**1. Logging and Debugging Utilities**

* **Log4j / Logback / SLF4J**: For generating logs to monitor test execution.
* **Assertions**: Built-in assertions (JUnit, TestNG) or libraries like AssertJ, Hamcrest.
* **Debugger**: Using IDE debugging tools like IntelliJ or Eclipse to troubleshoot automation code.
* **Allure Reports**: For detailed reporting, logs, and attachments.

**2. Reporting Tools**

* **Extent Reports**: For generating detailed HTML reports.
* **Allure Reports**: For generating visually appealing and insightful reports.
* **Custom Reporting**: Using libraries like Apache POI or custom HTML templates.

**3. File Handling Utilities**

* **Apache POI**: For handling Excel files (XLS/XLSX) for data-driven testing.
* **OpenCSV / Jackson / Gson**: For handling CSV and JSON files for configuration or validation.
* **Java NIO / Files API**: For working with files and directories.

**4. API Utilities**

* **RestAssured**: For API testing and validation.
* **Postman and Newman**: For running and automating API tests.
* **JSONPath / XPath**: For parsing and validating JSON/XML responses.

**5. Web Automation Utilities**

* **WebDriverManager**: For managing browser drivers automatically.
* **Waits (Explicit/Implicit/Fluent)**: Handling dynamic elements and synchronization.
* **JavaScriptExecutor**: For handling complex web actions (e.g., scrolling, clicking hidden elements).
* **Action Class**: For advanced interactions like drag-and-drop, hover, etc.

**6. Test Management and CI/CD Tools**

* **Jenkins**: For Continuous Integration and triggering automated test builds.
* **Test Management Tools**: Integration with tools like JIRA, TestRail, or Zephyr.
* **Maven/Gradle**: For build automation and dependency management.

**7. Data Utilities**

* **Faker / Java Faker**: For generating random test data (e.g., names, emails, addresses).
* **Parameterization**: Using Excel, JSON, CSV, or property files for test data.
* **Data Masking**: Techniques to mask sensitive information in logs or reports.

**8. Database Utilities**

* **JDBC**: For connecting to databases and running queries for validation.
* **Hibernate / JPA**: For interacting with databases at an ORM level.

**9. Performance Testing Utilities**

* **JMeter**: For performance and load testing.
* **Gatling**: Another tool for load and performance testing.

**10. Utilities for Test Design**

* **Page Object Model (POM)**: For better test structure and maintainability.
* **Data-Driven / Keyword-Driven / Hybrid Frameworks**: Understanding different test design approaches.

**11. Miscellaneous Utilities**

* **Docker**: For containerized test environments.
* **Git**: For version control and collaboration.
* **Cross-Browser Testing**: Using tools like BrowserStack, Sauce Labs, or LambdaTest.

**12. Core Java Utilities for Automation**

* **Collections Framework**: For handling lists, sets, maps, and queues.
* **Stream API**: For advanced data manipulations.
* **Java 8+ Features**: Lambda expressions, functional interfaces, etc.
* **Exception Handling**: For robust test automation.

**13. Mocking and Stubbing Utilities**

* **Mockito**: For mocking dependencies in unit tests.
* **WireMock**: For mocking APIs during integration tests.

**14. Version and Compatibility Utilities**

* **Dependency Management**: Understanding how to manage and resolve conflicts.
* **Driver-Browser Compatibility**: Using WebDriverManager for seamless updates.

**15. Security and Validation Utilities**

* **JWT Token Validation**: Libraries for decoding and validating JWT tokens.
* **Encryption/Decryption**: For handling sensitive data securely.

**Bonus: Behavioral/Conceptual Questions**

* How to debug failing tests in Selenium?
* Explain the utility of WebDriverWait in test synchronization.
* How to design reusable test utilities for framework extensibility?