

# SELENIUM INSTALLATION AND SETUP HANDBOOK

Saranya Tiruveedhula

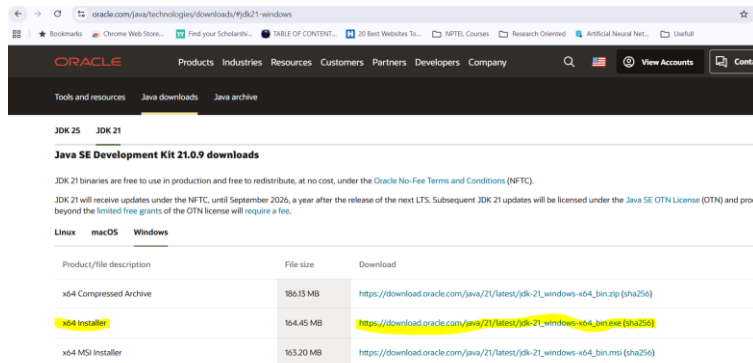
<https://github.com/sarupapu7/Selenium-HandBook>

# Selenium Installation and Setup Handbook

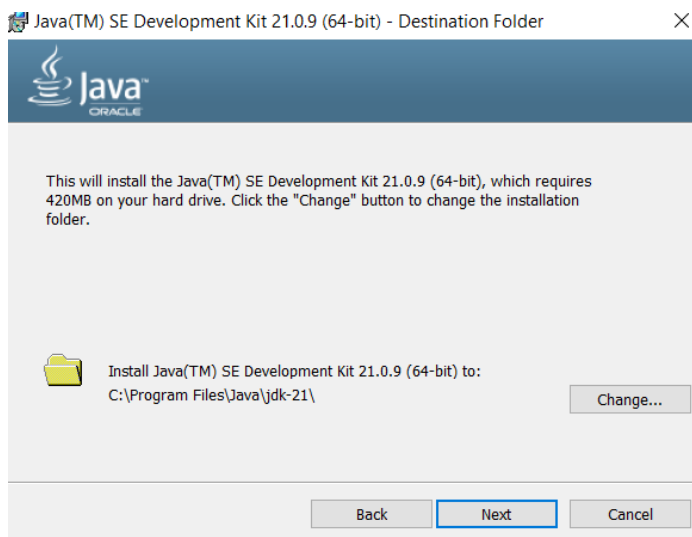
## Step 1: Install Java JDK and set environment variables

- Go to Oracle official site: <https://www.oracle.com/java/technologies/downloads/>
- Choose JDK 21 and download the installer for your OS

Ex: for Windows, refer below screenshot

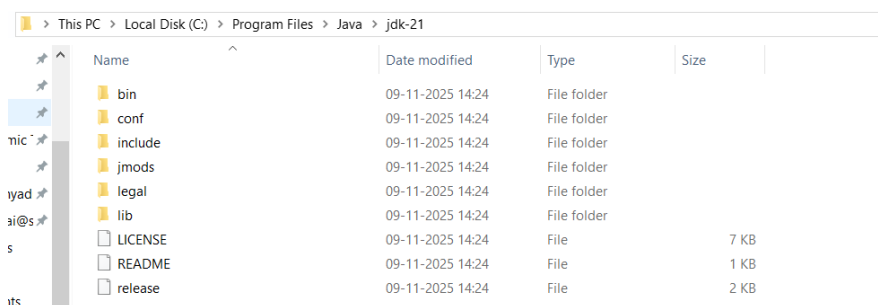


- Run the installer and follow prompts.

