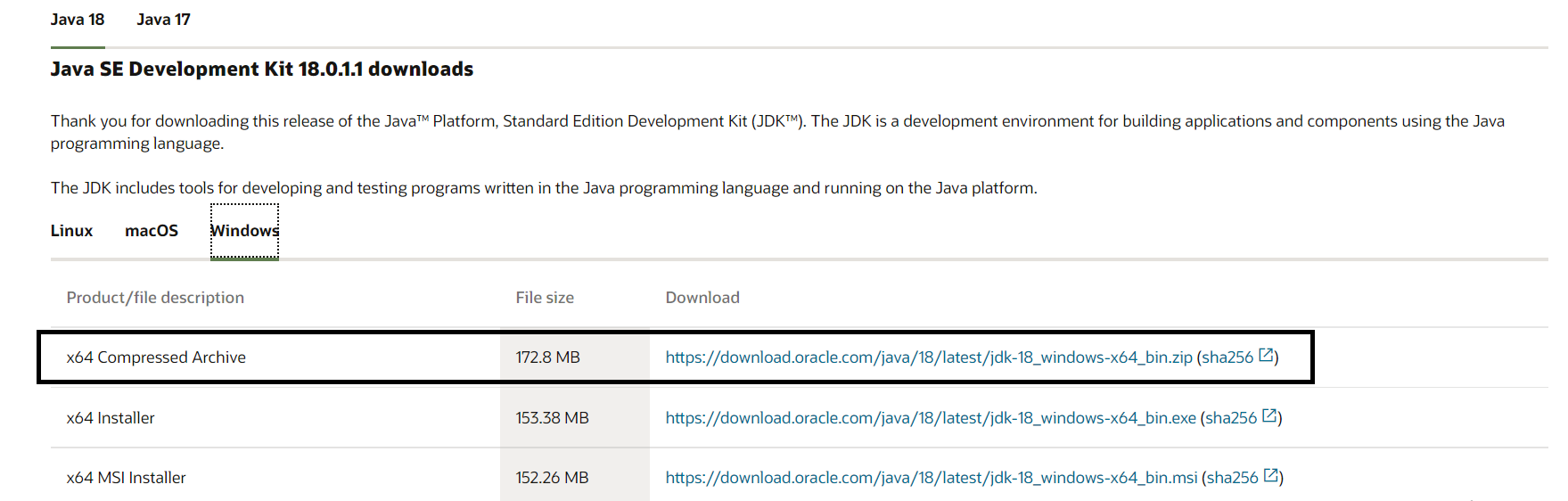
**Download & Install Selenium WebDriver**

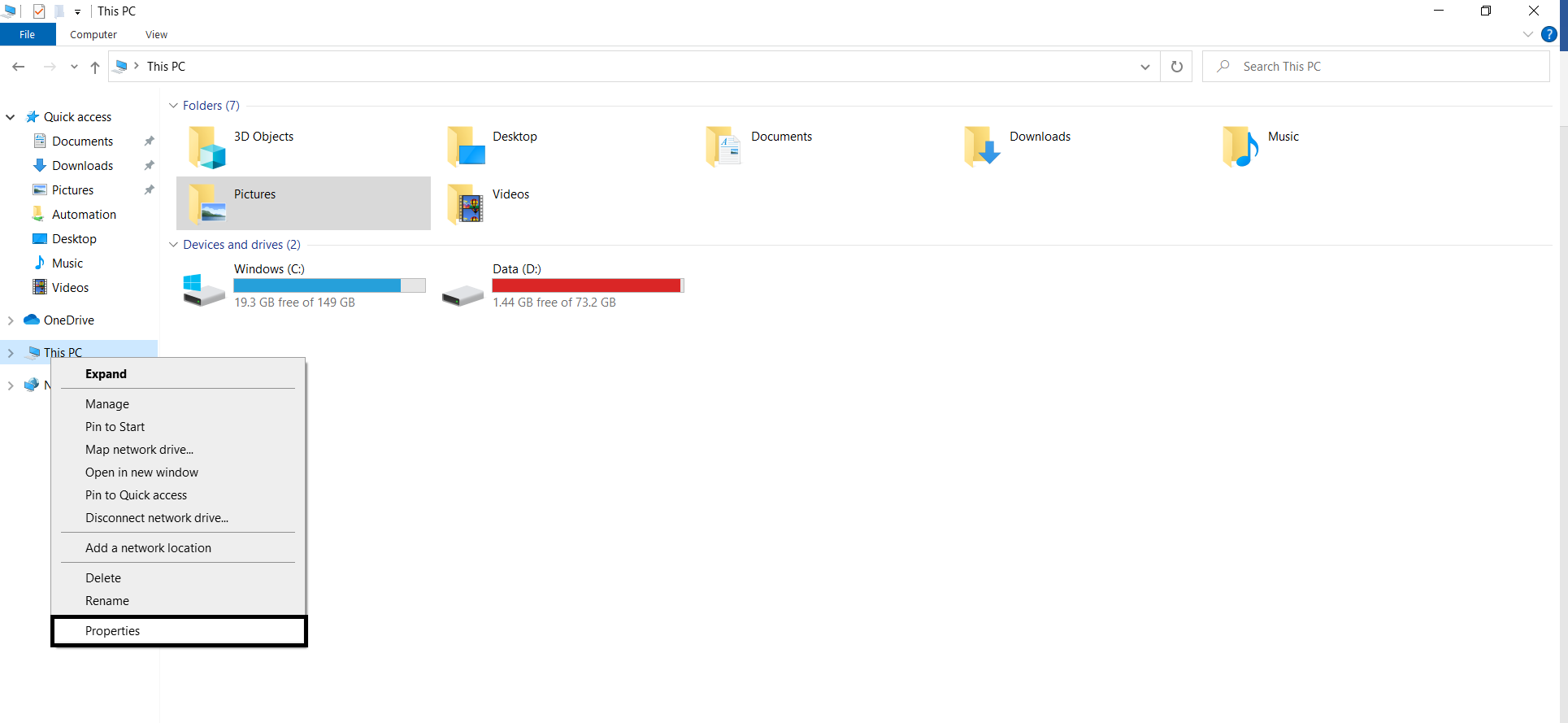
Selenium installation is a 3 step process:

