## Can you explain how you would handle dynamic web elements in Selenium? What strategies do you use to ensure your scripts are robust?

**:**

**Using Explicit Waits**: We can use WebDriverWait with ExpectedConditions like elementToBeClickable() or presenceOfElementLocated().

**Using Dynamic XPath or CSS Selectors**: If element attributes keep changing, we can use contains(), starts-with(), or following-sibling in XPath.

**Using JavaScript Executor**: If an element is hidden due to dynamic loading, JavaScript Executor (executeScript()) can interact with it.

**Handling AJAX Calls**: Using waits to ensure the element is fully loaded before interacting with it.

### Cross Question 1.2: What is stale element exception? How would you handle stale element exceptions caused by dynamic elements?

**:** StaleElementReferenceException occurs when an element is no longer attached to the DOM. To handle this:

1. **Re-locate the element** before interacting again.
2. **Use explicit waits** to wait until the element is stable.
3. **Try catching the exception** and reattempting the action.

## Question 2: How do you handle file uploads using Selenium?

driver.findElement(By.id("upload")).sendKeys("C:\\path\\to\\file.jpg");

### Cross Question 2.2: What would you do if the file upload control is inside an iframe?

First, we must switch to the iframe using:

Then, we can interact with the file upload element as usual. After performing the upload, we switch back to the default content using:

### How do you prioritize test cases in TestNG? Can you explain the use of dependsOnMethods and priority?

### What happens if a test method fails on which another method depends?

## How would you handle a scenario where elements are present in the DOM but are not interactable due to overlapping elements?

Ex: You try to click a button, but Selenium throws an ElementClickInterceptedException. How would you handle this situation?

-**Wait for the obstructing element to disappear**:

**Use JavaScript Executor to force click**:

**Scroll to the element before clicking**:

### Cross Question 11.1: Why does click() sometimes work on Chrome but fail on Firefox?

Chrome allows partial visibility clicks, while Firefox requires **full visibility**. That’s why scrollIntoView() may be needed in Firefox.

## How do you handle parallel execution of Selenium tests and avoid session conflicts?