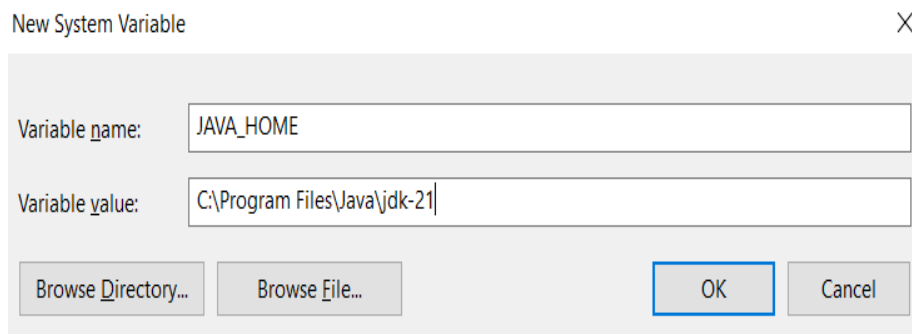
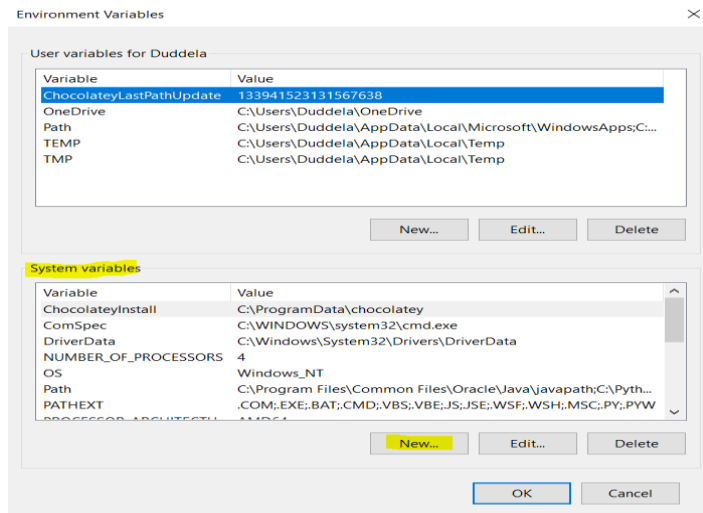


**Note:** If the installer asks to set JAVA\_HOME/add to PATH, choose yes if available. If not, finish install and set manually (steps below).

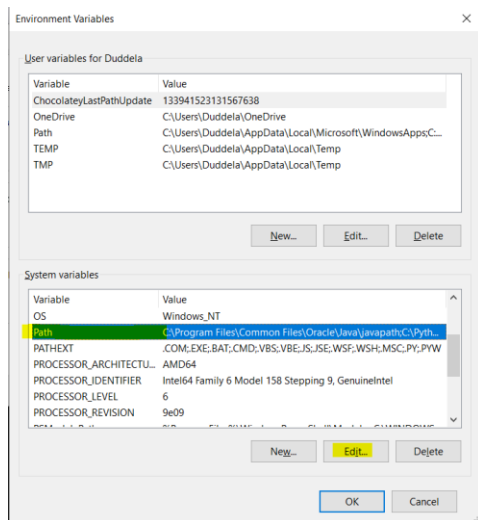
- Use the full path to the JDK installation directory (the folder that contains bin, lib).



- Right-click **This PC** → **Properties** → **Advanced system settings** → **Environment Variables**.
- Under **System variables**:
  - Click **New** → Name: JAVA\_HOME → Value: C:\Program Files\Java\jdk-21 (your actual path).
  - Find Path → Edit → New → %JAVA\_HOME%\bin



- Click Ok. It will be added.
- In the same **Environment Variables** window, scroll down in the **System variables** section, select the variable **Path** and Click **Edit**



- Then, click **New** and add this line: %JAVA\_HOME%\bin
- Click **OK** → **OK** → **OK** on all windows to close.

✓ This tells Windows to look inside your JDK's bin folder for Java commands.

