

Software Quality Assurance & Testing Tools

10-Day Practical Hands-on Training – Table of Contents

Recommended Delivery: Approximately 80% Hands-on / 20% Theory

Day 1 – Software QA, Testing Fundamentals & Jira

- QA vs QC
- SDLC and STLC
- Testing levels and testing types
- Functional and non-functional testing
- Test scenarios and test cases
- Defect lifecycle
- Severity vs priority
- Agile/Scrum testing workflow
- Jira overview and project setup
- Creating stories, tasks and test cases
- Test execution and defect reporting
- Jira dashboards and reports
- Hands-on: Create a Jira project, stories, test cases, execute tests and report defects

Day 2 – Java Programming for Test Automation

- Java fundamentals
- Variables, data types and operators
- Conditional statements and loops
- Arrays and Strings
- Methods, classes and objects
- Constructors
- Inheritance, encapsulation and polymorphism
- Exception handling
- Collections
- File handling and test data
- Creating reusable Java utilities
- Hands-on: Develop Java programs and utilities required for test automation

Day 3 – Selenium WebDriver Fundamentals

- Selenium architecture and WebDriver
- Browser automation
- Locators: ID, Name, Class Name, XPath and CSS Selectors
- WebElement operations
- Text boxes, buttons, checkboxes and radio buttons
- Dropdowns
- Alerts
- Frames
- Windows and tabs
- Mouse and keyboard actions
- Waits and synchronization
- Hands-on: Automate a real web application through a complete business workflow

Day 4 – Advanced Selenium & Page Object Model

- Dynamic web elements
- Explicit and implicit waits
- Advanced XPath
- JavaScriptExecutor
- Screenshots
- Handling dynamic pages
- Page Object Model (POM)
- Page Factory
- Reusable methods and utility classes
- Configuration management
- Test data management
- Assertions and validations
- Hands-on: Build a maintainable Selenium automation framework

Day 5 – TestNG & Maven for Automation

- TestNG architecture and annotations
- Assertions
- Test suites
- Groups
- Parameters
- DataProvider
- Test dependencies
- Parallel execution
- TestNG reports
- Maven fundamentals
- pom.xml and dependency management
- Running automation tests through Maven
- Hands-on: Build and execute a TestNG-based regression suite

Day 6 – API Testing with Postman

- API testing fundamentals
- REST architecture
- HTTP/HTTPS
- GET, POST, PUT, PATCH and DELETE
- Request and response
- Headers and parameters
- Authentication
- JSON
- HTTP status codes
- Postman interface
- Collections and environments
- Variables
- Assertions
- Pre-request scripts and test scripts
- Data-driven API testing
- Hands-on: Build and execute a complete Postman API test collection

Day 7 – API Automation with REST Assured & Java

- Introduction to REST Assured
- GET, POST, PUT/PATCH and DELETE requests

- Request and response specifications
- Headers, parameters and authentication
- JSON validation
- Response extraction
- Assertions
- Serialization and deserialization
- Data-driven API testing
- REST Assured with TestNG
- REST Assured with Maven
- Hands-on: Build an API automation framework using Java, REST Assured and TestNG

Day 8 – Performance Testing with JMeter

- Performance testing fundamentals
- Load, stress, spike and endurance testing
- JMeter architecture
- Test Plan
- Thread Groups
- Samplers and HTTP Request
- Controllers
- Listeners
- Assertions
- Timers
- Parameterization
- CSV test data
- Cookies and sessions
- Recording requests
- Result analysis: response time, throughput and errors
- Hands-on: Create and execute a realistic load-testing scenario

Day 9 – Integrated QA Automation Project

- Requirement analysis
- Test scenario identification
- Test case creation and management
- Jira test management
- Selenium UI automation
- API testing with Postman
- API automation with REST Assured
- Test data management
- Defect identification and reporting
- Regression testing
- Test execution and reporting
- Hands-on Project: Build a combined UI and API automation solution for a realistic application

Day 10 – End-to-End Testing & Final Practical Workshop

- End-to-end testing strategy
- Smoke testing
- Sanity testing
- Regression testing
- Integration testing
- UI and API testing approach

- Automation framework best practices
- Debugging and handling automation failures
- Test execution and reporting
- QA metrics and defect analysis
- Final practical assessment
- Final Hands-on Exercise: Requirement → Jira → Test Cases → Selenium → Postman/REST Assured → JMeter → Defect Reporting → Final Test Report

Recommended Tool Stack

Area	Tools
Test Management	Jira
Programming	Java
UI Automation	Selenium WebDriver
Test Framework	TestNG
Build Management	Maven
API Testing	Postman + REST Assured
Performance Testing	Apache JMeter

Training Approach

- Approximately 80% hands-on and 20% theory.
- Participants work with realistic applications and progressively build automation skills.
- Days 9–10 integrate the tools into an end-to-end QA project.