1. **Install Java JDK**

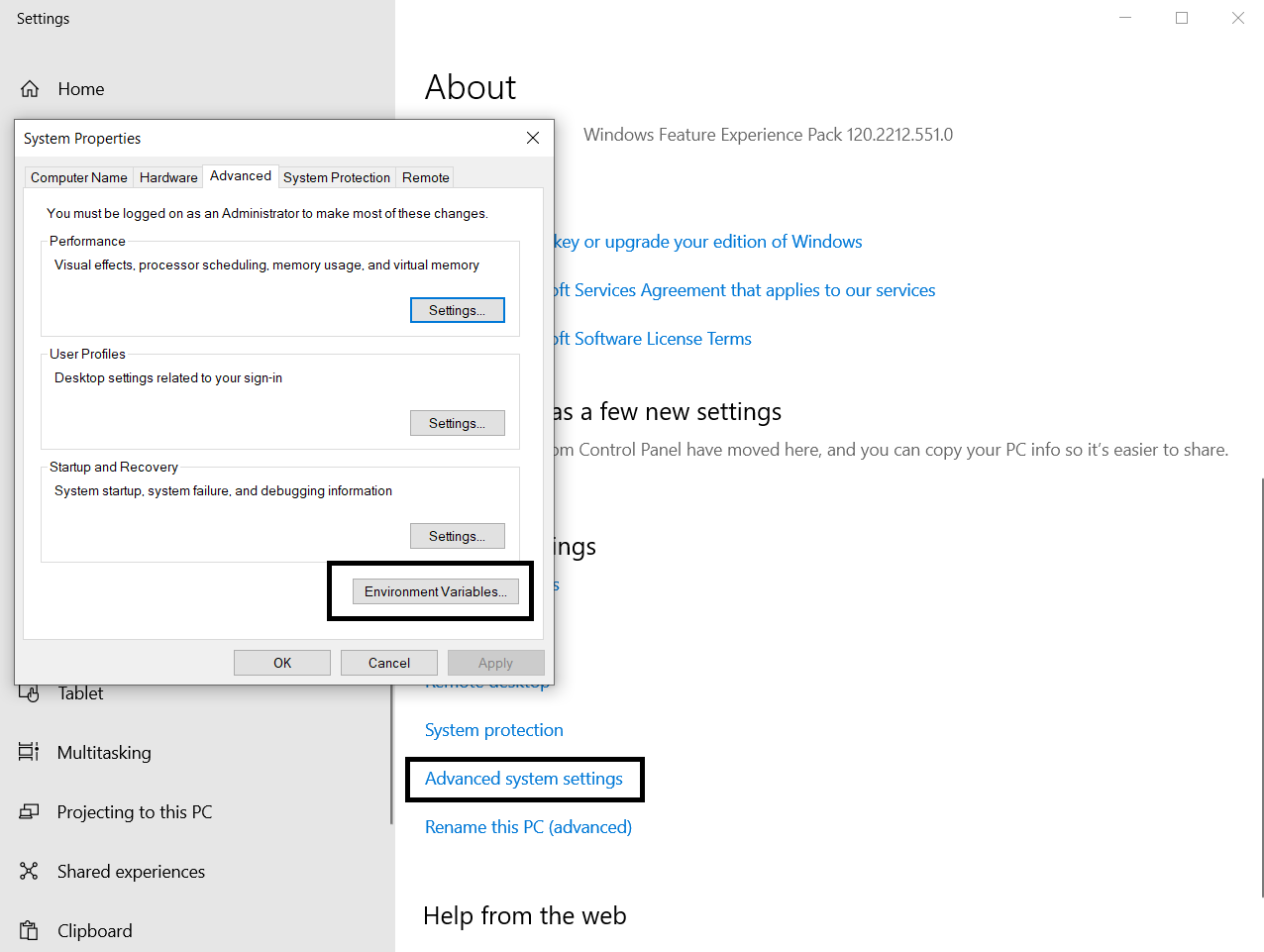
* Go to chrome and search for JDK download.
* Click on the first link <https://www.oracle.com/java/technologies/downloads/>
* Scroll down and select Windows and download file.