- Now, verify your configuration by opening a **new Command Prompt** (important! old ones won't pick up new variables) and run below commands one by one.

```

Command Prompt
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Duddela>echo %JAVA_HOME%
C:\Program Files\Java\jdk-21

C:\Users\Duddela>java -version
java version "21.0.9" 2025-10-21 LTS
Java(TM) SE Runtime Environment (build 21.0.9+7-LTS-338)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.9+7-LTS-338, mixed mode, sharing)

C:\Users\Duddela>javac -version
javac 21.0.9

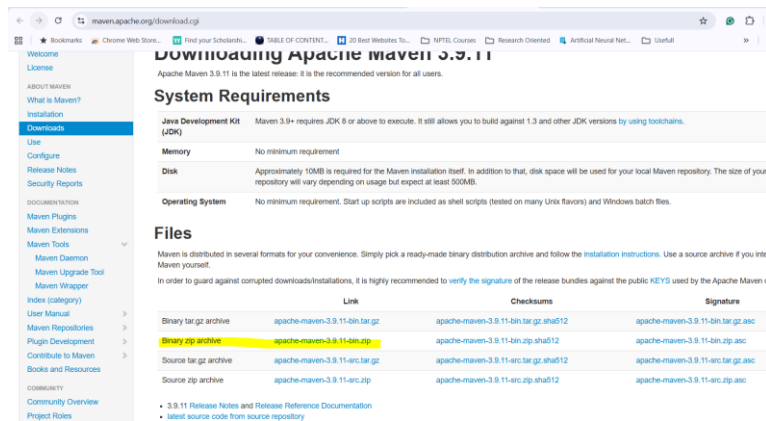
C:\Users\Duddela>

```

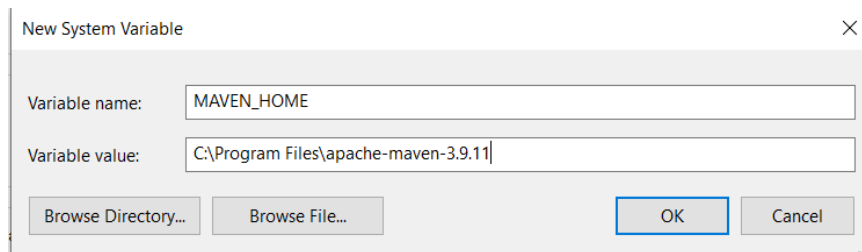
Java is fully configured 😊

## Step 2: Install and configure Maven

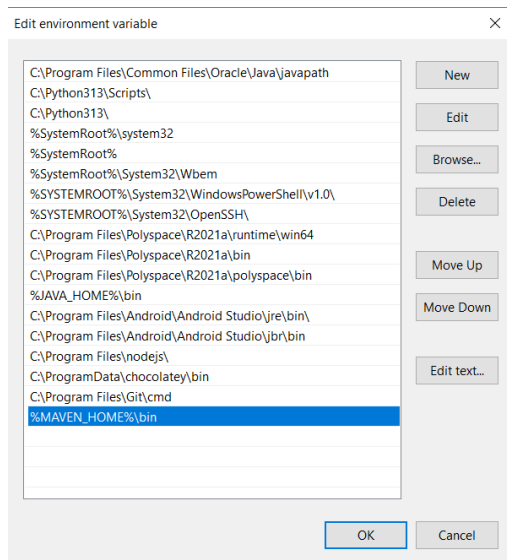
- Download Maven ZIP: 🖱️ <https://maven.apache.org/download.cgi>



- Extract it to a location, e.g.: C:\Program Files\Apache\maven
- Set Environment variables. In **System Variables**, add a new variable:
  - **Variable name:** MAVEN\_HOME
  - **Variable value:** C:\Program Files\Apache\maven



