



## **Test automation using Selenium IDE – Java and Web Driver Details:**

**Length: Approx 8 weeks/70 hours**

### **1.1 Selenium Introduction**

- 1.1.1. Selenium History
- 1.1.2. Migrating to Web driver latest Version
- 1.1.3. Selenium 2.0 Web driver Architecture

### **1.2. Installations and Configurations with Java basics**

- 1.2.1. Java Installation
- 1.2.2. Eclipse Installation, configuration
- 1.2.3. Selenium Jars download/Configuration
- 1.2.4. Brush up basic java concepts

### **1.3. Java OOPS Basics for Selenium**

- 1.3.1. First Steps
  - 1.3.1.1. Introduction
  - 1.3.1.2. Creating Your First Java Project
  - 1.3.1.3. Exploring the IntelliJ Interface

#### **1.3.2. Variables, Datatypes and Operators**

- 1.3.2.1. Introduction
- 1.3.2.2. What Are Variables?
- 1.3.2.3. Getting To Know Primitive Data Types – The Byte, Short, IntAnd Long
- 1.3.2.4. Getting To Know Primitive Data Types – Float And Double
- 1.3.2.5. Getting To Know Primitive Data Types – Char And Boolean
- 1.3.2.6. Understanding Strings And Finishing Up Primitive Data Types
- 1.3.2.7. Operators In Java
- 1.3.2.8. More On Operators And Operator Precedence

#### **1.3.3. Java Tutorial: Expressions, Statements, Code blocks, Methods and more**

- 1.3.3.1. Introduction
- 1.3.3.2. Keywords and Expressions
- 1.3.3.3. Statements, Whitespace and Indentation (Code Organization)
- 1.3.3.4. Code Blocks and The If Then Else Control Statements
- 1.3.3.5. Methods in Java
- 1.3.3.6. Method Overloading

#### **1.3.4. Control Flow Statements**

- 1.3.4.1. Introduction
- 1.3.4.2. The switch statement
- 1.3.4.3. The for Statement
- 1.3.4.4. The while and do while statements

#### **1.3.5. OOP Part 1 – Classes, Constructors and Inheritance**

- 1.3.5.1. Introduction
- 1.3.5.2. Classes Part
- 1.3.5.3. Constructors
- 1.3.5.4. Inheritance



### **1.3.6. OOP Part 2 – Composition, Encapsulation, and Polymorphism**

- 1.3.6.1. Introduction
- 1.3.6.2. Composition
- 1.3.6.3. Encapsulation
- 1.3.6.4. Polymorphism

### **1.3.7. Arrays, Java inbuilt Lists, Autoboxing and Unboxing**

- 1.3.7.1. Arrays
- 1.3.7.2. List and ArrayList Part 1
- 1.3.7.3. ArrayList Part 2
- 1.3.7.4. Autoboxing and Unboxing
- 1.3.7.5. LinkedList

### **1.3.8. Inner and Abstract Classes & Interfaces**

- 1.3.8.1. Interfaces
- 1.3.8.2. Inner classes
- 1.3.8.3. Abstract Classes

### **1.3.9. Java Generics**

- 1.3.9.1. Generics Introduction
- 1.3.9.2. Our Generics Class

### **1.3.10. Naming Conventions and Packages. static and final keywords**

- 1.3.10.1. Naming Conventions
- 1.3.10.2. Packages
- 1.3.10.3. Scope
- 1.3.10.4. Access Modifiers
- 1.3.10.5. The static statement
- 1.3.10.6. The final statement

### **1.3.11. Java Collections**

### **1.4. Basic Concepts for first Webdriver program**

- 1.4.1.1. Webdriver Interface explanation and Invoking Browser
- 1.4.1.2. Basic Methods of Webdriver
- 1.4.1.3. How to run tests in Google Chrome
- 1.4.1.4. How to run tests in Internet Explorer

### **1.5. Locator Techniques& Tools used**

- 1.5.1.1. Preview Browser Addons overview to identify elements
- 1.5.1.2. Preview Installing Firebug & Firepath Addons
- 1.5.1.3. Locator Techniques :Xpath identification using Firepath, Name ,ID, ClassName, LinkText,-Handling links

### **1.5.2. ADVANCED WAYS-locating objects**

- 1.5.2.1. Writing Customized xpath Using Attributes
- 1.5.2.2. Writing customizedxpath Using Tagnames Traversing
- 1.5.2.3. CSS Selectors locators

### **1.6. Techniques to automate Web UI**

- 1.6.1. Handle Dynamic dropdowns with Webdriver API
- 1.6.2. Handling Static dropdowns with Select webdriver API



