## 🚀 **Selenium Automation Testing Roadmap**

## **(Java + Eclipse + Maven)**

### 🌟 **PHASE 1 — Master the Basics**

✅ Get comfortable with:

* Java basics (if not confident: OOP, interfaces, exceptions, collections)
* Eclipse / Maven project structure
* WebDriver API (ChromeDriver, FirefoxDriver, EdgeDriver)
* Browser commands: get(), getTitle(), getCurrentUrl(), close(), quit()
* Locators: id, name, className, tagName, linkText, partialLinkText, xpath, cssSelector
* Element actions: click(), sendKeys(), clear(), getText(), isDisplayed()
* WebDriverManager for driver setup

💡 Output: Small programs automating search, form fill, login, navigation

### 🌟 **PHASE 2 — Intermediate Selenium**

✅ Learn:

* Handling dropdowns: Select class
* Handling alerts, popups, confirmation dialogs
* Switching windows/tabs
* Frames & iframes
* Taking screenshots
* Wait strategies:
  + Implicit wait
  + Explicit wait (WebDriverWait + ExpectedConditions)
  + Fluent wait
* Mouse actions: Actions class (hover, drag & drop, right-click)
* Keyboard actions

💡 Output: Automate websites with popups, multiple windows, frames

### 🌟 **PHASE 3 — Test Framework Integration**

✅ Learn:

* TestNG (recommended) — test structure, annotations (@Test, @BeforeMethod, @AfterMethod)
* Assertions
* Data-driven testing (Excel, CSV, or @DataProvider)
* Grouping tests, prioritization
* Parallel execution (TestNG XML config)
* Reporting: TestNG default reports, extent reports

💡 Output: Build reusable test cases, e.g., login flow, checkout flow

### 🌟 **PHASE 4 — Advanced Selenium + Design Patterns**

✅ Learn:

* Page Object Model (POM) + Page Factory
* Utility classes for common functions (e.g., wait utils, config readers)
* Singleton WebDriver pattern
* Custom listeners (TestNG Listeners)
* Logging with Log4j or SLF4J
* Headless browser automation (Chrome headless, Firefox headless)

💡 Output: Build a small framework for a demo app

### 🌟 **PHASE 5 — CI/CD + Real-World Readiness**

✅ Learn:

* Maven build life cycle
* Integrate Selenium tests in Jenkins (CI)
* Git/GitHub for version control
* Run tests in cloud (e.g., Selenium Grid, BrowserStack, LambdaTest)
* Cross-browser testing
* API + UI combined tests (optional: RestAssured + Selenium)

💡 Output: Build CI pipeline for Selenium tests

## 📌 **Bonus Learning Areas**

* BDD with Cucumber + Selenium
* Docker for running Selenium Grid
* Advanced XPath and CSSSelector techniques
* Handling dynamic elements + shadow DOM
* Mobile browser automation (Appium intro)

## 🎯 **Suggested Learning Timeline**

| Phase | Time (if 1-2 hours/day) |
| --- | --- |
| Phase 1 | 1-2 weeks |
| Phase 2 | 2 weeks |
| Phase 3 | 2 weeks |
| Phase 4 | 3-4 weeks |
| Phase 5 | 3-4 weeks |