

## Session 1.4

# INTRODUCTION To Selenium

AN INITIATIVE BY

**UNICAL ACADEMY**

## What will be covered in this session?

- Selenium brief history and key modules
- Selenium architecture
- Environment Setup
- Simple first automation code in Java

### Part 2: Concepts of Testing (30 h)

- Types of Testing
- Common Testing Tools
- Manual Testing – Test cases, Data, Scenarios etc.
- Case-studies and Scenarios

### Part 3: Automated Testing (50 h)

- Selenium Overview
- Web Driver, Locators, Elements, and more
- Automation and Runs

### Part 4: Hands-on Sessions (100 h)

- Use cases and traceability
- Test data and scenarios
- Day to day work
- Defect prevention, RCA and other value add aspects

### Part 1: The Basics (20 hours)

- Organization & its working
- SDLC & STLC Overview
- Basics of OOPS, Database
- Java essentials for Testing
- Overview on few Testing roles Job Descriptions



Let's go!!!



# Selenium automates browsers. That's it!

## What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that.  
Boring web-based administration tasks can (and should) also be automated as well.

Language specific scripts to drive  
a browser based automated  
testing



Selenium WebDriver

To record, edit, and replay test  
scripts for Functional Testing

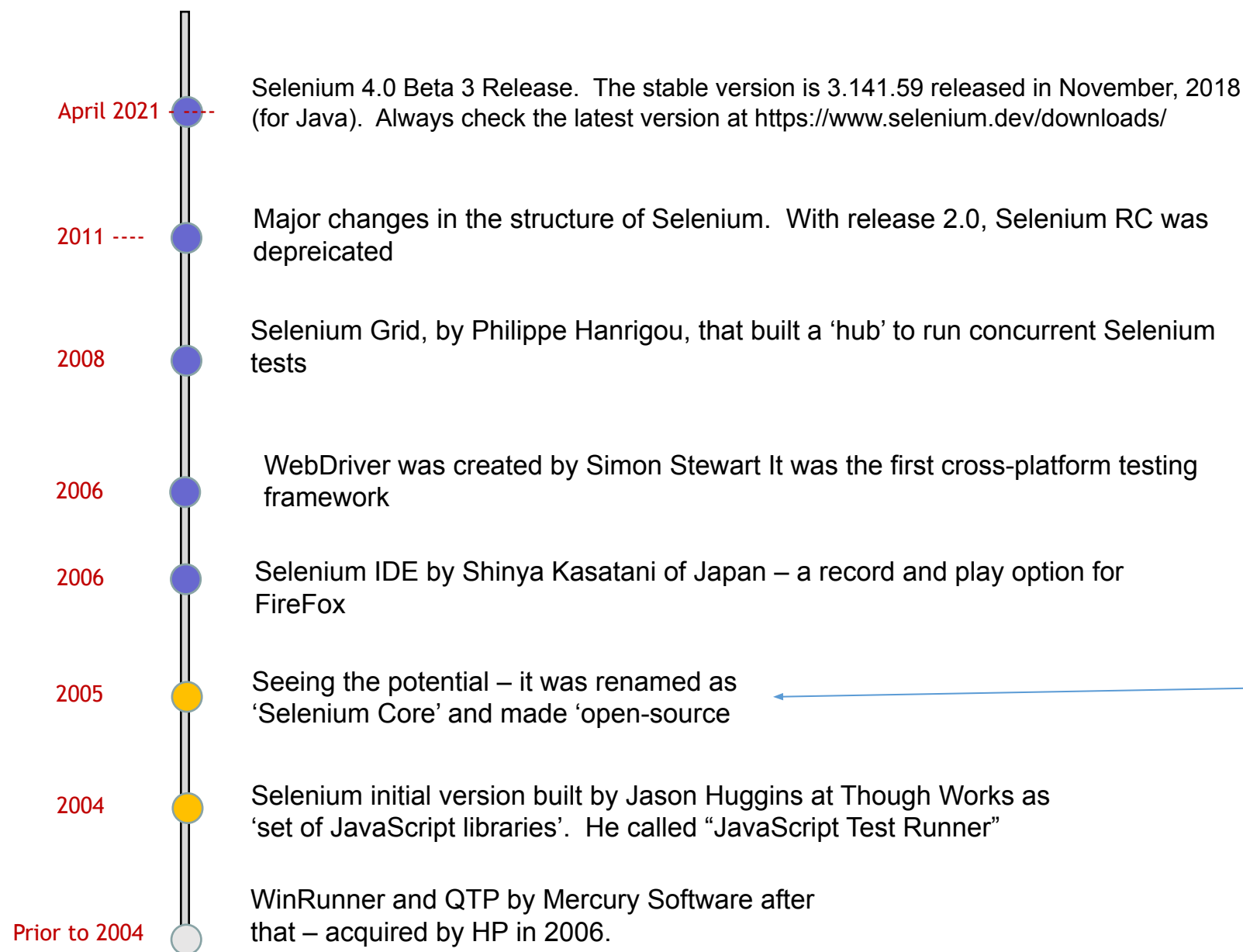


Selenium IDE

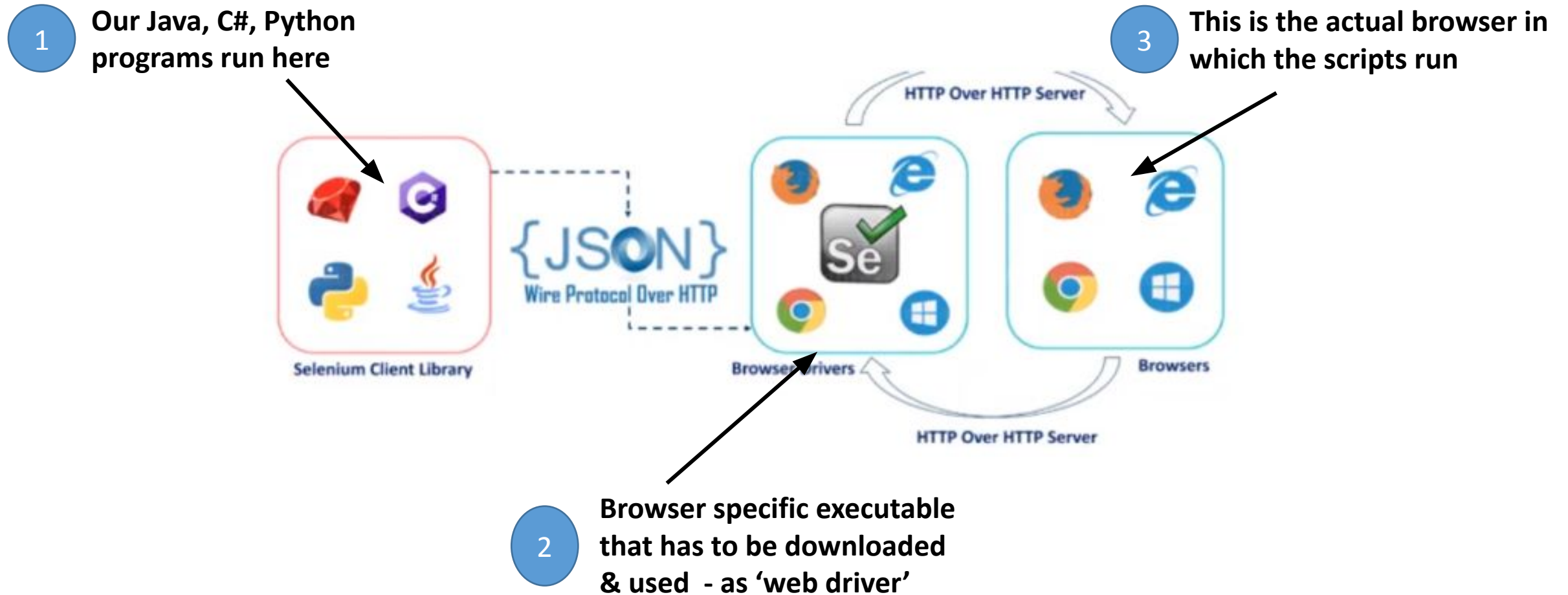
Grid a 'gird' – to run testing on  
multiple systems / multiple  
environments