- 1.6.3. Handling Checkboxes with webdriver API
- 1.6.4. Handling Radiobuttons with Customized xpath
- 1.6.5. Handling Radiobutton dynamically- real time examples
- 1.6.6. Types of Alerts present and Methods to handle them
- 1.6.7. Handling Java Alerts using Webdriver API
- 1.6.8. Web Elements Validation
- 1.6.9. End to End Practise Exercise

#### **1.7. Techniques to automate ADVANCED Web UI**

- 1.7.1. Handling Ajax/Mouse Interactions
- 1.7.2. Actions class-real time example
- 1.7.3. Handling Multiple Windows
- 1.7.4. Window Handle concepts-real time example
- 1.7.5. Live Example on working with Child windows
- 1.7.6. Handling ul li Tags in Selenium
- 1.7.7. How to handle Frames?
- 1.7.8. Frames Techniques-real time example

#### **1.8. Real Time Exercises (end to end Programming)**

- 1.8.1. Test Cases- Practise Exercise
- 1.8.2. Exercise 1.1-Limiting Webdriver scope
- 1.8.3. Getting Count of links in the pages, sections
- 1.8.4. Test cases-Practise Exercise-2
- 1.8.5. Exercise 2.1-Dynamic data in Websites
- 1.8.6. Exercise 2.2-Dynamic Links Handling
- 1.8.7. Exercise 2.3-Validations & checkpoints

#### **1.9. Practical problems and Methods to Handle them with Selenium**

- 1.9.1. How to handle table Grids in webpage
- 1.9.2. Techniques used for table grid-Real time example
- 1.9.3. How to overcome Synchronization problems
- 1.9.4. Maximizing window and deleting cookies
- 1.9.5. Handling HTTPS certifications
- 1.9.6. How to troubleshoot if its not invoking in firefox
- 1.9.7. Killing the Process and Cookies using Selenium
- 1.9.8. How to take Screenshots in Selenium

#### **1.10. Data driving from Excel for feeding data**

- 1.10.1. What is Apache POI API & Download Instructions
- 1.10.2. Excel API Methods explanation
- 1.10.3. Program for Retrieving data from excel
- 1.10.4. Program for Updating data back to excel

#### **1.11. Framework Part -1 – TestNG**

- 1.11.1. Why TestNG and Its Advantages
- 1.11.2. TestNG Installation and Setup in Eclipse
- 1.11.3. TestNG Annotations Part
- 1.11.4. Prioritizing the tests using TestNG
- 1.11.5. Disabling Enabling the Testcases and putting Timeout
- 1.11.6. Importance of TestNG xml file
- 1.11.7. Including and excluding the Testcases from Execution with TestNG xml file
- 1.11.8. Importance of Groups in TestNG
- 1.11.9. Data driving Testing with TestNG



**Phone:** 1-732-325-1126 | **E-mail:** nanduv@gmail.com

---

- 1.11.10. DataProvider Annotation -Parameterizing Testcases
- 1.11.11. Parameterising from TestNG xml file
  
- 1.12. Framework Part -2 -Data driven Framework**
  - 1.12.1. Why we should not hard code the data?
  - 1.12.2. How to write Global parameters with java code
  - 1.12.3. Data driving parametrization from Properties file
  - 1.12.4. How to deal with Reusable Components
  
- 1.13. Framework Part 3 – Keyword Driven**
  
- 1.14. Framework Part 4 – Modular Driven, Hybrid**
  
- 1.15. Framework part 5 – ANT-Build Management Tool**
  - 1.15.1. What is Build Management tool?
  - 1.15.2. Installing & configuring ANT
  - 1.15.3. Understanding Build.xml file
  - 1.15.4. Different ANT Commands to trigger framework
  
- 1.16. Framework Part -6 -Jenkins- CI Tool**
  - 1.16.1. Why Jenkins? and where it going to help us in Framework design?
  - 1.16.2. Installing & Configuring Jenkins with Java and ANT
  - 1.16.3. Creating Jenkins project anf integrating Existing Framework
  - 1.16.4. Running the Framework and Scheduling it from Jenkins
  
- 1.17. Framework Part 7 – Page Object Model**
  - 1.17.1. What is Page object model?
  - 1.17.2. Creating Page object Constructor in classes
  - 1.17.3. Practical Exercise explaining Page object Model
  
- 1.18. Cross Browser Testing with Selenium Grid**
  - 1.18.1. How to execute Selenium Tests Remotely
  - 1.18.2. Grid Concepts & Architecture
  - 1.18.3. Configuring Hub and Node
  - 1.18.4. Registering Nodes with Hub Server
  - 1.18.5. Desired Capabilities-Grid Program
  - 1.18.6. Execution Selenium scripts in Remote Machine
  - 1.18.7. Code and Commands

**Disclaimer:** Yes-M Systems and/or their instructors reserve the right to make any changes to the syllabus as deemed necessary to best fulfill the course objectives. Students registered for this course will be made aware of any changes in a timely fashion using reasonable means.