Orange Days

Detailed Syllabus

Selenium Java Cucumber Framework Design and Development Program

Selenium Java Cucumber Framework Design and Development Program

Design in class. Develop in parallel. Discuss challenges during the sessions.

Syllabus

Program Summary

This program is for Selenium Java automation testers who want to move from script writing to framework design and development.

The program focuses on designing and developing a real Selenium Java Cucumber framework step by step.

In class, the trainer will explain the framework design, code flow, best practices, design patterns, common mistakes, and anti-patterns.

In parallel, participants will develop the framework and discuss their challenges during the sessions.

Target Audience

This program is mainly for Selenium Java automation testers with 2 to 6 years of experience.

This program is suitable for testers who:

- Know basic Java
- Know basic Selenium WebDriver
- Have written Selenium scripts
- Have used an existing framework
- Have not designed and developed a framework themselves
- Want to build Selenium Java Cucumber framework confidence
- Want to explain framework-related interview questions clearly

This program is not for complete beginners.

Program Outcome

After this program, you should be able to:

- Design a Selenium Java Cucumber framework with clear component responsibilities
- Develop the framework step by step
- Build feature files, step definitions, hooks, page classes, reusable components, utilities, config, reports, logs, and test data handling
- Understand the complete framework execution flow
- Debug framework-level issues with more confidence
- Add new scenarios, pages, and modules in a maintainable way
- Avoid common framework design mistakes
- Explain your framework architecture, execution flow, component responsibilities, and design decisions confidently in interviews
- Speak about framework development as someone who has designed and developed the framework, not just used it

Detailed Syllabus

Session 1: Framework Mindset and Project Setup

Topics Covered

- Difference between Selenium scripting and framework development
- Why many testers get stuck at script-writing level
- What a framework should solve in a real project
- Overview of Selenium Java Cucumber framework
- Framework components and their responsibilities
- Initial project setup
- Maven project structure
- Required dependencies
- Basic folder and package structure

Practical Work

- Create the base project
- Add required dependencies
- Set up the initial framework folders
- Understand the planned framework structure

Outcome

You will understand the framework direction and prepare the base project setup.

Session 2: Cucumber Layer Design

Topics Covered

- Role of Cucumber in the framework
- Feature file design
- Writing business-readable scenarios
- Avoiding technical logic inside feature files
- Scenario and scenario outline usage
- Step definition structure
- Mapping feature file steps to Java code
- Common Cucumber design mistakes

Practical Work

- Create feature files
- Write sample scenarios

- Create step definition classes
- Connect feature files with step definitions

Outcome

You will understand how the Cucumber layer should be designed and connected to automation code.

Session 3: Page Object Model Design

Topics Covered

- Why Page Object Model is needed
- Page class responsibility
- Locator handling
- Page action methods
- Page-level validations
- Connecting step definitions with page classes
- Avoiding heavy logic inside step definitions
- Common POM mistakes

Practical Work

- Create page classes
- Add locators and page actions
- Connect step definitions to page classes
- Build flow using feature file, step definition, and page object

Outcome

You will understand how to design page classes and connect them with Cucumber step definitions.

Session 4: Browser Setup, Hooks, and Execution Flow

Topics Covered

- Browser setup design
- Driver lifecycle
- Hooks in Cucumber
- Before and After hooks
- Setup and teardown activities
- Scenario-level execution control
- Screenshot capture on failure
- Complete execution flow from feature file to browser action

Practical Work

- Create hook class
- Add browser setup
- Add teardown logic
- Capture screenshot on failure
- Execute scenarios through the framework

Outcome

You will understand how framework execution starts, flows, and ends.

Session 5: Reusable Components and Common Actions

Topics Covered

- Why reusable components are needed
- Common browser actions
- Wait handling
- Click, type, select, get text, and validation helpers
- Screenshot helper
- Exception handling approach
- Avoiding duplicate code
- Avoiding utility dumping

Practical Work

- Create reusable action methods
- Add wait handling
- Add screenshot helper
- Refactor repeated code into reusable components

Outcome

You will understand how to create reusable framework components with clear responsibility.