- Add this to the **Path** variable: %MAVEN\_HOME%\bin



- Click OK → OK → OK
- Open a new **Command Prompt** and verify: “mvn -v”

```
Command Prompt
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

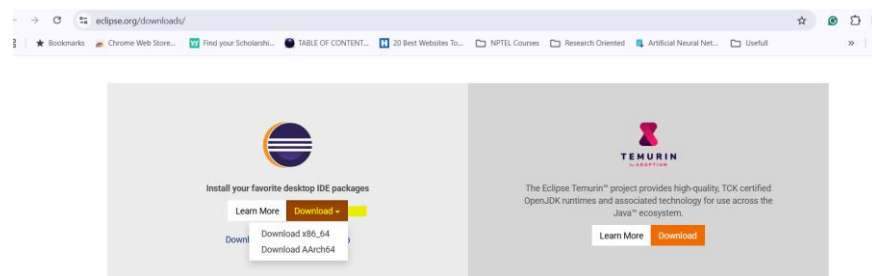
C:\Users\Duddela>mvn -v
Apache Maven 3.9.11 (3e54c93a704957b63ee3494413a2b544fd3d825b)
Maven home: C:\Program Files\apache-maven-3.9.11
Java version: 21.0.9, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-21
default locale: en_IN, platform encoding: UTF-8
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\Duddela>
```

If you see maven version and java version, maven is installed successfully!

### Step 3: Install Eclipse IDE for Java and point to JDK

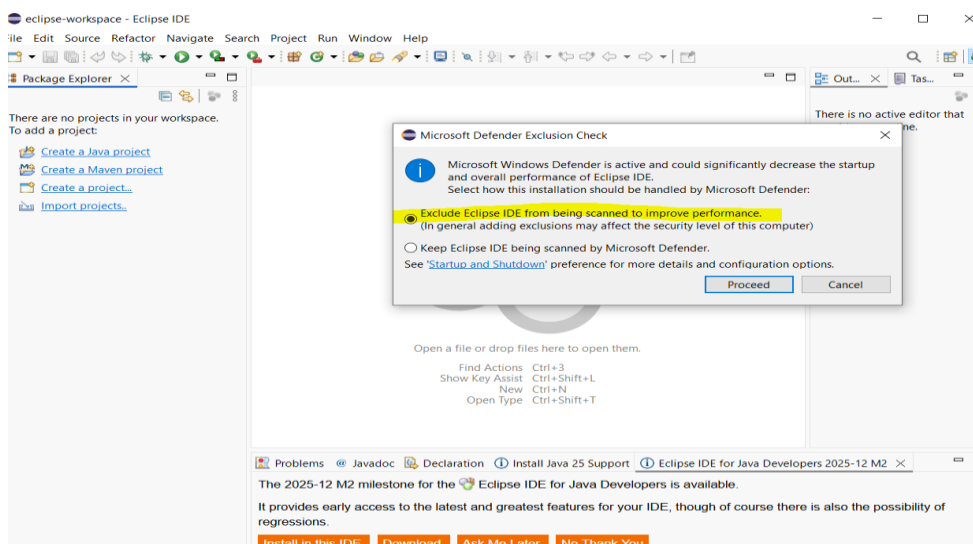
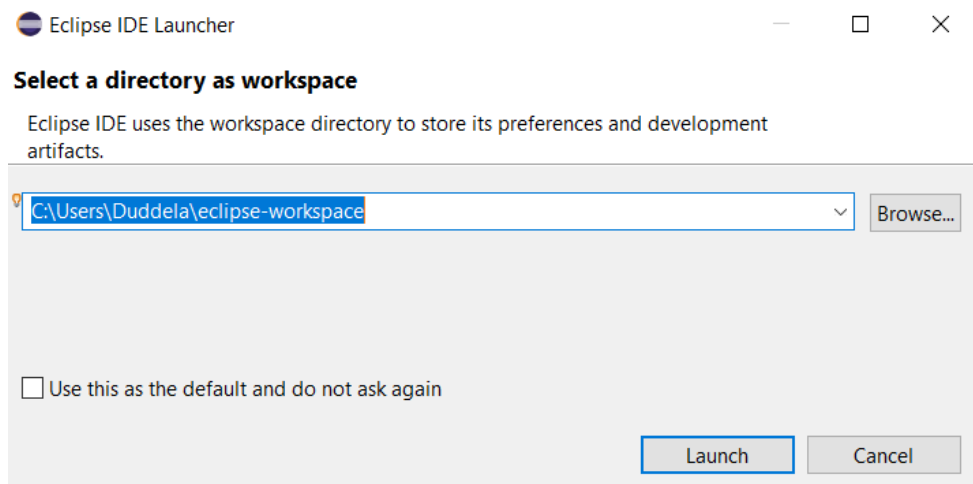
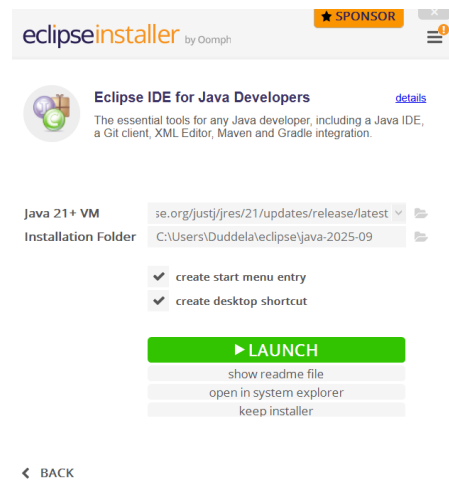
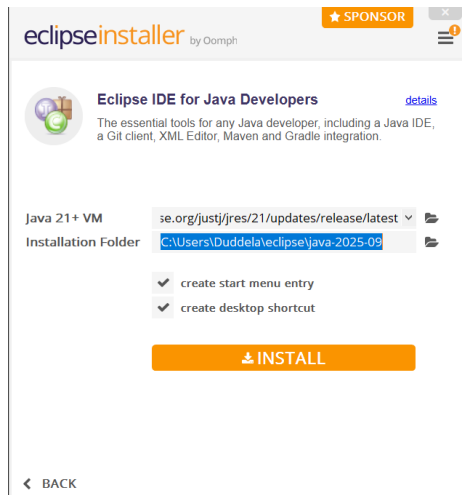
- Go to  <https://www.eclipse.org/downloads/> and download