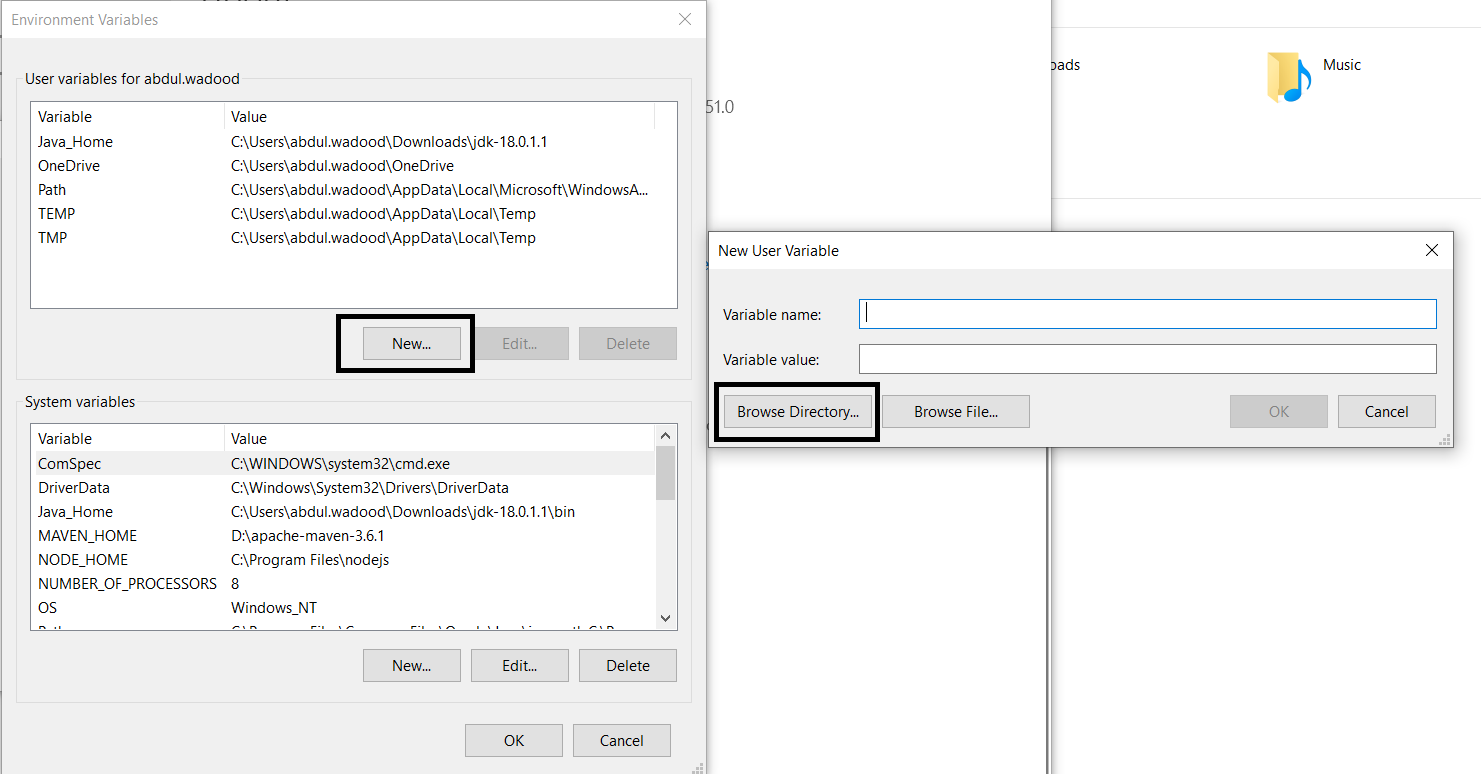
* After downloading JDK set Environment Variables.
* Go to This PC , right click on it and go to properties.



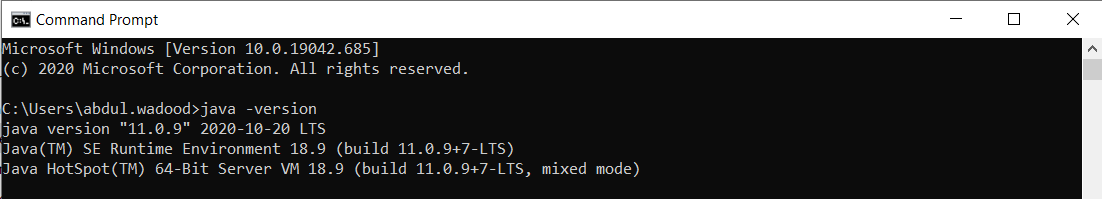
* Select **Advance System Setting** and then select **Environment Variables**.



* Click on New and then on Browse Directory where your JDK is available and use variable name as **“Java\_Home”**.