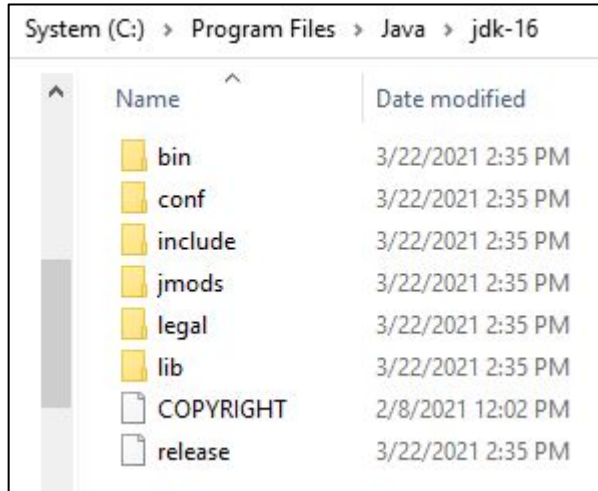
Selenium Grid



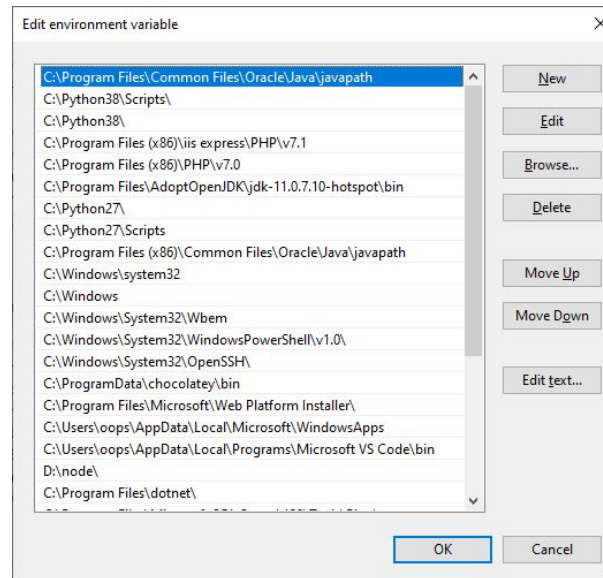
An internal joke says - 'Selenium' as antidote to 'Mercury'