Detailed Hands-on Lab Plan

Labs and Practical Exercises Included Across the 10-Day Program

Day 1 – Software QA, Testing Fundamentals & Jira

- **Lab 1:** Create a Jira Testing Project – create project, stories, tasks and configure the basic workflow.
- **Lab 2:** Test Scenario & Test Case Design – analyze a user story and create positive/negative test cases.
- **Lab 3:** Test Execution in Jira – execute test cases, record results and attach evidence.
- **Lab 4:** Defect Reporting & Tracking – log defects, assign severity/priority and track the defect lifecycle.
- **Lab 5:** Jira Testing Dashboard – create a dashboard and review testing/defect status.

Day 2 – Java Programming for Test Automation

- **Lab 6:** Java Programming Fundamentals – variables, data types, operators, conditions and loops.
- **Lab 7:** Arrays & String Handling – manipulate and validate test data.
- **Lab 8:** Methods, Classes & Objects – create reusable methods and basic automation-ready classes.
- **Lab 9:** Object-Oriented Programming – practice inheritance, encapsulation and polymorphism.
- **Lab 10:** Exception Handling – handle common runtime and automation-related exceptions.
- **Lab 11:** Java Collections – use ArrayList, HashMap and HashSet for test data.
- **Lab 12:** File & Test Data Handling – read and write external test data.

Day 3 – Selenium WebDriver Fundamentals

- **Lab 13:** Selenium Environment Setup – configure Java, Selenium WebDriver and browsers.
- **Lab 14:** Web Element Locators – practice ID, Name, Class Name, XPath, CSS Selector and Link Text.
- **Lab 15:** Automating Web Forms – automate text boxes, buttons, checkboxes, radio buttons and dropdowns.
- **Lab 16:** Handling Alerts – accept, dismiss and validate browser alerts.
- **Lab 17:** Handling Frames – switch between frames and the main page.
- **Lab 18:** Handling Multiple Windows/Tabs – switch between browser windows/tabs.
- **Lab 19:** Mouse & Keyboard Actions – hover, right-click, double-click, drag/drop and keyboard actions.
- **Lab 20:** Selenium Waits – implement implicit/explicit waits and synchronization.
- **Mini Project 1:** Automate a Complete Business Workflow – Login → Search → Select Product → Add to Cart → Checkout → Logout.

Day 4 – Advanced Selenium & Page Object Model

- **Lab 21:** Dynamic XPath – create robust XPath expressions for changing elements.
- **Lab 22:** Dynamic Web Elements – handle changing attributes and dynamic content.
- **Lab 23:** JavaScriptExecutor – scroll, click and interact with elements using JavaScript.
- **Lab 24:** Screenshot Capture – capture screenshots during execution and on failure.
- **Lab 25:** Page Object Model – create page classes, locators and reusable page methods.
- **Lab 26:** Page Factory – initialize and manage WebElements using Page Factory.
- **Lab 27:** Test Data & Configuration – externalize environment settings and test data.
- **Lab 28:** Build a Selenium Framework – combine tests, pages, utilities, configuration and test data.

Day 5 – TestNG & Maven

- **Lab 29:** First TestNG Test – create tests and use TestNG annotations.
- **Lab 30:** TestNG Assertions – validate expected versus actual results.
- **Lab 31:** TestNG Test Suite – organize and execute multiple tests.
- **Lab 32:** Groups & Parameters – create smoke/regression groups and parameterized tests.
- **Lab 33:** DataProvider – execute data-driven tests with multiple datasets.
- **Lab 34:** Test Dependencies – configure dependent tests.
- **Lab 35:** Parallel Execution – configure and execute tests in parallel.

- **Lab 36:** Maven Automation Project – configure pom.xml, dependencies and Maven execution.
- **Mini Project 2:** Build a TestNG + Maven Regression Framework.