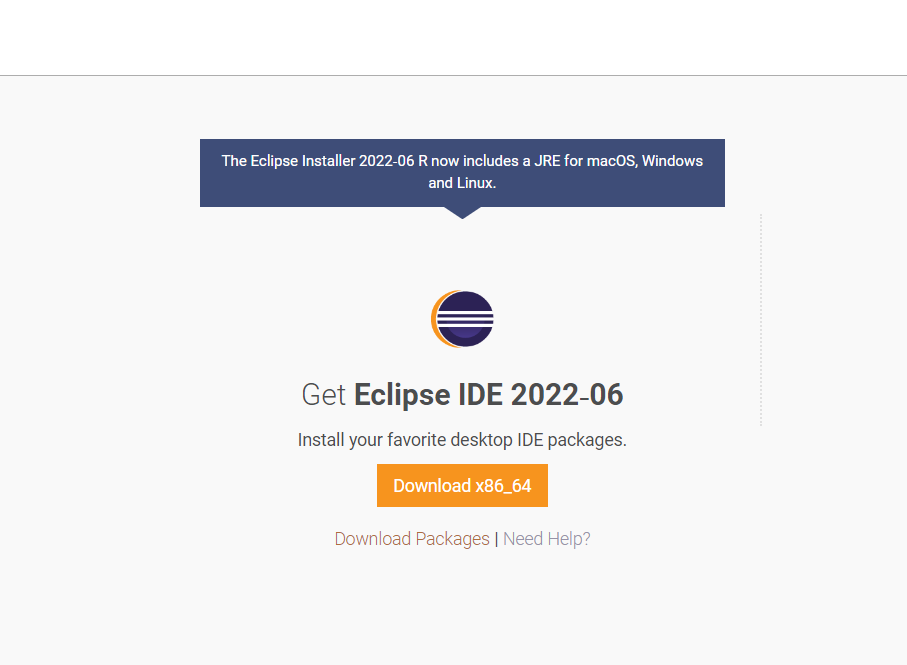


* Follow the above steps again for the System Variables.
* Once the installation is done go to “**cmd**” and write **java -version** to confirm the installation.



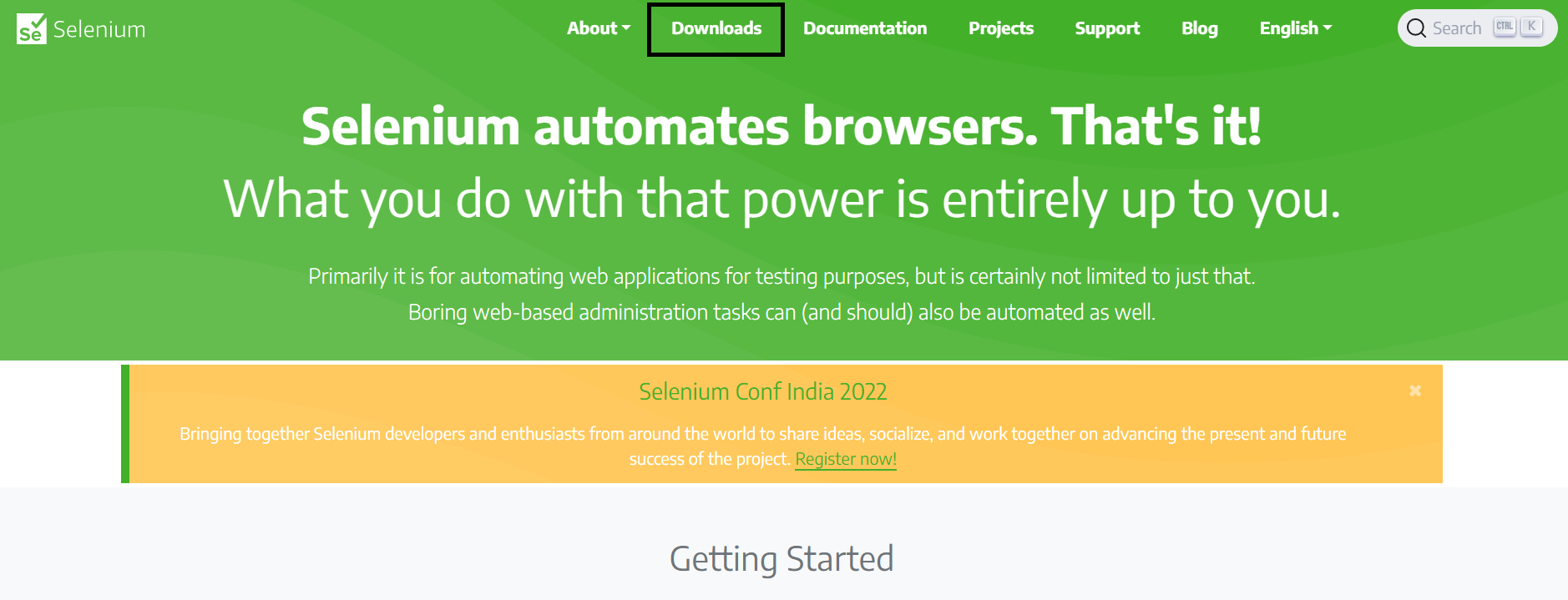
1. **Install Eclipse**

* Go to eclipse website. <https://www.eclipse.org/downloads/>
* Scroll down and download the Eclipse IDE.