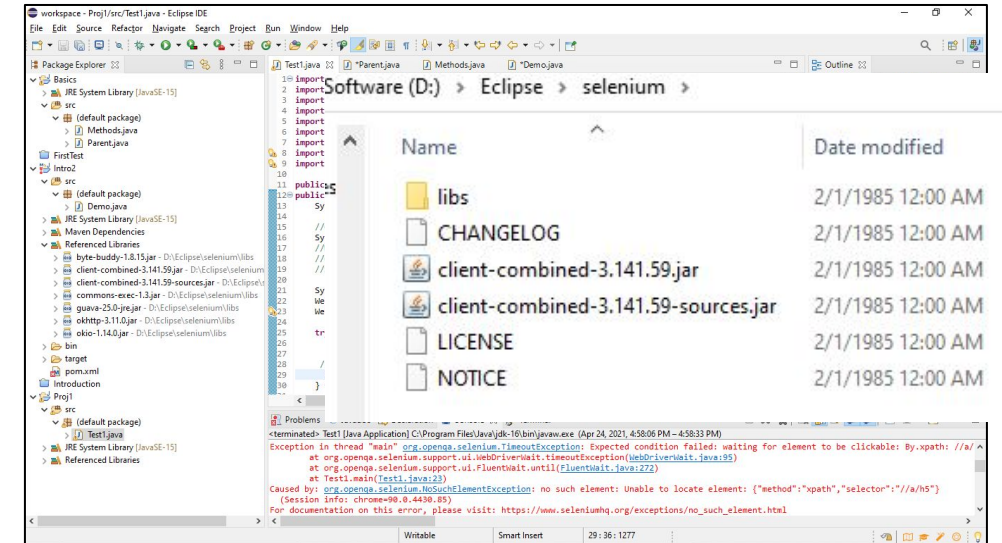




### 1. Step 1: C:\Program Files\Java\jdk-16



### 2. Configure Java Path environment






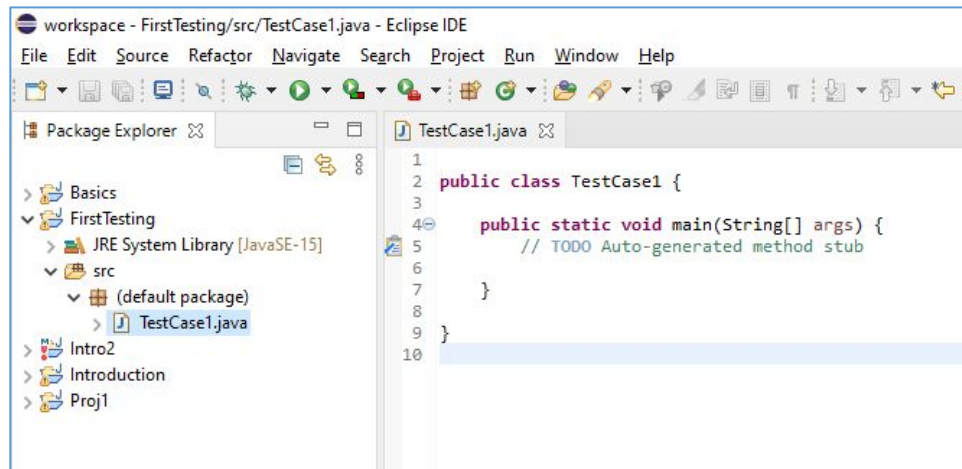
### 3. Install IDE and configure Selenium Jars / Classpath

**After the initial three