- Install and open it – Then, Choose Eclipse IDE for Java Developers



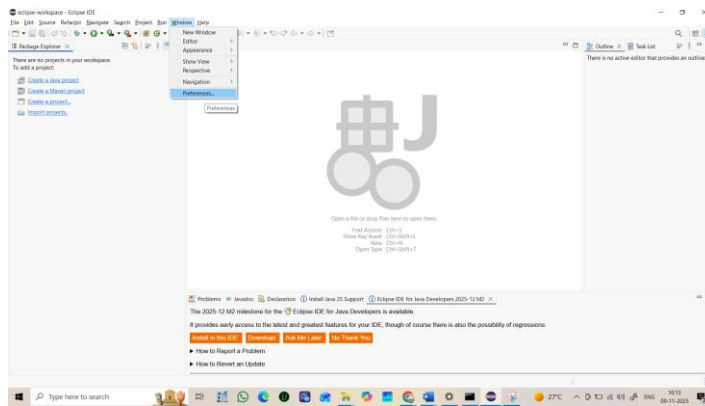
- Select the default location, click install and then launch.



➤ Verify JDK integration

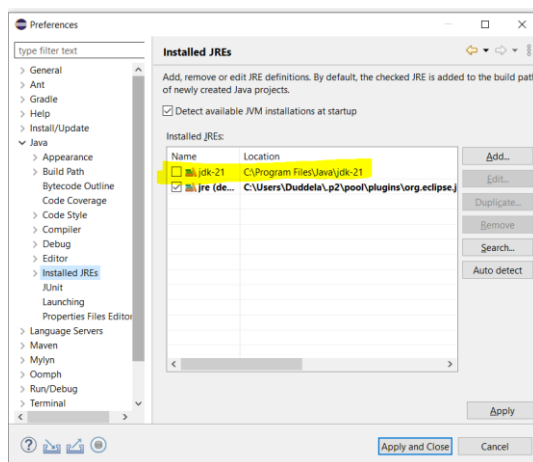
Once Eclipse opens:

1. Go to **Window** → **Preferences**



2. Expand **Java** → **Installed JREs**

3. Make sure you see:



4. If not:

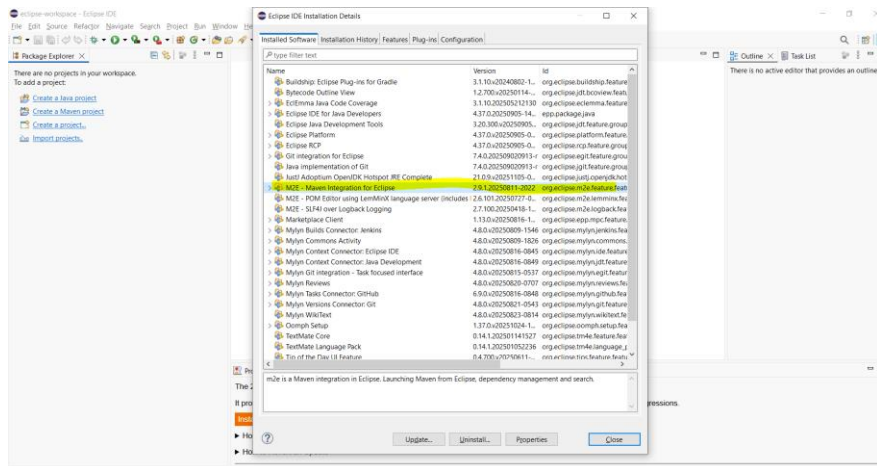
- Click **Add** → **Standard VM**
- Click **Directory**
- Select your JDK folder (C:\Program Files\Java\jdk-21)
- Click **Finish** → **Apply and Close**

Now Eclipse knows where your JDK is!

➤ Verify Maven integration

Eclipse for Java Developers already includes Maven by default. You can confirm:

1. Click **Help** → **About Eclipse IDE**
2. Click **Installation Details** → **Installed Software**
3. You should see “Maven Integration for Eclipse (m2e)”



#### Step 4: Download ChromeDriver

- Selenium cannot open chrome browser, unless ChromeDriver is installed.
- Check your chrome version (Chrome://settings/help)
- Download matching ChromeDriver
- Extract and place it in C and add path in code.

Ex. For chrome browser of version **141.0.7390.123**

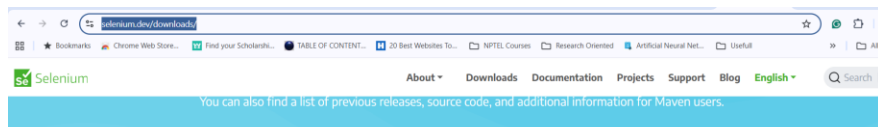
👉 <https://edgedl.me/gvt1.com/edgedl/chrome/chrome-for-testing/141.0.7390.123/win64/chromedriver-win64.zip>

After extracting, the single file should look like below

Clipboard	Organize	New	Open	Select
↑	📁 This PC > Local Disk (C:) > Driver > chromedriver-win64			
Name	Date modified	Type	Size	
chromedriver	20-10-2025 19:16	Application	19,126 KB	
LICENSE.chromedriver	20-10-2025 19:16	CHROMEDRIVER F...	2 KB	
THIRD_PARTY_NOTICES.chromedriver	20-10-2025 19:16	CHROMEDRIVER F...	730 KB	

#### Step 5: Download Selenium Java JARs

- Download link : <https://www.selenium.dev/downloads/>
- Click below link to download latest Selenium java zip.



## Selenium Clients and WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers.

While language bindings for [other languages exist](#), these are the core ones that are supported by the main project hosted on GitHub.