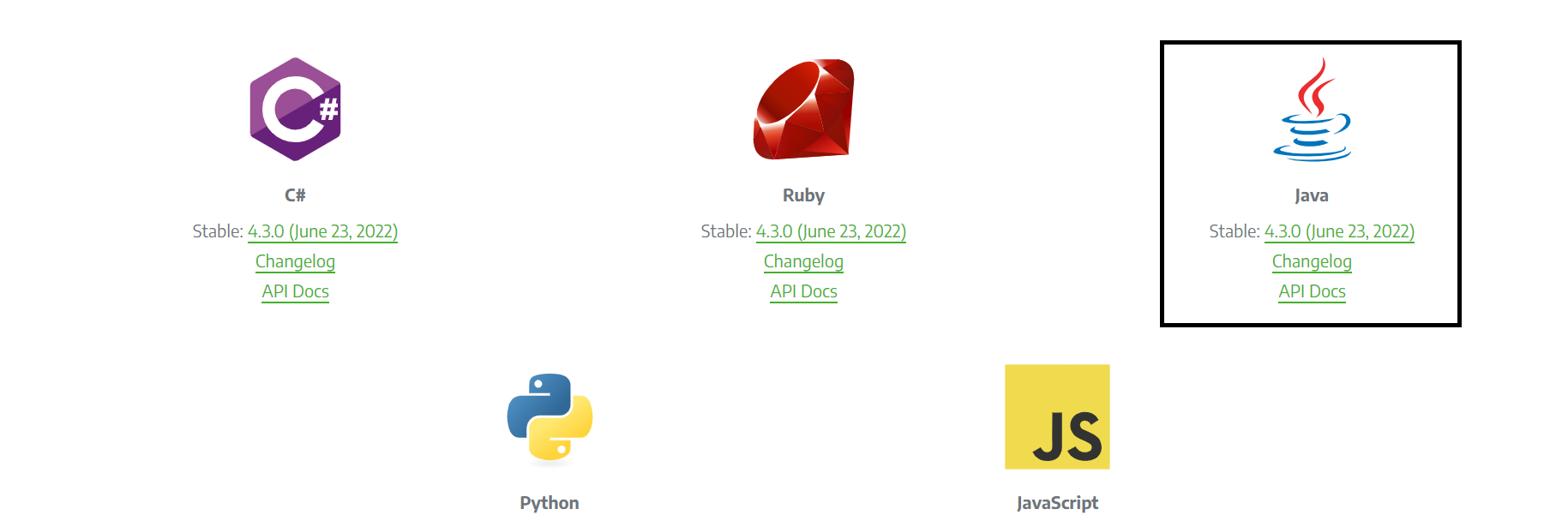


1. **Install Selenium Web driver Files**

* Go to Selenium official website. <https://www.selenium.dev/>
* Go to downloads on the site.



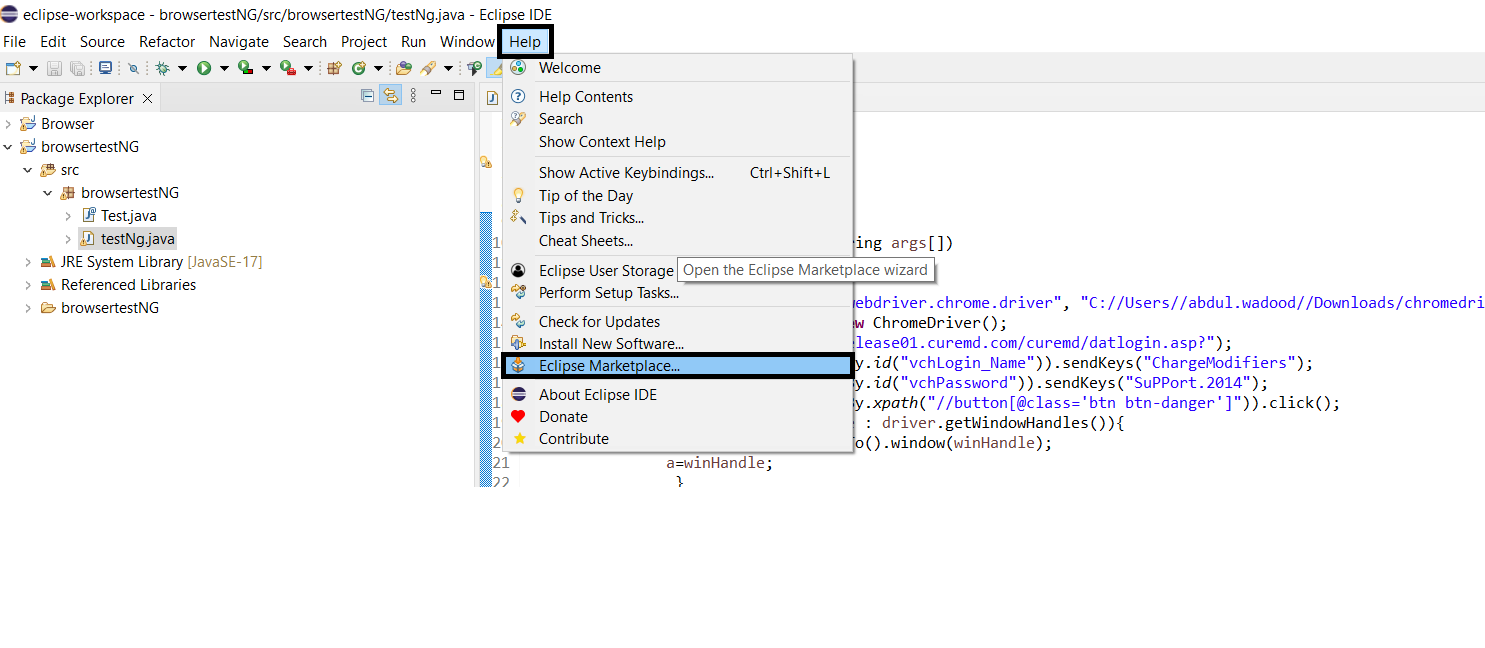
* Scroll down the window and select the Java and click on the link the downloading will start.



