**Adv. Selenium - Web Driver Advance**

1. **What is WebDriver Desired Capabilities class**

* The WebDriver Desired Capabilities class is a mechanism in Selenium used to define browser and environment-specific settings for automated testing. It acts as a container for key-value pairs that specify details like browser type, version, platform, and other configurations that influence how the browser behaves during test execution. Essentially, it allows you to tell the WebDriver what kind of browser environment you want to use for your tests.
* In Selenium WebDriver, the Desired Capabilities class is used to set properties for the WebDriver, such as browser name, version, platform, etc., before the test starts. It was commonly used to define the types of capabilities or settings needed to configure the browser instance for automation.

1. **Validations and its usage in Automation test scripts**

* Validations in automation test scripts are crucial for verifying that the software behaves as expected and meets the defined requirements.
* Validations ensure the software functions correctly and fulfills user needs.
* OR Validations are checks or assertions used in test scripts to verify the actual behavior of an application against the expected behavior. They help determine if a test has passed or failed.
* **Purpose of Validations in Automation :**
* Ensure that the application behaves as expected.
* Catch bugs or unexpected behavior early.
* Provide evidence in reports (pass/fail).
* Maintain consistency across repeated test runs.
* **When to Use Validations in Test Scripts:**
* After actions like login, submit, save, etc.
* To validate UI elements are present (e.g., buttons, labels).
* While checking error messages for invalid inputs.
* During end-to-end flow testing (e.g., item added to cart → confirmation).
* In data-driven tests, to compare actual vs expected results.

1. **Implementing Logs using Apache Logj API**

* Log4j enables logging in Java projects.
* Steps: Add log4j dependency, create log4j.properties, use Logger class in code.

1. **Reading data from Property files using java program**

* Java Properties class reads key-value pairs from config files.
* Example:
* Properties prop = new Properties();
* prop.load(new FileInputStream("config.properties"));

1. **Reading and Writing data from/to Excel files using Apache POI API and JXL API**

* Apache POI: For .xlsx files
* JXL: For .xls files
* Used to manage test data.

1. **What is a frame work**

* A **framework** insoftware development is a **predefined structure** or **set of tools and guidelines** that helps developers build and organize applications more efficiently and consistently.
* In automation testing, a framework is a structured set of guidelines, practices, and tools that standardize and streamline the process of creating and executing automated tests. It provides a foundation for building, organizing, and maintaining automated tests, promoting reusability, and improving the overall efficiency of the testing process.

1. **Types of frame work**

* **Main Types of Automation Frameworks:**

| * Framework Type | * Description | * Key Features |
| --- | --- | --- |
| * 1. Linear Scripting Framework | * Simple and sequential test scripts without modularization. | * - Record & playback - Fast to create - Low reusability |
| * 2. Modular Driven Framework | * Divides the test script into small, independent modules. | * - Code reuse - Easy maintenance - Better structure |
| * 3. Data-Driven Framework | * Test data is separated from scripts and stored in external files (Excel, CSV, etc.). | * - Run same test with multiple data sets - Uses loops & variables |
| * 4. Keyword-Driven Framework | * Uses keywords (e.g., "Click", "Input", "Verify") to define actions in external files. | * - Non-programmers can write tests - High reusability - Data & logic separated |
| * 5. Hybrid Framework | * Combines two or more frameworks (like Data-Driven + Keyword Driven). | * - Flexible - Robust - Reusable & maintainable |
| * 6. Behavior Driven Development (BDD) Framework | * Uses natural language (like Gherkin) to write test cases (Given-When-Then). | * - Easy collaboration between devs/testers/business - Tools: Cucumber, Behave |

* **Which Framework to Choose?**
* Small projects ➝ Linear or Modular
* Projects with lots of test data ➝ Data-Driven
* Non-technical testers involved ➝ Keyword or BDD
* Complex enterprise testing ➝ Hybrid or BDD

1. **Data driven frame work**

* A Data-Driven Framework is a type of test automation framework where test data is separated from test scripts, allowing the same script to run multiple times with different sets of data.
* **Common Tools Used in Data-Driven Frameworks:**

|  |  |
| --- | --- |
| Language : | Tools : |
| Java | TestNG, Apache POI (for Excel), JUnit |
| Python | PyTest + parametrize, openpyxl, pandas |
| JavaScript | Cypress + Fixtures |
| C# | NUnit + ExcelDataReader |

* **Advantages:**
* Easy to test with multiple data sets
* Reduces code duplication
* Scalable and maintainable
* **Disadvantages:**
* More setup effort (for reading external files)
* Requires good error handling
* Data file management is essential

1. **Modular driven frame work**

* A Modular Driven Framework is an approach where the test script is broken down into smaller, reusable modules, each representing a part of the application under test (AUT). These modules are then combined to build larger test scenarios.
* **Feature : Description**
* Modularization - Test scripts are divided based on functionalities (e.g., Login, Search, Logout).
* Reusability - Modules can be reused across multiple test cases.
* Maintainability - Easier to update a module if functionality changes.
* Function Libraries - Common functions are stored separately (e.g., Login(), AddToCart(), etc.).
* **Advantages:**
* High reusability
* Better test maintenance
* Easier debugging and updates
* Promotes clean code separation
* **Disadvantages:**
* Initial setup may be time-consuming
* Requires good planning of modular boundaries
* May need coordination among team members on module naming