**.NET/C#**  
Stable: [4.38.0 \(October 25, 2025\)](#)  
[Changelog](#)  
[API Docs](#)

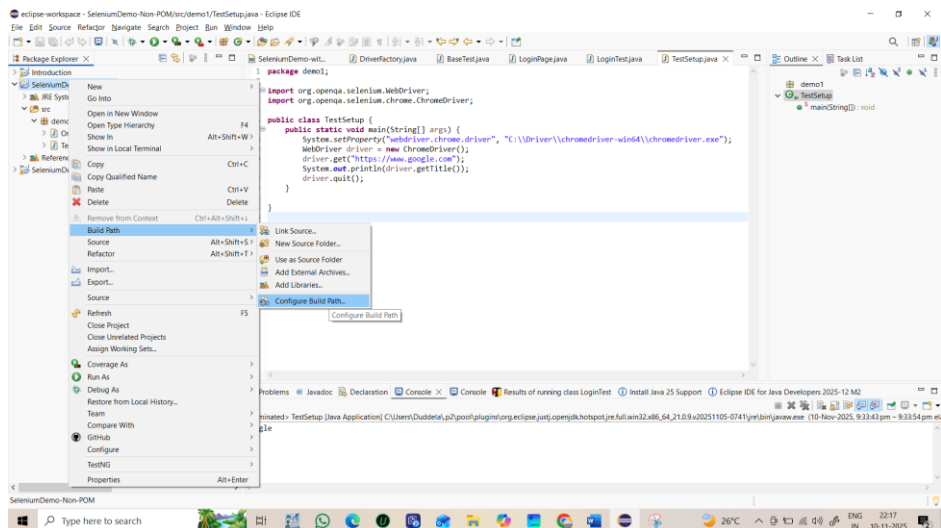


**Ruby**  
Stable: [4.38.0 \(October 25, 2025\)](#)  
[Changelog](#)  
[API Docs](#)

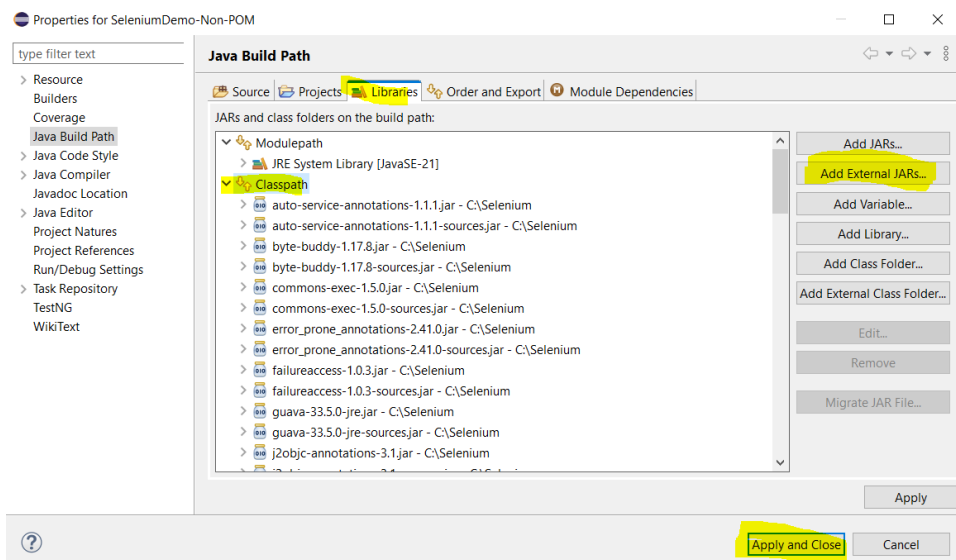


**Java**  
Stable: [4.38.0 \(October 25, 2025\)](#)  
[Changelog](#)  
[API Docs](#)

- Extract the ZIP to C drive.
- Create a Java Project. File → New → Java Project → Name: SeleniumDemo-Non-POM
- Go to configure build path.