steps, for every run / test cases –**

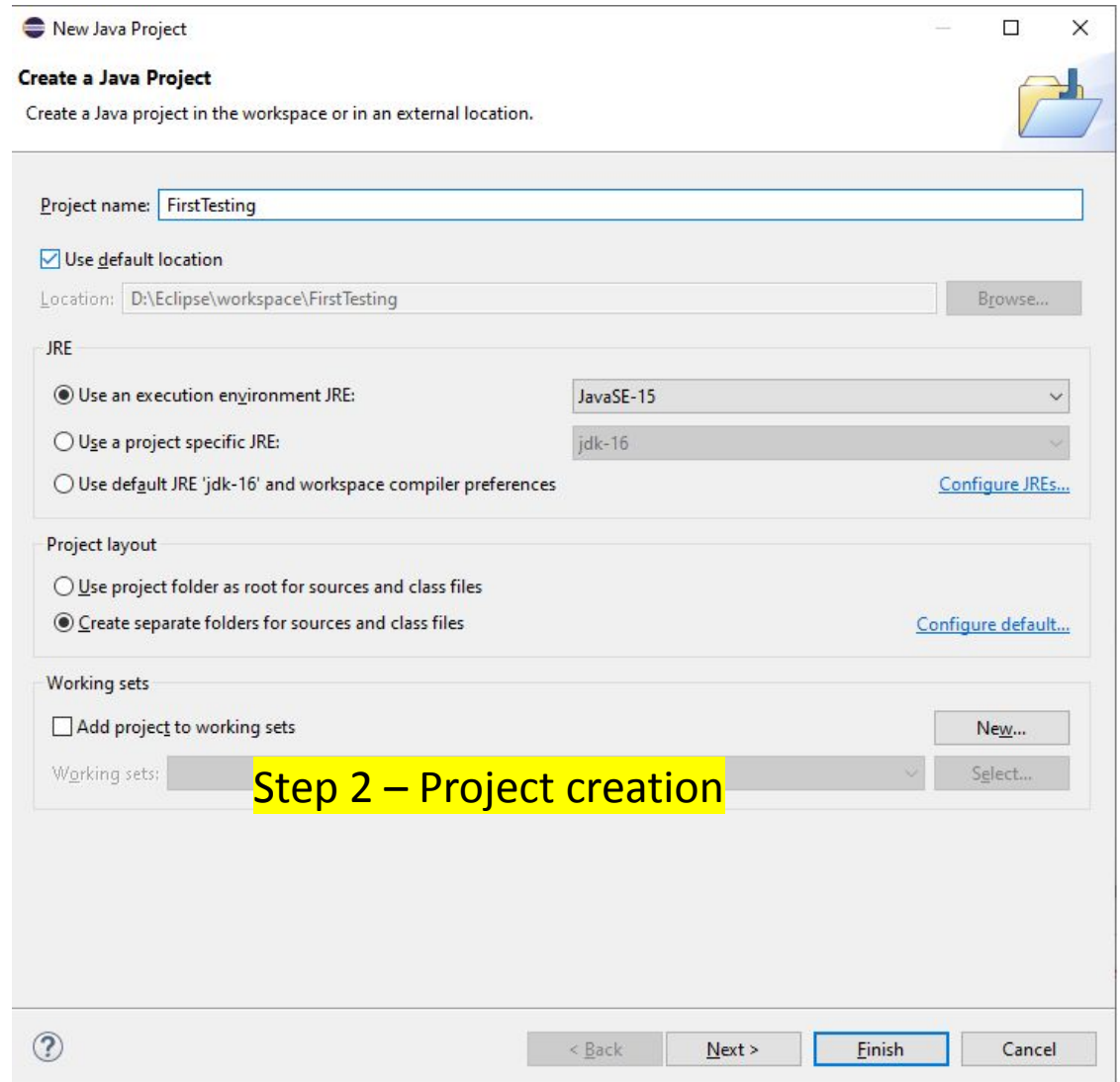
1. Choose a browser to run
2. Create a driver object based on the chosen browser
3. Run the test program
4. Work on the advanced testing – data-driven testing, framework based testing etc.

