Interview Q&A; Guide (Java, Selenium, API, Manual, CI/CD)

Q: Introduce yourself briefly.

A: I have X years of experience in Software Testing (Manual + Automation) with expertise in Selenium WebDriver, Java, TestNG, Cucumber BDD, API Testing (Postman, RestAssured), and CI/CD integration with Jenkins.

Q: Three OOP concepts you used in your framework?

A: Encapsulation (keeping locators + actions inside Page Classes), Inheritance (BaseClass used by all tests), Polymorphism (method overloading for utilities, method overriding for setup/teardown in hooks).

Q: Difference between ArrayList and LinkedList?

A: ArrayList is better for retrieving data (index-based), while LinkedList is better for insert/delete operations.

Q: POST vs PUT in API testing?

A: POST is used to create a new resource, while PUT is used to update/replace an existing resource.

Q: Java program: Count 'A's in a string

A: for(char c : str.toCharArray()) { if(c=='A' || c=='a') count++; }

Q: Stale Element Exception in Selenium?

A: Occurs when the DOM changes after locating the element. Handled by re-locating element or using WebDriverWait.

Q: 401 vs 500 error codes?

A: 401: Unauthorized (invalid credentials). 500: Internal Server Error (issue in server-side processing).

Q: Element Not Found Exception?

A: Occurs when locator strategy cannot find element. Handled by explicit waits or correcting locator.

Q: Factorial Program (Java)?

A: int fact=1; for(int i=1;i<=n;i++){ fact*=i; }

Q: Reverse a Number?

A: while(num!=0){ rev=rev*10+num%10; num/=10; }

Q: List vs Set in Java?

A: List allows duplicates & maintains insertion order. Set does not allow duplicates & order is not guaranteed.

Q: Program: Count duplicate characters in string?

A: Use HashMap to count occurrences >1.

Q: CI/CD tools experience?

A: I have used Jenkins for automated execution, Git for version control, and integrated Maven for dependency management.

Q: Domain experience?

A: Worked in Finance (Payments), Telecom, and Data Analytics projects.

Q: Team leadership experience?

A: Led 2–3 junior testers in automation, code reviews, and task allocation.

Q: Explain Polymorphism with examples?

A: Overloading: same method name with different params. Overriding: same method in parent & child with different behavior (used in Page classes & Hooks).

Q: Automation Framework experience?

A: Implemented Hybrid + BDD framework using Selenium, Java, TestNG, and Cucumber. Used POM for reusability, integrated Extent Reports, handled data with Excel/JSON.

Q: How do you handle dynamic elements?

A: Use dynamic XPath, CSS, explicit waits, or JavaScriptExecutor.

Q: How do you integrate automation in CI/CD?

A: Automation suite triggered via Jenkins pipeline on every Git commit. Reports published automatically after execution.

Q: How do you manage test data?

A: Stored in Excel/JSON/properties files. Utility classes fetch test data dynamically based on scenario.

Q: Severity vs Priority?

A: Severity = impact (High: system crash). Priority = order of fixing (High: login not working).

Q: Smoke vs Sanity testing?

A: Smoke: broad but shallow checks (basic app health). Sanity: narrow but deep checks (specific functionality works).

Q: BDD vs TDD?

A: BDD: write tests in Gherkin (user-focused). TDD: write unit test first, then code to pass it.

Q: Background vs Scenario Outline in Cucumber?

A: Background: common steps before every scenario. Scenario Outline: used for data-driven testing with Examples table.

Q: How do you handle waits in Selenium?

A: Implicit Wait: global delay. Explicit Wait: wait for specific condition. Fluent Wait: polling with timeout.

Q: Handling Alerts/Popups in Selenium?

A: Use driver.switchTo().alert() → accept(), dismiss(), getText().

Q: Reports in framework?

A: Used Extent Reports & Allure integrated with TestNG/Cucumber for HTML reports with screenshots on failure.

Q: Difference between Smoke & Regression testing?

A: Smoke: basic validation after build. Regression: verify existing features after new changes.

Q: Git experience?

A: Used GitHub/GitLab for branching, merging, pull requests, and version control of automation framework.