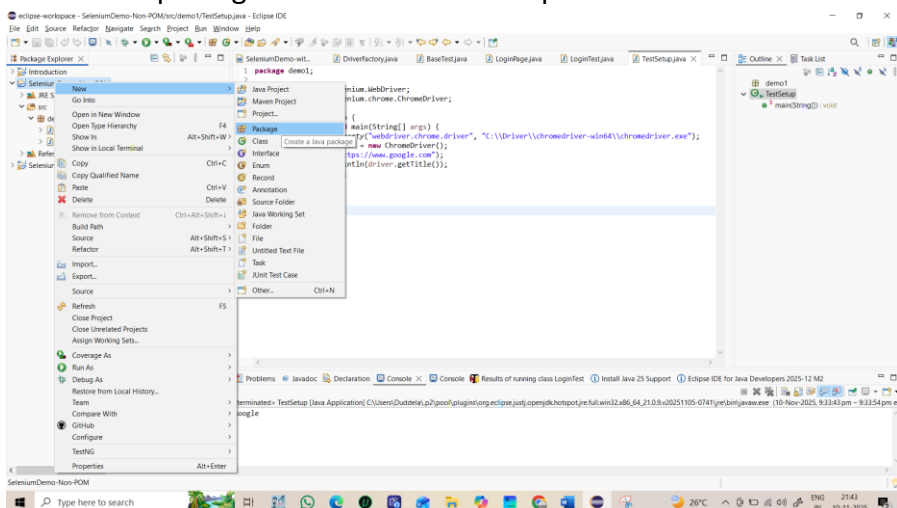


- Under Libraries → select classpath → click Add external JARs → select the location of above extracted ZIP file → click apply and close.



## Step 6: Verify Setup

- Create a package under the above java project and name it demo1. Then create a class in the package and name it TestSetup.



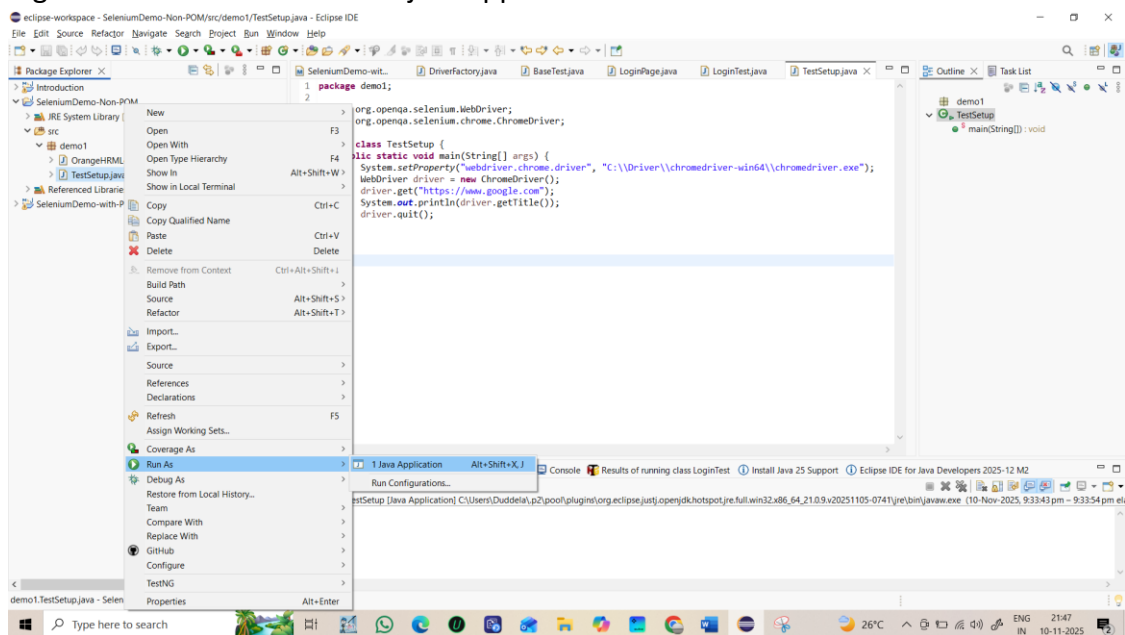
- Paste the below code and run as java application.

```
package demo1;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class TestSetup {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
"C:\\Driver\\chromedriver-win64\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");
        System.out.println(driver.getTitle());
        driver.quit();
    }
}
```

- Right click the class and run as java application



- Expected output is, chrome will be launched and google browser opens and google page title is printed in console.