Session 6: Config, Environment, and Test Data Handling

Topics Covered

- Why config handling is important
- Environment-based execution
- QA, UAT, and production-like config handling
- Reading browser, URL, and other framework values from config
- Test data handling approach
- Avoiding hardcoded test data

- Passing test data into scenarios and code

Practical Work

- Create config file
- Read values from config
- Use config values in framework execution
- Add practical test data handling approach

Outcome

You will understand how to manage environment details and test data without hardcoding values everywhere.

Session 7: Reports, Logs, and Debugging Framework Failures

Topics Covered

- Why reporting is important
- What a useful report should show
- Screenshot attachment on failure
- Logging design
- How logs help debugging
- Understanding failure points
- Debugging feature file, step definition, page class, utility, and config issues
- Common framework debugging mistakes

Practical Work

- Add reporting support
- Add logs
- Attach screenshot on failure
- Debug sample framework failures
- Understand failure flow

Outcome

You will understand how to make failures easier to analyze and debug.

Session 8: Framework Maintainability and Best Practices

Topics Covered

- How to keep framework clean as it grows
- Adding new modules
- Adding new pages

- Adding new scenarios
- Avoiding duplicate step definitions
- Avoiding over-generic reusable methods
- Utility class responsibility
- Package organization
- Common design patterns
- Common anti-patterns

Practical Work

- Add a new module or scenario
- Extend the framework without breaking existing flow
- Refactor duplicate or poor design areas
- Discuss better framework design choices

Outcome

You will understand how to maintain and extend the framework in a real project style.

Session 9: Complete Framework Review and Execution Flow

Topics Covered

- Complete framework walkthrough
- Feature file to step definition flow
- Step definition to page class flow
- Page class to reusable component flow
- Config and test data flow
- Hook execution flow
- Report and screenshot flow
- Framework execution from start to end

Practical Work

- Run the full framework
- Review the complete execution flow
- Trace how a scenario moves through different framework layers
- Identify and fix weak areas

Outcome

You will understand the full framework flow and how every component connects.

Session 10: Interview Explanation and Framework Discussion

Topics Covered

- How to explain framework architecture
- How to explain Cucumber framework flow
- How to explain feature files, step definitions, hooks, POM, utilities, config, reports, and logs
- How to answer design decision questions
- How to explain debugging approach
- How to explain framework maintainability
- Common framework interview questions
- Mistakes candidates make while explaining frameworks

Practical Work

- Practice explaining framework architecture
- Discuss framework interview questions
- Review participant implementation challenges
- Discuss improvements and better explanations

Outcome

You will be able to explain your Selenium Java Cucumber framework confidently in interviews.

Implementation Approach

How You Will Learn

In Class

The trainer will explain framework design, code flow, best practices, design patterns, common mistakes, and anti-patterns.

In Parallel

You will develop the framework on your side.

During Discussion

You will bring your implementation challenges, errors, doubts, and design confusion to the class.

For Interviews

Important topics will be connected to interview explanations.

What Makes This Program Practical

This program is not only about watching a trainer code.

You will:

- Understand the design
- Develop the framework yourself
- Make mistakes
- Discuss challenges
- Fix issues
- Improve the design
- Understand the complete execution flow
- Learn how to explain framework decisions

Framework confidence comes when you build the framework yourself.

Program Details

Program Name

Selenium Java Cucumber Framework Design and Development Program

Fee

Rs.10,000

Morning Batch

20 July to 24 July
27 July to 31 July
7:30 AM to 10:00 AM IST

Evening Batch

20 July to 24 July
27 July to 31 July
7:30 PM to 10:00 PM IST

Free Class Details

You can attend the free class before joining the paid program.

Free Class Dates

- 11 July - 10:00 AM to 2:00 PM IST
- 12 July - 10:00 AM to 2:00 PM IST
- 18 July - 10:00 AM to 2:00 PM IST
- 19 July - 10:00 AM to 2:00 PM IST

Attend the free class first.
Join the paid program only if you feel it is useful for you.

Contact

For questions, contact:

support@orangedays.in

WhatsApp: +91 8217394357