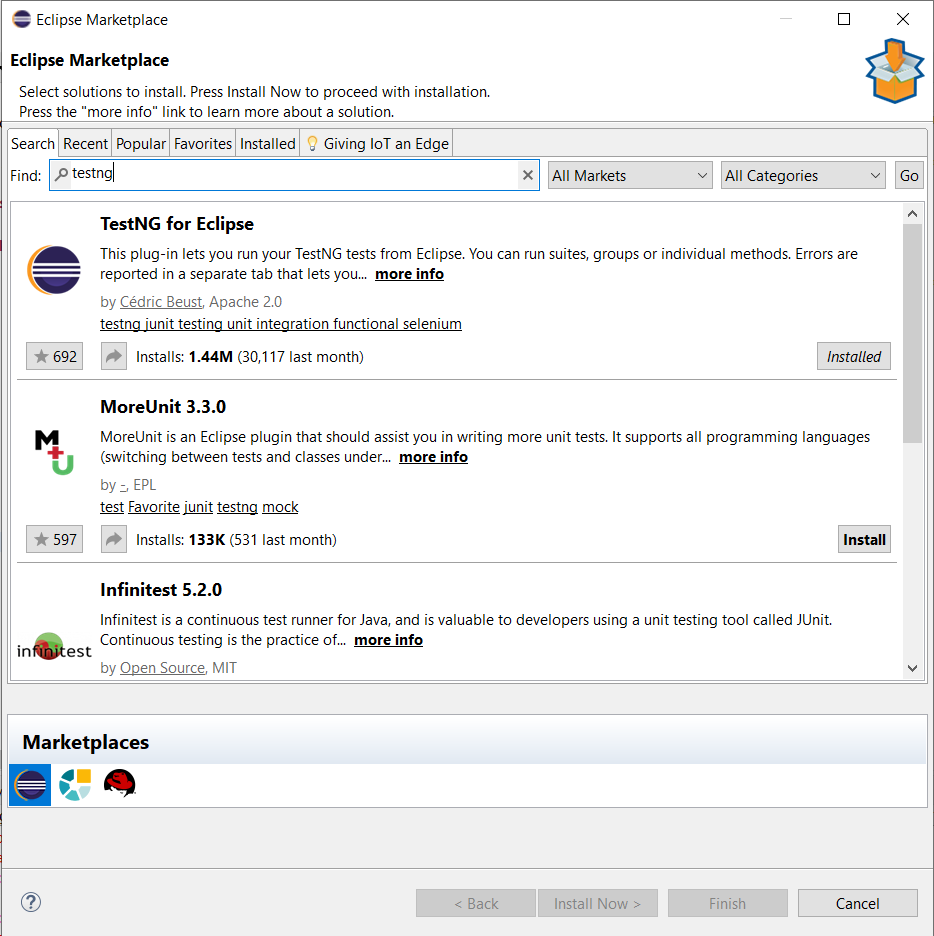
## 

## **TestNG installation**

* Launch the eclipse IDE and go to Help and then to Eclipse Marketplace.

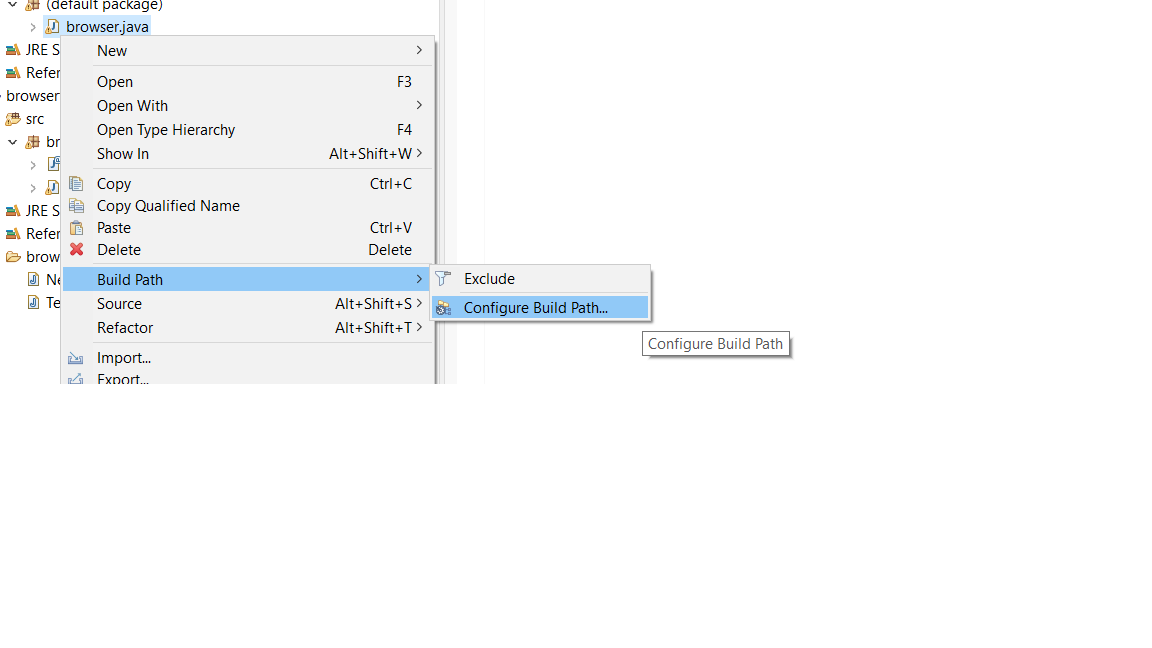


* Write TestNG in the search bar and install it.