**That's it – that simple. You are all set to go**

1. Open Eclipse IDE
2. Go to File  New  Java Project  Create 'FirstTesting' project
3. PLEASE NOTE: Do not create 'Module' – even if Eclipse prompts.
4. Right click the 'src' folder and create first class i.e. your first test case (to open a browser and to read the webpage title)

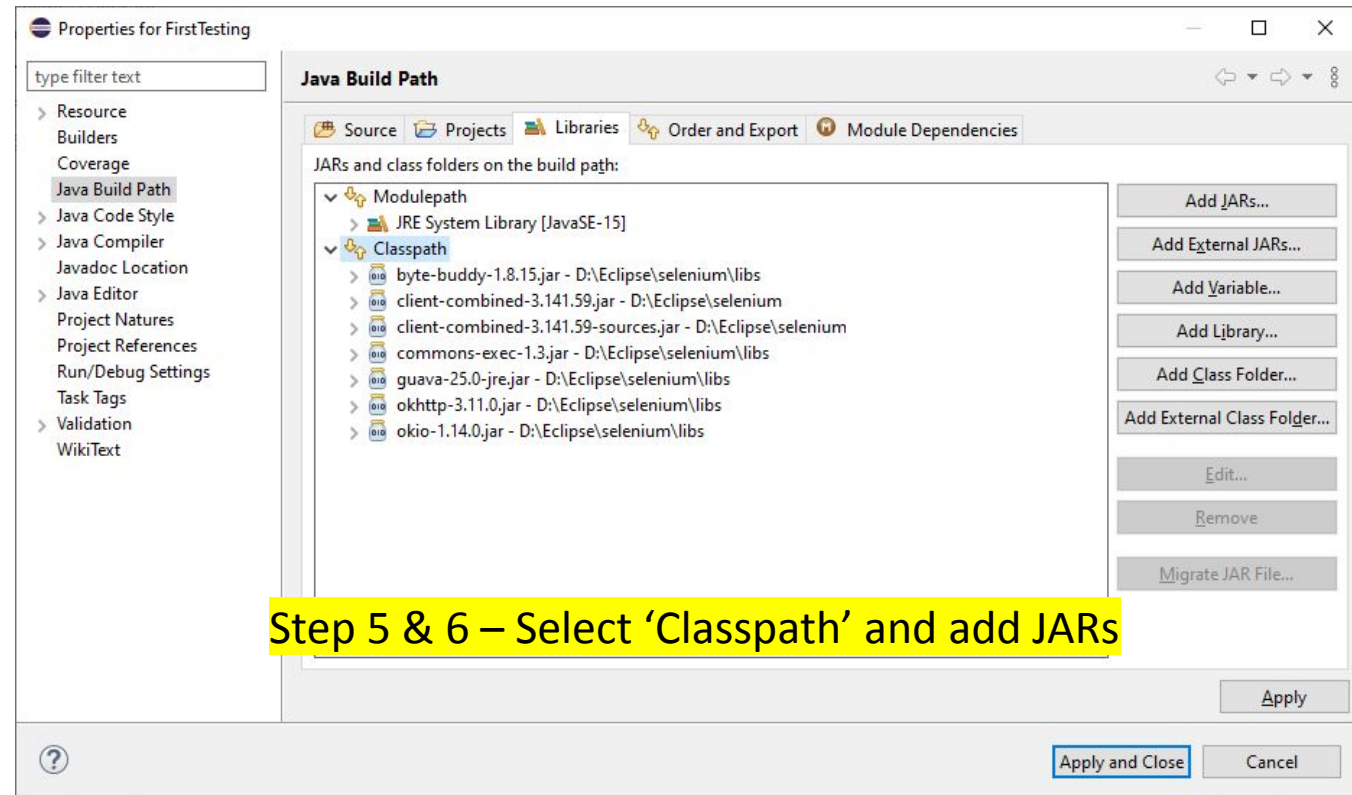


Step 4 – 'Blank class' file



Step 2 – Project creation

5. Right click Project (“FirstTesting”)  
□ Properties □ Java Build Path □  
Then go to Libraries and select  
‘Classpath’.
6. Select ‘Add External JARs’ and  
click ‘Apply and Close’





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

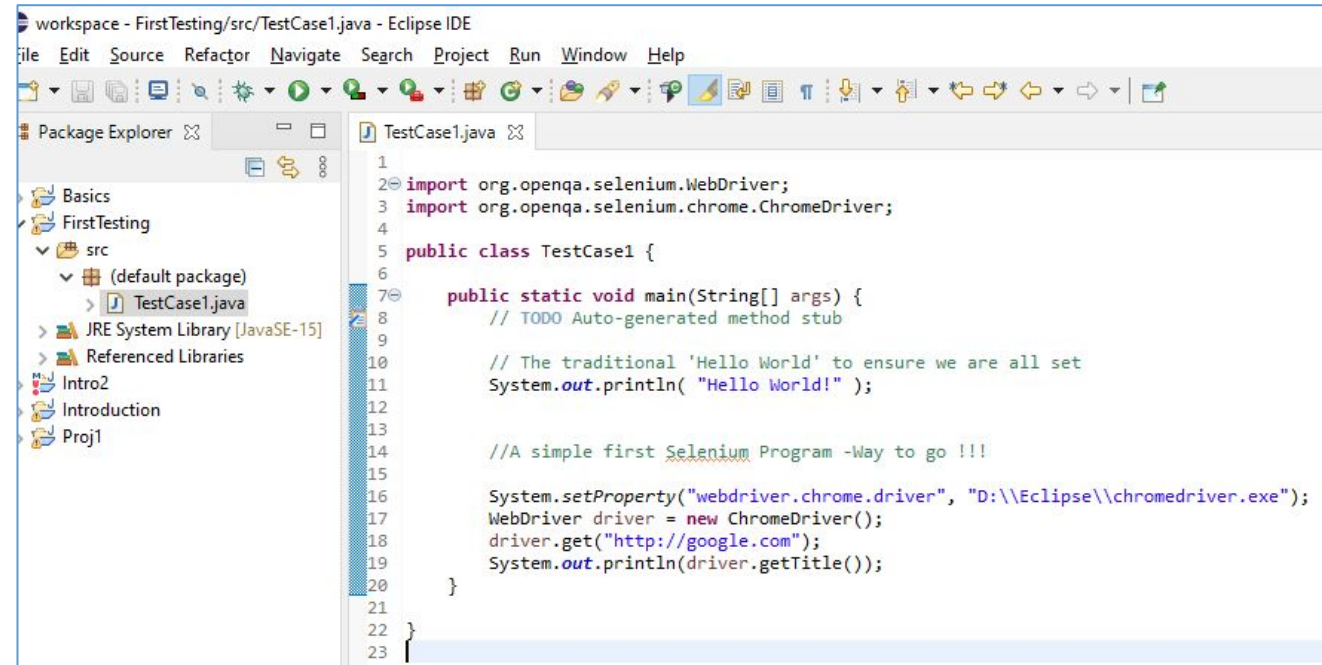
public class TestCase1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        // The traditional 'Hello World'
        System.out.println( "Hello World!" );

        //A simple first Selenium Program -Way to go !!!