1. **Keyword driven frame work**

* A Keyword-Driven Framework is a type of test automation framework where all operations and instructions are written using keywords in external data sources (e.g., Excel). The actual logic to interpret and execute those keywords is written in the automation code.
* **Feature : Description**
* Keywords = Actions - Uses simple words like Click, Input, Select, Verify to define actions.
* External Data - Test steps and keywords are stored in Excel, CSV, or XML files.
* Non-technical Friendly - Allows manual testers or non-programmers to write tests.
* Separation of Concerns - Business logic is separated from the automation code.
* **Example Test Case in Excel:**

| **Test Case ID** | **Action** | **Object** | **Value** |
| --- | --- | --- | --- |
| TC001 | OpenURL | browser | <http://app.com> |
| TC001 | Input | username\_field | user1 |
| TC001 | Input | password\_field | pass123 |
| TC001 | Click | login\_button |  |
| TC001 | Verify | welcome\_text | Welcome User |

* **Advantages:**
* Very user-friendly
* Allows non-technical testers to contribute
* Highly reusable and maintainable
* Easy to understand and modify tests
* **Disadvantages:**
* Complex to design initially
* Debugging is harder if Excel and code are out of sync
* Can be slow if not optimized (due to parsing external files
* **When to use it :**
* In large teams with both technical and non-technical testers.
* When reusability of actions is critical.
* If business analysts or QA want to define tests in plain language.

1. **Hybrid Framework**

* A Hybrid Framework is a combination of two or more automation frameworks — typically Data-Driven, Keyword-Driven, and Modular — to leverage the strengths of each and provide flexibility, reusability, and scalability in test automation.
* **Feature - Description**
* Flexible Design - Integrates multiple frameworks based on project needs.
* Supports Reusability - Uses modular functions for repeated tasks.
* Data & Keyword Separation - Test steps and test data are kept separate from code.
* Business + Technical Friendly - Can be used by both technical testers and non-technical stakeholders.
* Custom Logic - Can integrate reporting, logging, exception handling, retry logic, etc.
* **Example Test Case in Excel:**

| **Test Case** | **Keyword** | **Locator** | **Value** |
| --- | --- | --- | --- |
| TC001 | OpenURL | browser | <http://app.com> |
| TC001 | Input | username\_field | user1 |
| TC001 | Input | password\_field | pass123 |
| TC001 | Click | login\_button |  |
| TC001 | Verify | welcome\_text | Welcome |

* **Advantages of Hybrid Framework:**
* Highly flexible and scalable
* Combines reusability, easy data management, and non-technical accessibility
* Easy to extend with tools like logging, reporting, retry logic, etc.
* **Disadvantages:**
* More complex to design and maintain
* Requires good understanding of multiple framework concepts
* Initial setup can take time and planning

1. **Introduction about maven**

* Maven is an automation and management tool developed by Apache Software Foundation.
* It was initially released on 13 July 2004.
* In Yiddish language the meaning of Maven is "accumulator of knowledge".
* It is written in Java Language and used to build and manage projects written in C#, Ruby, Scala, and other languages.
* It allows the developer to create projects using Project Object Model and plugins.
* It helps to build projects, dependency, and documentation.
* Its development process is very similar to ANT.However, it is much advanced than ANT.
* Maven is also able to build any number of projects into desired output such as jar, war, metadata.

1. **Ant v/s Maven**

|  |  |
| --- | --- |
| **Ant** | **Maven** |
| Ant is procedural, you need to provide information about what to do and when to do through code. You need to provide order. | Maven is declarative, everything you define in the pom.xml file. |
| It is less preferred than Maven. | It is more preferred than Ant. |
| There is no life cycle in Ant. | There is life cycle in Maven. |
| It is a tool box. | It is a framework. |
| It is mainly a build tool. | It is mainly a project management tool. |
| The ant scripts are not reusable. | The maven plugins are reusable. |

1. **Installation of Maven**

* To install maven on windows, you need to perform following steps: 1. Download maven and extract it 2. Add JAVA\_HOME and MAVEN\_HOME in environment variable 3. Add maven path in environment variable 4. Verify Maven.

1. **Maven Structure**

* Folders: src/main/java, src/test/java, pom.xml
* pom.xml defines dependencies and plugins.

1. **Maven Dependencies**

* Dependencies are added in pom.xml.
* Example:
* <dependency>
* <groupId>org.seleniumhq.selenium</groupId

1. **Maven Repositories**

* Maven Central Repository: Default source for dependencies.
* Can define custom repositories too.

1. **Maven Eclipse Integration**

* Use m2e plugin or import via Eclipse -> Import -> Maven -> Existing Project.