

# Selenium Automation Testing using JAVA Syllabus

1

## Introduction

- What is Automation Testing?
  - Manual Testing vs Automation Testing
  - Why Selenium?
  - Selenium Components & Tools (WebDriver, IDE, Grid)
  - Role of Automation in the Software Development Lifecycle
  - Limitations of Automation Testing
- **Practice:**
1. Explain the advantages of automation testing over manual testing.
  2. Research and list different automation testing tools other than Selenium.

2

## Software Development Lifecycle Models

- **Waterfall Model**
  - Phases: Requirement Analysis, Design, Implementation, Testing, Deployment, Maintenance
  - Pros & Cons
- **V-Model**
  - Verification & Validation
  - Testing in parallel with development.

Pg. 1



# Selenium Automation Testing using JAVA Syllabus

- **Agile Software Development**
  - Scrum, Kanban
  - Agile Manifesto & Principles.
- **Incremental & Iterative Development**
- **Introduction to Scrum**
  - Scrum Roles: Product Owner, Scrum Master, Development Team
  - Scrum Artifacts: Product Backlog, Sprint Backlog
  - Scrum Events: Sprint Planning, Daily Standup, Sprint Review, Sprint Retrospective
- **Difference between Sequential, Iterative, & Incremental Development**
- **Practice:**
  1. Explain Agile Model and how testing is incorporated into it.
  2. Create a test case scenario for an e-commerce product page..

3

## Concepts of Software Testing

- What is Software Testing?
- Importance of Testing in SDLC
- Difference between Static & Dynamic Testing
- Validation vs Verification
- Objectives of Testing
- Testing & Debugging
- Test Process: Planning, Design, Execution, Reporting, Closure

Pg. 2



# Selenium Automation Testing using JAVA Syllabus

- Test Levels: Unit, Integration, System, Acceptance
- Testing Types: Functional, Non-Functional, Regression, Performance, Security.

- **Practice:**

1. Write 5 test cases for a login page.
2. Differentiate between functional and regression testing.

4

## Installations

- Install Java (JDK)
- Install & Set up Eclipse/IntelliJ IDEA
- Install Selenium WebDriver
- Configure Selenium in a Java project
- Install necessary browser drivers (Chrome, Firefox, Edge, etc.)
- Setting up Maven and adding dependencies
- Writing and running the first Selenium test

- **Practice:**

1. Set up a Selenium project in Eclipse and run a basic test case.
2. Write a simple Java program to launch a browser.

Pg. 3



# Selenium Automation Testing using JAVA Syllabus

5

## Basics of Selenium WebDriver

- Introduction to Selenium WebDriver
- Selenium WebDriver Architecture
- WebDriver Interface & Methods
- Managing Browsers (Chrome, Firefox, Edge, etc.)
- Launching & Closing Browsers
- Navigation Commands (back, forward, refresh)
- **Practice:**

1. Write a program to open Google.com and print the title.
2. Write a script to navigate between multiple pages.

6

## Java for Automation

- **Java Basics**
  - Data Types, Variables, Operators
  - Conditional Statements (if-else, switch)
  - Loops (for, while, do-while)
  - Methods & Functions
- **Object-Oriented Programming (OOP)**
  - Classes & Objects
  - Inheritance, Encapsulation, Polymorphism, Abstraction
  - Access Modifiers (public, private, protected, default)

Pg. 4



# Selenium Automation Testing using JAVA Syllabus

- **Exception Handling**

- try-catch-finally
- Throws & Throw
- Custom Exceptions

- **File Handling**

- Read & Write Operations
- Handling CSV, JSON, XML files

- **Collections Framework**

- List, Set, Map, Queue
- Iterators & Streams

- **Advanced Java Concepts**

- Multi-threading & Synchronization
- Java Reflection API
- Handling JSON & XML in Java
- JDBC for Database Connectivity

- **Practice:**

1. Write a Java program to read data from a file.
2. Write a program using Streams to filter a list of web elements.
3. Implement multi-threading in a sample application.
4. Fetch and validate data from a database using JDBC.

Pg. 5



# Selenium Automation Testing using JAVA Syllabus

7

## Locator Techniques & Identifying Web Elements

- **Introduction to Locators**
- **Types of Locators:**
  - ID, Name, Class, TagName, Xpath, CSS Selectors
- **Inspecting Elements using Browser DevTools**
- **Using Selectors Hub Extension for Advanced Element Identification**
- **Practice:**
  1. Identify the locators for elements on a sample website.
  2. Write test scripts using different locator strategies.