        System.setProperty("webdriver.chrome.driver",
            "D:\\Eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://google.com");
        System.out.println(driver.getTitle());
    }
}
```



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

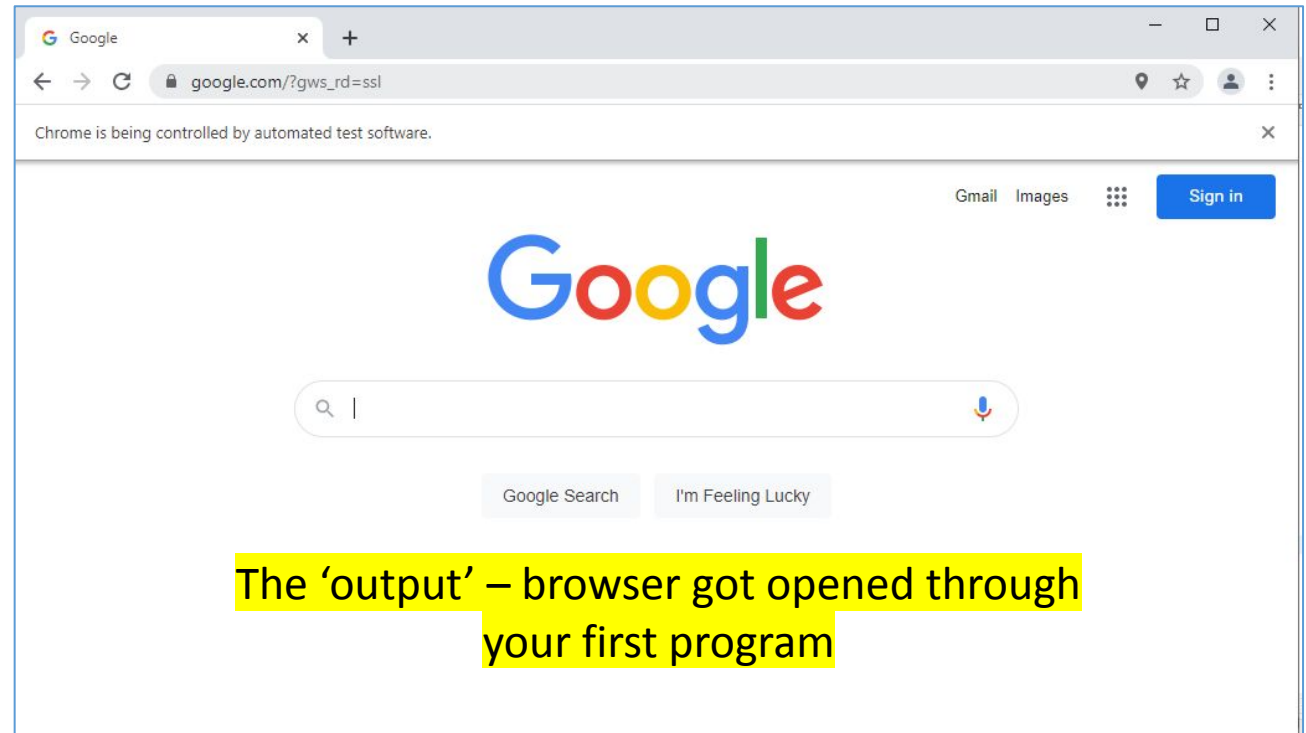
public class TestCase1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        // The traditional 'Hello World'
        System.out.println( "Hello World!" );

        //A simple first Selenium Program -Way to go !!!

        System.setProperty("webdriver.chrome.driver",
            "D:\\Eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://google.com");
        System.out.println(driver.getTitle());
    }
}
```



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

public class TestCase1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        // The traditional 'Hello World'
        System.out.println( "Hello World!" );

        //A simple first Selenium Program -Way to go !!!

        System.setProperty("webdriver.chrome.driver",
            "D:\\Eclipse\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://google.com");
        System.out.println(driver.getTitle());
    }
}
```

'WebDriver' is a critical Interface (**not a class**) throughout our Selenium learning. A reference to that interface is added here.

This particular program is trying to open a Google Chrome driver. Hence, this ensures a reference to Google Chrome class (**not an interface**).

This is the main code. We will get more insight on this snippet during our training. But, below are the key points for now—

- First, we are informing the Selenium Jars that we will be using 'Chrome driver'.
- Then, we are creating an instance of WebDriver interface ("driver"). But, by casting it to ChromeDriver class.
- Once WebDriver instance is created (we called "driver" in this example, we can invoke various methods of it.
  - Example 1: get – is to 'Open a URL'
  - Example 2: getTitle – is to read 'opened URL's title'

