot notation to access its members.

Q:What is Overriding in Java?
A:Overriding is the ability of a subclass to provide its own implementation of a method that is already provided by its parent class.

Q:Is Overriding applicable for constructors in Java?
A:No, overriding is not applicable for constructors in Java because constructors cannot be inherited. However, a subclass can call its parent's constructor using the `super()` keyword to initialize its own state.

Q:What are the different modifiers in Java?
A:In Java, there are four access modifiers:
Public
Private
Protected
default

Q:What is the difference between default and protected
access modifiers in Java?
A: Both modifiers "default" and "protected" are used to control the visibility and accessibility of class members (fields, methods, and inner classes) from other classes:

Default Access Modifier- If a class member is not specified with any access modifier, then it is considered as default access modifier. A class member declared with the default modifier is accessible within the same package but not accessible outside the package.

Protected Access Modifier: A class member declared with the protected access modifier is accessible within the same package and from subclasses, even if they are outside the package. To summarize, default access modifier restricts the access to class members outside the package, while protected access modifier allows access to subclass even if it is outside the package.

Q:How to select a radio button in Selenium WebDriver?
A: use click() method ex: radioBtn.click();

Q:Why do you get NoSuchElementException?
A:NoSuchElementException is thrown when the WebDriver is not able to locate a web element using the specified locator strategy. This can happens when there is an Incorrect Locator Strategy, Timing Issues,or during Incorrect Frame/Window Switching.

Q:How can we fetch the page source in Selenium?
A: getPageSource() method provided by the WebDriver interface

Q:How can we fetch the title of the page in Selenium?
A: getTitle() method

Q:What is the difference between static and instance variables in Java?
A:Difference between static and instance variable in Java:
Static variables:
- are declared using the "static" keyword and are shared among all instances of a class.
- They are stored in the class memory area and can be accessed without creating an object of the class.

Instance variables:
- are declared inside a class but outside of any method, and each object of the class has its copy of the variable.
- are stored in the object memory area and are accessed using object references.

Q:What is the difference between static and non-static methods in Java?
A: The difference between static and non-static methods in Java:
Static methods:
-are also declared using the "static" keyword and can be called without creating an object of the class.
-are associated with the class rather than any particular object.

Non-static methods:
- also known as instance methods
-are called using an object reference and operate on the state of an object.

Q:What happens when specifying a “ final” modifier with variables & methods in Java?
A: When specifying a "final" keyword with variables and methods Final variables cannot be reassigned, and final methods cannot be overridden by subclasses.
-Final classes cannot be extended.
- are used to declare variables, methods, or classes that cannot be changed or overridden.

Q:What is the difference between abstract classes and interfaces in Java?
A: Differences between abstract and interfaces in Java are:
Abstract classes:
- that cannot be instantiated and are meant to be subclassed.
- they may contain both concrete and abstract methods, and subclasses must implement the abstract methods.
-can have constructors

Interfaces:
-are a collection of abstract methods and constants that define a contract for classes to implement. Classes can implement multiple interfaces but can only inherit from one abstract class.
- cannot have constructors
—----
Q:What is the keyword used for inheriting the interfaces in Java?
A: "implements"

Q:Can Captcha be automated using Selenium?
A:No

Q:How to clear the text inside the text box fields using Selenium WebDriver?
A: clear() method
Ex:
WebElement textBox = driver.findElement(By.id("textbox"));
textBox.clear();

Q:What is the default timeout of Selenium WebDriver?
A: 30 seconds

Q:How can we submit a form in Selenium?
A: submit() method
Ex:
```
WebElement form = driver.findElement(By.tagName("form"));
form.submit();
```

Q:What is the purpose of getOptions() method?
A: getOptions() method in Selenium WebDriver is used to retrieve all the options available in a select element. This method returns a list of WebElements, which represent the individual options.

Q:Write an XPath to find all the hyper-links on a web page?
A: //a

Q:What is Select Class in Selenium WebDriver and how to use it?
A:Select class in Selenium WebDriver is used to work with dropdown menus, which are implemented using the select element in HTML.
- can be used to select and deselect options from a dropdown menu.

Ex:
```
WebElement dropdown = driver.findElement(By.id("dropdown"));
Select select = new Select(dropdown);
select.selectByVisibleText("Option 1");
```

Q:How to handle alerts in Selenium WebDriver?
A:
```
Alert alert = driver.switchTo().alert();
alert.accept();
```

Q:What is click() command in Selenium WebDriver?
A: click() - use to click on a web element such as a button, link, or checkbox.

```
WebElement button = driver.findElement(By.id("submit-button"));
button.click();
```

Q:What is sendKeys() command in Selenium WebDriver?
A:sendKeys()- simulate typing text into a text field or other input element

```
WebElement textBox = driver.findElement(By.id("text-box"));
textBox.sendKeys("Hello World");
```

Q:How to delete cookies in Selenium?
A:driver.manage().deleteAllCookies();

Q:How to get the href of a link?
A:
```
WebElement link = driver.findElement(By.tagName("a"));
String href = link.getAttribute("href");
```

Q:How do you click on a menu item in a drop down menu?
A:
```
WebElement dropdown = driver.findElement(By.id("dropdown"));
Select select = new Select(dropdown);
select.selectByVisibleText("Option 1");
```

Q:How to get typed text from a textbox?
A:
```
WebElement textBox = driver.findElement(By.id("text-box"));
String text = textBox.getAttribute("value");
```

Q:How to type text in a new line inside a text area?
A:
```
WebElement textArea = driver.findElement(By.id("text-area"));
textArea.sendKeys("Line 1" + Keys.ENTER + "Line 2");
```

Q:How to resize browser window using Selenium WebDriver?
A:
```
driver.manage().window().setSize(new Dimension(800, 600));
```

Q:How to scroll web page up and down using Selenium WebDriver?
A:
```
Actions actions = new Actions(driver);
actions.keyDown(Keys.CONTROL).sendKeys(Keys.END).perform();
```

Q:How to perform right click (Context Click) action in Selenium WebDriver?
A:
```
WebElement element = driver.findElement(By.id("menu-item"));
Actions actions = new Actions(driver);
actions.contextClick(element).perform();
```

Q:How to perform double click action in Selenium WebDriver?
A:
```
WebElement element = driver.findElement(By.id("double-click-item"));
Actions actions = new Actions(driver);
actions.doubleClick(element).perform();
```

Q:How to perform drag and drop action in Selenium WebDriver?
A: dragAndDrop() method needs two Actions Class arguments which are source and target elements:

```
WebElement sourceElement = driver.findElement(By.id("source")); //which will be dragged
WebElement targetElement = driver.findElement(By.id("target")); //which will be dropped

Actions actions = new Actions(driver);
actions.dragAndDrop(sourceElement, targetElement).perform();

// perform() method to execute the drag and drop action.
```