Day 6 – API Testing with Postman

- **Lab 37:** First API Request – send GET requests and analyze request/response data.
- **Lab 38:** CRUD API Testing – execute GET, POST, PUT, PATCH and DELETE operations.
- **Lab 39:** Request Parameters – work with query parameters, path parameters, headers and request bodies.
- **Lab 40:** JSON Validation – validate response fields, values and structure.
- **Lab 41:** API Authentication – practice API key, Bearer token and Basic authentication.
- **Lab 42:** Postman Collections – organize API requests into collections.
- **Lab 43:** Postman Environment Variables – create and use environments and variables.
- **Lab 44:** Postman Assertions – validate status codes, response times and JSON fields.
- **Lab 45:** Pre-request & Test Scripts – create scripts for dynamic test data and validations.
- **Lab 46:** Data-Driven API Testing – execute API tests using multiple datasets.
- **Mini Project 3:** Build a Complete Postman API Test Collection.

Day 7 – API Automation with REST Assured & Java

- **Lab 47:** First REST Assured Test – set up REST Assured and validate a GET request.
- **Lab 48:** REST Assured CRUD Operations – automate GET, POST, PUT/PATCH and DELETE.
- **Lab 49:** Request Specifications – configure base URI, headers, parameters and request body.
- **Lab 50:** Response Validation – validate status, response time, headers and JSON.
- **Lab 51:** JSON Path & Response Extraction – extract values and reuse them in subsequent requests.
- **Lab 52:** Authentication – configure authenticated API requests.
- **Lab 53:** Serialization & Deserialization – convert between Java objects and JSON.
- **Lab 54:** REST Assured + TestNG – create API tests, assertions and suites.
- **Lab 55:** REST Assured + Maven – build and execute API automation through Maven.
- **Mini Project 4:** Build an API Automation Framework using Java + REST Assured + TestNG + Maven.

Day 8 – Performance Testing with JMeter

- **Lab 56:** Create First JMeter Test Plan – configure Test Plan, Thread Group and users.
- **Lab 57:** HTTP Request Testing – create and execute HTTP Request samplers.
- **Lab 58:** Multiple User Load Test – configure users, ramp-up and load execution.
- **Lab 59:** Assertions in JMeter – validate response content and status.
- **Lab 60:** Timers & Controllers – simulate realistic user behavior and control execution.
- **Lab 61:** Parameterization Using CSV – use external data for multiple users.
- **Lab 62:** Cookies & Sessions – handle session-based applications.
- **Lab 63:** JMeter Recording – record browser traffic and convert it into test requests.
- **Lab 64:** Performance Results Analysis – analyze response time, throughput, errors and requests.
- **Mini Project 5:** Create a Realistic JMeter Load Test.

Day 9 – Integrated QA Automation Project

- **Lab 65:** Requirement Analysis – identify testable requirements and functional scenarios.
- **Lab 66:** Test Design – create test scenarios, test cases, positive/negative tests and test data.
- **Lab 67:** Jira Test Management – manage stories, tests, execution and defects.
- **Lab 68:** UI Automation – automate selected scenarios using Java + Selenium + TestNG.
- **Lab 69:** API Testing – test corresponding APIs using Postman.
- **Lab 70:** API Automation – automate selected APIs using REST Assured.
- **Lab 71:** Defect Management – identify failures, determine severity/priority, log and retest defects.
- **Lab 72:** Regression Execution – execute UI/API regression suites and analyze failures.

Day 10 – End-to-End Testing & Final Practical Workshop

- **Lab 73:** Smoke Test Suite – create and execute a focused smoke suite.
- **Lab 74:** Regression Test Suite – execute selected UI and API regression tests.
- **Lab 75:** Debugging Failed Automation – troubleshoot locator, synchronization, assertion, data and API failures.
- **Lab 76:** End-to-End QA Execution – Requirement → Jira → Test Cases → Selenium → Postman/REST Assured → JMeter → Defects → Regression → Report.
- **Lab 77:** Final Practical Assessment – independently complete test design, Selenium automation, API testing/automation, defect reporting and basic performance testing.

Practical Delivery Note

The 77 labs/exercises are designed as progressive practical activities rather than 77 separate standalone lab sessions. The exercises build toward five mini-projects and one integrated end-to-end QA project.