### Q1. What is Selenium WebDriver?

Selenium WebDriver is a popular tool for automating web browsers and has a program interface.

### Q2. How can we type text in a textbox using Selenium?

To type text in a textbox using Selenium, first, we need to locate the textbox using any of the locating strategies in Selenium, and then use the sendKeys() method to enter text into the textbox.

### Q3. How to click on a hyperlink in Selenium?

driver.findElement(By.linkText(“Today’s deals”)).click();

The command finds the element using link text and then clicks on that element, where after the user would be redirected to the corresponding page.

driver.findElement(By.partialLinkText(“Service”)).click();

The above command finds the element based on the substring of the link provided in the parenthesis and thus partialLinkText() finds the web element.

### Q4. How to mouse hover over a web element?

Actions class utility is used to hover over a web element in Selenium WebDriver

Instantiate Actions class.

    Actions action = new Actions(driver);

In this scenario, we hover over search box of a website

  actions.moveToElement(driver.findElement(By.id("id of the searchbox"))).perform();

### Q5. Can Captcha be automated?

No, Selenium cannot automate Captcha. Well, the whole concept of Captcha is to ensure that bots and automated programs don’t access sensitive information - which is why, Selenium cannot automate it. The automation test engineer has to manually type the captcha while other fields can be filled automatically.

### Q6. What is the major difference between driver.close() and driver.quit()?

driver.close()

This command closes the browser’s current window. If multiple windows are open, the current window of focus will be closed.

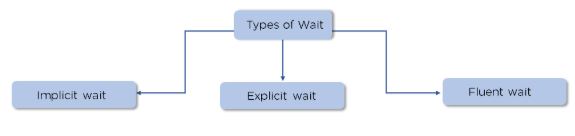
driver.quit()

 When quit() is called on the driver instance and there are one or more browser windows open, it closes all the open browser windows.

### Q7. Mention the advantages of using Selenium as an automation tool.

Selenium is an automation tool and has some unique benefits that give it a competitive edge over others such as open source, multi-language support, platform support, multi-browser support, framework availability and flexibility, reusability, and integrated and parallel test execution.

### Q8. What are the types of waits supported by WebDriver?



Implicit wait - Implicit wait commands Selenium to wait for a certain amount of time before throwing a “No such element” exception.

driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);

Explicit wait - Explicit wait is used to tell the Web Driver to wait for certain conditions before throwing an "ElementNotVisibleException" exception.

WebDriverWait wait = new WebDriverWait(WebDriver Reference, TimeOut);

Fluent wait - It is used to tell the web driver to wait for a condition, as well as the frequency with which we want to check the condition before throwing an "ElementNotVisibleException" exception.

Wait wait = new FluentWait(WebDriver reference).withTimeout(timeout, SECONDS).pollingEvery(timeout, SECONDS).ignoring(Exception.class);

### Q9. Mention the advantages of using Selenium as an automation tool.

Selenium is an automation tool and has some unique benefits that give it a competitive edge over others such as open source, multi-language support, platform support, multi-browser support, framework availability and flexibility, reusability, and integrated and parallel test execution.

**Q10. Write a program to launch chrome browser & navigate & set value into text box**