8

## Advanced Locators

- **Xpath**
  - Relative, Absolute, Axes (Parent, Child, Sibling, Following, Preceding)
- **CSS Selectors**
  - Attribute Selectors, Pseudo-classes, Pseudo-elements
- **Navigating Between Elements**
- **Handling Dynamic Elements**

Pg. 6



# Selenium Automation Testing using JAVA Syllabus

- **Practice:**

1. Write an XPath expression to find a specific element dynamically.
2. Find an element using a CSS selector and interact with it.

9

## Selenium WebDriver - Automating Web Elements

- **Basic Web Element Interaction**
  - Click, Type, Get Text, Get Attribute
- **Handling Form Elements**
  - Radio Buttons, Checkboxes, Drop-downs
- **Handling Calendar in Selenium (Date Picker Automation)**
- **Handling Dynamic Elements**
- **Practice:**

1. Automate a login form.
2. Select values from a dropdown using Selenium.
3. Automate a date selection from a calendar.

Pg. 7



# Selenium Automation Testing using JAVA Syllabus

10

## Functional Testing with Selenium

- **Testing an E-commerce Application**
  - User Login, Product Search, Add to Cart, Checkout
- **Assertions & Validations**
  - Soft and Hard Assertions
  - Using TestNG for Functional Test Automation
- **Practice:**
  1. Automate login, search, and adding a product to the cart.
  2. Validate the order confirmation page.

11

## Synchronization in Selenium WebDriver

- **Introduction to Synchronization in Selenium**
- **Types of Waits in Selenium:**
  - Implicit Wait
  - Explicit Wait
  - Fluent Wait

Pg. 8



# Selenium Automation Testing using JAVA Syllabus

- **Handling Dynamic Elements & Loading Issues**
- **Best Practices for Synchronization in Selenium**
- **Practice:**
  1. Implement explicit wait in a script.
  2. Handle dynamic elements using synchronization techniques.

12

## Handling Ajax Calls, Child Windows, Tables, iFrames

- **Handling Ajax-based Applications**
  - Automating AJAX Requests
  - Handling Loading Spinners & Progress Bars
- **Handling Multiple Windows & Tabs**
  - Switching Between Windows
  - Closing Specific Windows
- **Handling Frames & iFrames**
  - Switching Between Frames
- **Handling Web Tables**
  - Extracting Data from HTML Tables
  - Handling Dynamic Tables

Pg. 9



# Selenium Automation Testing using JAVA Syllabus

- **Practice:**

1. Automate filling a table dynamically.
2. Handle an alert box using Selenium..

13

## Miscellaneous - Browser Options & Strategies

- HTTPS Certificate Handling
- Proxy Settings, Plugins, Cookies
- Managing Browser Options & Capabilities
- Handling Popups & Alerts
- Automating Broken Links Verification
- Checking HTTP Response Codes
- Taking Screenshots for Test Evidence

- **Practice:**

1. Write a script to validate all links on a webpage.
2. Take a screenshot of a webpage.

14

## Framework - TestNG

Pg. 10



# Selenium Automation Testing using JAVA Syllabus

- Introduction to TestNG
- TestNG Annotations
- Assertions (Hard & Soft Assertions)
- Running Tests in Groups
- Data-Driven Testing using Data Providers
- Generating TestNG Reports
- Parallel Test Execution with TestNG

## • Practice:

1. Implement TestNG in an existing script.
2. Write a TestNG test with data providers.

15

## Framework - Maven

- Introduction to Maven
- Creating a Maven Project
- Adding Dependencies in pom.xml
- Maven Build Lifecycle
  - Clean, Compile, Test, Package, Install
- Running Tests with Maven Surefire Plugin

## • Practice:

1. Convert a Java project into a Maven project.
2. Execute tests using Maven commands.

Pg. 11



# Selenium Automation Testing using JAVA Syllabus

16

## Page Object Model & Factory

- Introduction to Page Object Model (POM)
- Benefits of POM in Automation Frameworks
- Implementing Page Factory in Selenium
- Creating Reusable Page Methods
- Managing Test Data in POM

