

QA

ADVANCED

4 DAYS

UI Automation Testing with Selenium (Java)

Build a strong foundation in UI automation testing using Selenium with Java by understanding its architecture and core concepts. Learn how to design, execute, and maintain automated web tests while applying best practices through hands on, real world testing scenarios.

Why Selenium (Java)?

Improved software quality

UI automation testing with Selenium (Java) helps businesses identify defects and bugs in their software products before they are released, leading to higher quality and more reliable software.

Cost savings

UI automation testing with Selenium (Java) can help businesses save costs by reducing the need for manual testing and minimizing the risk of defects and bugs in software products.

Faster testing

UI automation testing with Selenium (Java) can significantly reduce the time and effort required to test software applications by automating the testing process, which is faster and less error-prone than manual testing.

What you'll be able to do

- ✓ Gain a basic understanding of the basics of UI automation testing
- ✓ Learn about the architecture of Selenium and understand how it can be used to automate testing processes
- ✓ Learn how to work with Selenium WebDriver to create automated tests for web applications
- ✓ Get introduced to best practices for UI automation testing with Selenium (Java)
- ✓ Get hands-on training opportunities and learn to apply the concepts and techniques

Who this is for

- Software Testers
- Quality Assurance Professionals
- Developers
- Technical leads
- Test Automation Engineers

Curriculum

01 Foundations

- Apply foundations using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

02 Environment Setup

- Apply environment setup using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

03 Locators and Interactions

- Apply locators and interactions using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

04 Waits and Dynamic Elements

- Apply waits and dynamic elements using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

05 Alerts, Frames, and Windows

- Apply alerts, frames, and windows using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

06 POM and Design Patterns

- Apply pom and design patterns using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

07 Data-Driven Tests

- Apply data-driven tests using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

08 Frameworks and Reporting

- Apply frameworks and reporting using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

09 Cross-Browser and Parallel

- Apply cross-browser and parallel using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

10 CI/CD Integration

- Apply ci/cd integration using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

11 Debugging and Best Practices

- Apply debugging and best practices using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks