

# Orange Days

## Selenium Java Cucumber Framework Masterclass - Detailed Course Outline

Free 4-Hour Selenium Framework Class + Paid Selenium Java Cucumber Framework Masterclass

### Program Promise

You may know Selenium commands. This program helps you understand how a professional automation framework is designed, reviewed, debugged, and explained in interviews.

If frameworks feel like a black box today, Orange Days helps you understand the moving parts one layer at a time.

### Free Class Schedule

Date	Slot	Time
July 11	Morning	10:00 AM - 2:00 PM IST
July 11	Evening	3:00 PM - 7:00 PM IST
July 12	Morning	10:00 AM - 2:00 PM IST
July 12	Evening	3:00 PM - 7:00 PM IST

### Paid Masterclass Batches

**Morning Batch:** July 13-17 and July 20-24, 7:30 AM - 10:00 AM IST (25 hours)

**Evening Batch:** July 13-17 and July 20-24, 7:30 PM - 10:00 PM IST (25 hours)

**Weekend Batch:** July 18, 19, 25, 26, 10:00 AM - 1:30 PM and 2:30 PM - 6:00 PM IST (28 hours)

**Fee:** Rs. 10,000 introductory batch fee. Online payment is accepted through Razorpay. Cards, UPI, netbanking, and other Razorpay-supported payment options can be used.

### Free Class vs Paid Masterclass

Free 4-Hour Selenium Framework Class	Paid Masterclass
Understand the framework big picture	Build and understand framework components deeply
Reference framework walkthrough	Component-by-component implementation and review
Cucumber framework preview	Detailed Cucumber framework layer design
Interview explanation preview	Full interview explanation playbook and practice questions

# Concise Curriculum

## 1. Framework Mindset and Architecture

Why automation frameworks are needed; framework problems and capabilities; architecture overview; reference framework walkthrough; execution flow.

## 2. Maven Primer for Framework Development

pom.xml, dependencies, plugins, Maven project structure, Surefire awareness, runtime parameters, and command-line execution.

## 3. Cucumber Framework Foundation

Gherkin, feature files, scenarios, step definitions, runner design, hooks, tags, Background, Scenario Context, Scenario Outline, DataTable, and Cucumber quality rules.

## 4. Java for Framework Design

Packages, access modifiers, static usage caution, inheritance vs composition, interfaces, abstract classes, exceptions, collections, generics, and clean framework code principles.

## 5. Configuration Layer

Environment config, browser config, headless mode, Grid flag, base URL, API URL, timeout config, runtime parameter priority, and fail-fast validation.

## 6. DriverFactory and Browser Lifecycle

Centralized driver creation, BrowserOptions, local and remote execution, ThreadLocal WebDriver awareness, driver cleanup, and common driver anti-patterns.

## 7. Page Object Model

POM purpose, locator ownership, page behavior methods, object repository vs page object design, page readiness, fluent POM awareness, and POM quality checklist.

## 8. Component Object Model

Header, menu, form, table/grid component design, row search, cell extraction, pagination, row-level actions, and COM quality checklist.

## 9. Core Selenium, Waits, and Stability

Common Selenium problems, explicit waits, business condition waits, spinner/loading waits, stale element thinking, special browser interactions, and stability practices.

## 10. Test Data Strategy

Test data problems, JSON/CSV awareness, builders and factories, unique test data, scenario-owned data, cleanup responsibility, and data folder structure.

## 11. API-Assisted Test Data Setup Samples

Why API-assisted setup helps UI automation; creating prerequisite data; using returned ID/value in UI test; cleanup; separating API setup failure from UI failure.

## 12. Assertions, Upload, Download, and Evidence

Assertion strategy, clear failure messages, upload standards, download strategy, file verification, screenshots, logging, and Allure evidence.

## 13. Suite Execution and Maven Commands

Smoke/regression thinking, Cucumber tag execution, Maven command catalogue, browser/environment parameters, and one-command execution principle.

## 14. Parallel/Grid/CI Readiness Awareness

Parallel execution risks, ThreadLocal, scenario isolation, unique data, unique download folders, Selenium Grid awareness, and CI readiness expectations.

## **15. TestNG-Based Framework Walkthrough**

High-level TestNG framework architecture, test class vs feature file thinking, BaseTest vs Cucumber Hooks, groups vs Cucumber tags, listener vs hook-based evidence.

## **16. Final Framework Review and Interview Explanation**

Framework review checklist, POM quality checklist, COM quality checklist, Cucumber quality checklist, Framework Readiness Scorecard, and Interview Explanation Playbook.

# Signature Learning Assets

## Framework Understanding

- Reference Framework Walkthrough - see how a professional framework connects from feature file to report.
- Framework Problem-to-Capability Map - understand which automation problem each framework component solves.
- Framework Design Decision Playbook - learn why key framework decisions are made and how to explain them.
- Maven Command Catalogue - understand how to run the framework professionally from the command line.

## Quality and Review

- Framework Review Checklist - review whether a framework is clean, maintainable, and explainable.
- POM Quality Checklist - check whether page objects are designed properly instead of becoming locator dumps.
- COM Quality Checklist - review reusable component objects like tables, grids, menus, and forms.
- Cucumber Quality Checklist - evaluate feature files, steps, hooks, tags, Background, and Scenario Context.
- Test / Step Definition Quality Gates - check whether test code and step definitions stay clean, thin, and reusable.
- Framework Readiness Scorecard - measure how ready the framework is for real project usage.

## Interview Confidence

- Interview Explanation Playbook - learn how to explain framework architecture and design decisions confidently.

# Sample Framework Questions You Will Learn to Answer

- Explain your Selenium framework architecture.
- Why do you use DriverFactory?
- Why is static WebDriver risky?
- How do hooks work in your Cucumber framework?
- How do feature files connect to step definitions and page objects?
- What is the difference between POM and COM?
- Where do waits belong in a framework?
- How do you manage test data?
- How do you handle failed test evidence?
- How do you make your framework Grid/CI ready?

# Contact and Registration

Website: [www.orangedays.in](http://www.orangedays.in)

Email: [rajamanthiram@hotmail.com](mailto:rajamanthiram@hotmail.com)

WhatsApp: +91 8217394357