- **Practice:**

1. Implement POM in a sample project.
2. Create a reusable function for logging in.

17

## Test Configuration & Parallel Runs

- Global Configuration & Environment Setup
- Running Tests in Parallel on Different Browsers
- Managing Test Execution Across Multiple Environments
- Using TestNG XML for Configurations

- **Practice:**

1. Execute tests parallelly on different browsers.
2. Implement global properties and configurations.

Pg. 12



# Selenium Automation Testing using JAVA Syllabus

18

## Execution Control - Test Strategy

- Skipping Tests & Conditional Execution
- Implementing Retry Logic for Flaky Tests
- Managing Dependencies Between Tests
- Test Prioritization & Categorization

- **Practice:**

1. Execute tests using groups and priorities.
2. Implement retry logic for handling flaky tests.

19

## Extent Reports & Listeners

- Introduction to Extent Reports
- Configuring Extent Reports in Selenium Framework
- Customizing Report Formats
- Capturing Screenshots in Reports
- Implementing TestNG Listeners for Logging

- **Practice:**

1. Create an Extent Report for test execution.
2. Capture and attach failure screenshots in Extent Reports

Pg. 13



# Selenium Automation Testing using JAVA Syllabus

20

## Running Tests from Maven

- Running Tests from Command Line Using Maven
- Using Maven Surefire Plugin for Test Execution
- Generating and Analyzing Maven Test Reports

- **Practice:**

1. Execute tests using Maven CLI commands.
2. Generate test reports using Maven plugins.

21

## Cucumber BDD

- Introduction to Behavior-Driven Development (BDD)
- Understanding Gherkin Syntax (Feature Files, Scenarios, Steps)
- Writing Feature Files for Automation Testing
- Integrating Cucumber with Selenium & TestNG
- Data-Driven Testing with Cucumber Examples

- **Practice:**

1. Write a feature file for login functionality.
2. Execute a test using Cucumber and Selenium.

Pg. 14



# Selenium Automation Testing using JAVA Syllabus

22

## Jenkins CI/CD Integration

- Introduction to Continuous Integration/Continuous Deployment (CI/CD)
- Setting Up Jenkins for Automated Test Execution
- Running Selenium Tests in Jenkins Pipeline
- Configuring Jenkins to Trigger Tests on Code Changes
- Generating and Publishing Test Reports in Jenkins

- **Practice:**

1. Configure Jenkins to run Selenium tests automatically.
2. Trigger test execution upon code commit.

23

## Final Project

Objective: Build an end-to-end automation framework using Selenium WebDriver integrated with TestNG, Maven, Cucumber, Extent Reports, and Jenkins CI/CD.

- **Project Scope:**

Automate an E-commerce Website covering the following features:

Pg. 15



# Selenium Automation Testing using JAVA Syllabus

## 1. User Registration & Login

- Automate the sign-up process
- Validate login with valid & invalid credentials

## 2. Product Search & Add to Cart

- Search for a product dynamically
- Apply filters and sort options
- Add the selected product to the cart

## 3. Checkout & Order Confirmation

- Fill in shipping and payment details
- Validate the order confirmation page
- Capture order details for validation

## 4. Data-Driven Testing using Excel (Apache POI)

- Read test data from an external Excel file
- Execute multiple test scenarios using different input values

## 5. Reporting with Extent Reports

- Generate structured HTML reports
- Capture screenshots for test failures
- Include logs for better debugging

## 6. CI/CD Integration with Jenkins

- Configure Jenkins to execute automated tests
- Schedule test runs and trigger execution on code commits
- Generate and analyze test reports in Jenkins



# Selenium Automation Testing using JAVA Syllabus

## **Project Deliverables:**

- Selenium Automation Framework with structured POM (Page Object Model)
- TestNG Test Suite with Parallel Execution enabled
- Maven Build & Execution using pom.xml
- Cucumber Feature Files for BDD testing
- Excel-based Data-Driven Testing Implementation
- Extent Reports Integration for Reporting
- Jenkins Pipeline Configuration for CI/CD

## **Practice Tasks:**

1. Implement user login automation with different test data.
2. Automate the product search and checkout process using dynamic selectors.
3. Validate test execution using Extent Reports and attach screenshots.
4. Configure Jenkins to execute the automated test suite.



**Pg. 17**