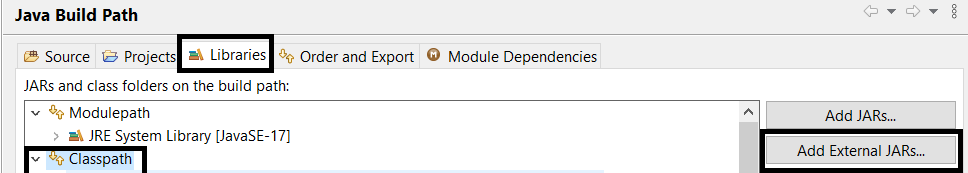


1. **Launching Web Browser in Selenium**

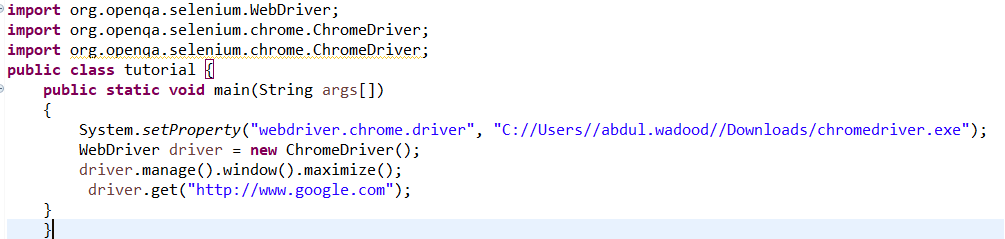
* Launch Eclipse IDE.
* Create New Java Project.
* Right click on src in project and add class.
* After that right click on class and go to **Build Path->Configure Build Path.**



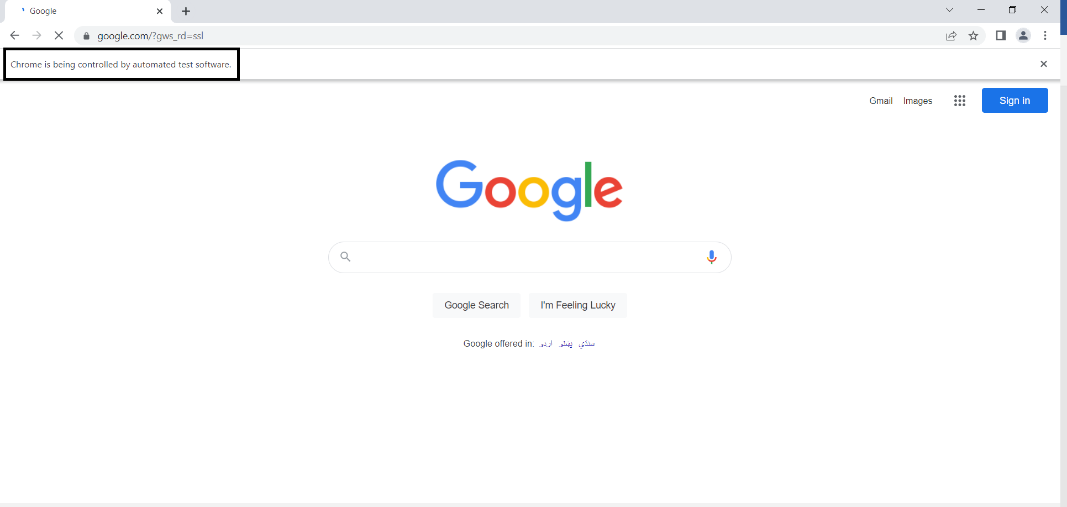
* After that go to **libraries->classpath** and select **Add External Jar files**.



* Add all the files available in the Selenium folder and click **Apply and Close**.
* Download chrome exe depending upon your chrome version.
* Set property by defining the path of chrome exe and run the script.



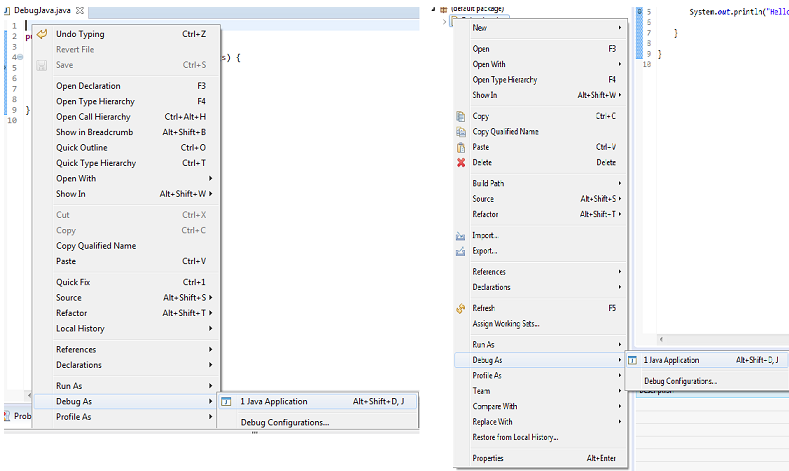
* The script would open the Chrome browser.

