Document Generated: 09/12/2026

Learning Style: Virtual Classroom

Technology:

Difficulty: Beginner

Course Duration: 3 Days

Introduction to Selenium 4.5 (TT3610)

About This Course:

This is a hands-on web testing course focused on practical skills and best practices. Participants will learn to design, implement, and run automated tests using Selenium WebDriver with Java, JUnit 5, and modern Selenium 4.5 features such as Relative Locators, new Window and Tab APIs, updated Actions API, and integration with DevTools.

Course Objectives:

Working in a hands-on learning environment led by our expert team you'll learn to:

- Understand web page testing needs and how Selenium meets those needs
- Analyze a web application from a functional and testing perspective
- Use Selenium constructs to locate elements on a web page
- Test web page forms and elements
- Apply Java constructs relevant for WebDriver automation
- Design, code, and run automated Selenium tests using the Java WebDriver API
- Use the WebDriver API to test advanced features such as delayed responses and Ajax
- Apply modern wait patterns with Duration
- Implement Relative Locators, manage windows and tabs, and leverage advanced user interactions
- Integrate JUnit 5 features with WebDriver for parameterized and resilient tests
- Capture and analyze console logs, network traffic, and performance metrics with DevTools
- Apply best practices and patterns for scalable and maintainable test automation

Audience:

- This is an introductory-level Selenium course, designed for experienced web developers that are familiar with Java

Prerequisites:

- Ideally students should have approximately 6 months to a year of web development working knowledge.

Course Outline:

1) Introduction to Selenium 4.5

Gain a clear understanding of what Selenium is and why it is used for web testing. Explore key terminology and see how automated testing compares with manual approaches. Begin working with a sample web application to identify testing requirements and prepare for the hands-on labs.

- Introduce Selenium
- Selenium concepts and terminology
- Manual vs Automated testing
- Lab: Tutorial: The Web Application
- Lab: Analyzing the Web application

2) A first look at Selenium

Learn how to inspect elements on a page using browser developer tools and translate that knowledge into Selenium commands. Create and run your first Selenium WebDriver test to experience how automation interacts with a live browser.

- A First Look at Selenium
- Using browser developer tools to inspect elements
- Locating elements on a page
- Writing and running the first Selenium WebDriver test
- Lab: Running initial Selenium WebDriver tests

3) JUnit 5 for Selenium

Build a foundation in JUnit 5 by writing structured test cases and applying annotations for setup, execution, and cleanup. Use parameterized tests to simplify repetitive checks and apply timeouts to control test reliability.

- A First Look at Selenium
- Using browser developer tools to inspect elements
- Locating elements on a page
- Writing and running the first Selenium WebDriver test
- Lab: Running initial Selenium WebDriver tests
- Lab: Writing JUnit tests

4) Selenium WebDriver

Work directly with the Selenium WebDriver API to launch and control browsers. Understand how to configure drivers for different browsers and use WebDriverManager to simplify setup and maintenance across environments.

- Exploring the Selenium API
- Installing and configuring drivers for different browsers
- WebDriverManager for driver lifecycle management
- Lab: Demo â€“ Using JUnit with Selenium
- Lab: Basic WebDriver automation

5) Working with Selenium WebDriver

Practice locating elements using ids, names, CSS selectors, and XPath expressions. Interact with text and collections of elements, and apply assertions to validate that a page behaves as expected.

- Locating elements with id, name, CSS, and XPath
- WebElement class usage
- Collections of elements
- Assertions and text validation
- Manipulating elements and attributes
- Lab: WebDriver Locators

6) Working with Forms

Automate common form tasks such as filling in input fields, selecting options, and submitting forms. Explore how to handle alerts, take screenshots, and use the Selenium 4.5 APIs for managing multiple windows and tabs.

- Commands for form input and validation
- Handling complex components
- Working with alerts
- Capturing screenshots (page and elements)
- Managing windows and tabs with new Selenium 4.5 APIs
- Lab: Testing forms with WebDriver
- Lab: Screenshots and multiple windows

7) (Asynchronous) Actions

Use the Actions API to perform advanced interactions including keyboard input, mouse movements, scrolling, and wheel actions. Apply modern waiting techniques to ensure tests work reliably with dynamic content and single-page applications.

- Actions API including keyboard, mouse, and wheel input
- Waiting for events with modern wait patterns
- Handling dynamic and SPA elements
- Lab: Advanced Actions
- Lab: Dynamic pages

8) Selenium Best Practices

Discover proven techniques for writing automation that is clear, maintainable, and efficient. Learn how to manage naming conventions, set up multi-browser testing, and apply consistent standards across a test suite.

- Strategies for preparation and naming conventions
- Multi-browser testing
- Creating high quality, stable tests
- Efficient operations and standards for teams

9) Testing Strategies and Patterns

Explore different types of web tests and how to design them effectively. Create reusable wrappers for Selenium commands, understand when to verify versus assert, and handle delayed exceptions to improve test resilience.

- Types of tests for web applications
- Robust element location strategies
- Wrapping Selenium calls for reusability
- Verifying vs asserting
- Delayed exception handling
- Lab: Assertions and Verification
- Lab: Parameterized Selenium tests

10) Lesson: Introducing the Page Object Model

Organize automation using the Page Object Model to separate test logic from page structure. Create page classes that act as object repositories, making tests easier to read, maintain, and extend.

- Benefits of POM
- Decoupling implementation from web elements
- Object repositories
- Lab: Building a Page Object Mode

11) Page Object Factories

Extend Page Objects with PageFactory to simplify element management. Use annotations like @FindBy and lazy initialization to reduce boilerplate and keep code efficient.

- Page Factory in Selenium
- Using @FindBy annotation
- Lazy initialization in Page Factory
- Lab: Using PageFactory

12) DevTools Integration in Selenium 4.5

Leverage the browserâ€™s DevTools features to capture console logs, monitor network requests, and gather performance metrics. Simulate real-world scenarios by emulating geolocation and network conditions within your tests.

- Accessing the DevTools interface from WebDriver
- Capturing console logs
- Intercepting and analyzing network events

- Capturing JavaScript exceptions and performance metrics
- Emulating geolocation and network conditions
- Lab: Using DevTools with Selenium

Additional Topics: Time Permitting

These topics will be included in your course materials but may or may not be presented during the live class depending on the pace of the course and attendee skill level and participation.

1. Introduction to Selenium Grid 4

Run tests across multiple browsers and machines using Selenium Grid. Learn how to configure a Grid, execute tests in parallel, and scale your automation using Docker or standalone setups.

- Grid architecture and use cases
- Running tests on remote machines
- Parallel execution
- Running a local Grid with Docker or standalone jar
- Lab: Running tests on Selenium Grid