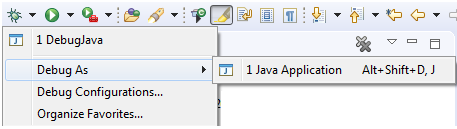


## **Debugging In Eclipse**

### **1. Launching and Debugging a Java program**

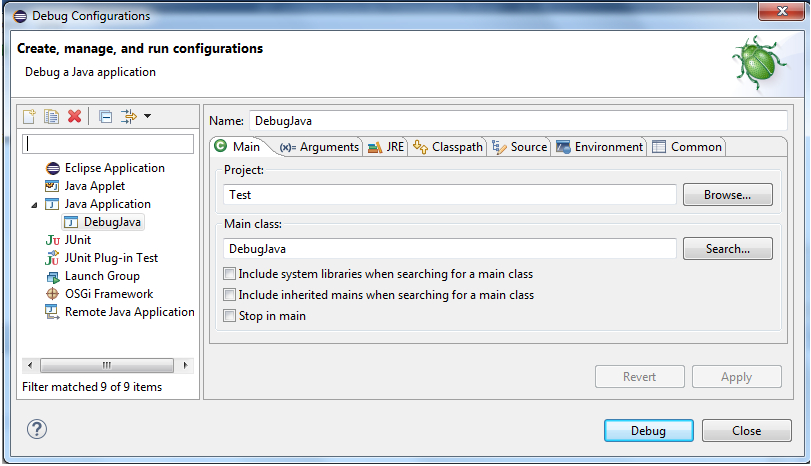
A Java program can be debugged simply by right clicking on the Java editor class file from Package explorer. Select **Debug As → Java Application** or use the shortcut **Alt + Shift + D, J** instead.





Either actions mentioned above creates a new **Debug Launch Configuration** and uses it to start the Java application.

Either actions mentioned above creates a new **Debug Launch Configuration** and uses it to start the Java application.

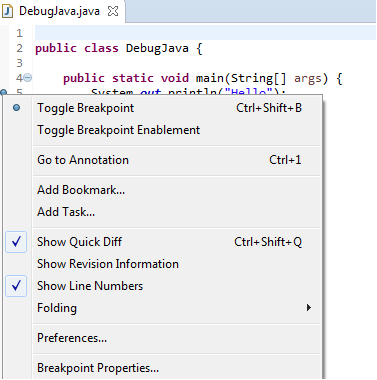


In most cases, users can edit and save the code while debugging without restarting the program.This works with the support of **HCR** (Hot Code Replacement), which has been specifically added as a standard Java technique to facilitate experimental development and to foster iterative trial-and-error coding.

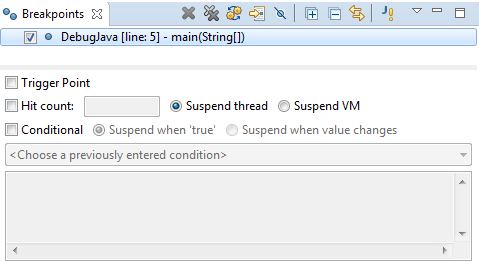
### **2. Breakpoints**

A breakpoint is a signal that tells the debugger to temporarily suspend execution of your program at a certain point in the code.

To define a breakpoint in your source code, right-click in the left margin in the Java editor and select *Toggle Breakpoint*. Alternatively, you can double-click on this position.



The *Breakpoints* view allows you to delete and deactivate Breakpoints and modify their properties.



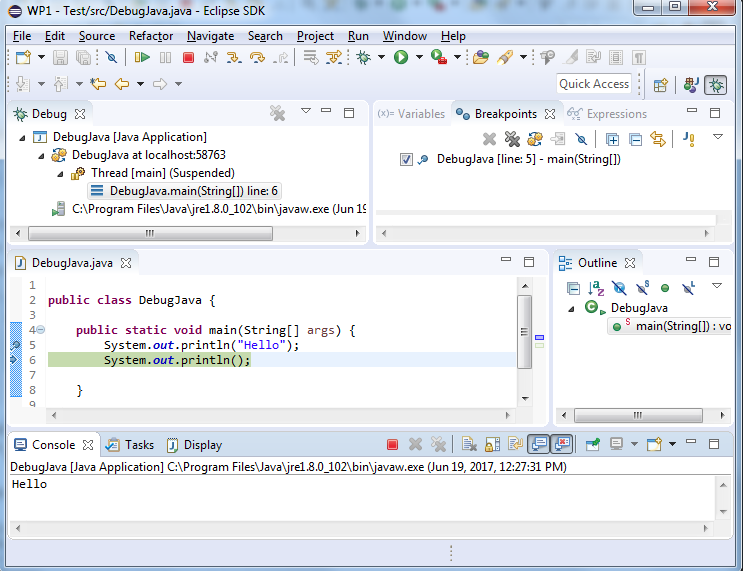
https://www.eclipse.org/community/eclipse_newsletter/2017/june/images/Breakpoints-3.PNG

All breakpoints can be enabled/disabled using **Skip All Breakpoints**. Breakpoints can also be imported/exported to and from a workspace.

### **3. Debug Perspective**

The debug perspective offers additional views that can be used to troubleshoot an application like Breakpoints, Variables, Debug, Console etc. When a Java program is started in the debug mode, users are prompted to switch to the debug perspective.

* **Debug view** – Visualizes call stack and provides operations on that.
* **Breakpoints view** – Shows all the breakpoints.
* **Variables/Expression view** – Shows the declared variables and their values. Press **Ctrl+Shift+d** or **Ctrl+Shift+i** on a selected variable or expression to show its value. You can also add a permanent watch on an expression/variable that will then be shown in the *Expressions view* when debugging is on.
* **Display view** – Allows to Inspect the value of a variable, expression or selected text during debugging.
* **Console view** – Program output is shown here.



### **4.Stepping commands**

The Eclipse Platform helps developers debug by providing buttons in the toolbar and key binding shortcuts to control program execution.

