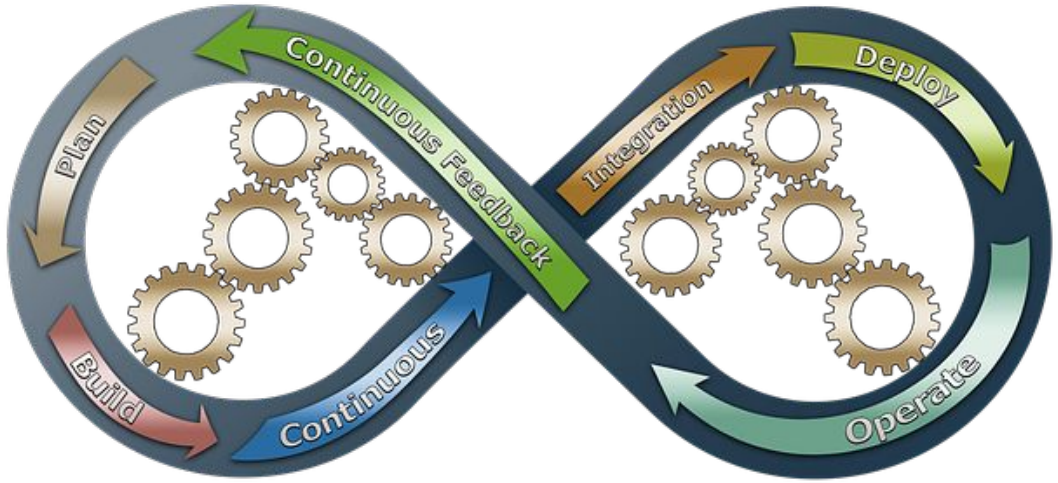


Session 1.4

INTRODUCTION To **Selenium**

AN INITIATIVE BY

UNICAL ACADEMY

Introduction

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 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

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

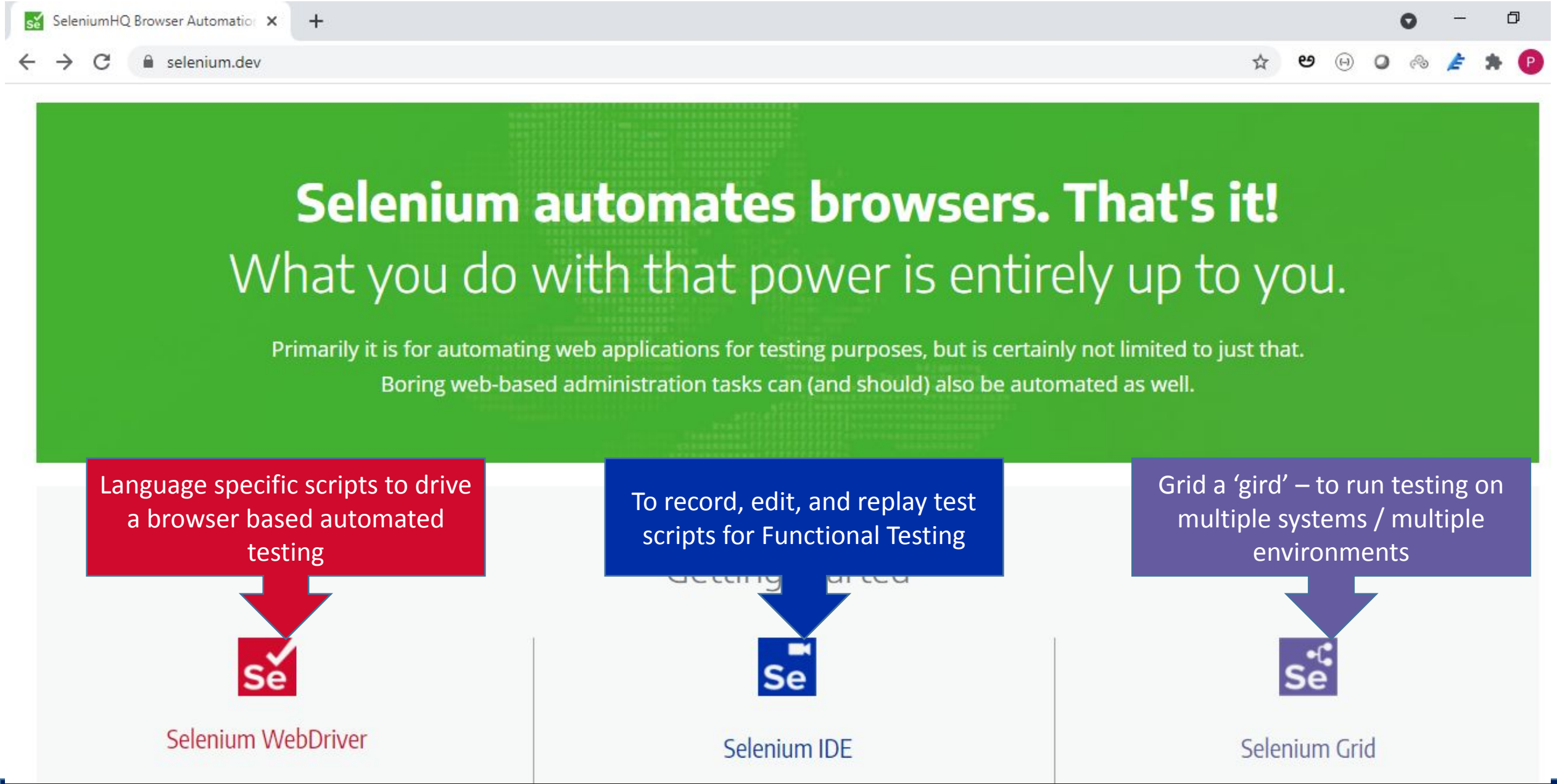
Part 4: Hands-on Sessions (100 h)

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

Let's go!!!

What is Selenium?

UNICAL ACADEMY



Selenium – brief history

UNICAL ACADEMY

April 2021 ----
Selenium 4.0 Beta 3 Release. The stable version is 3.141.59 released in November, 2018 (for Java). Always check the latest version at https://www.selenium.dev/downloads/

2011 ----
Major changes in the structure of Selenium. With release 2.0, Selenium RC was depreicated

2008
Selenium Grid, by Philippe Hanrigou, that built a 'hub' to run concurrent Selenium tests

2006
WebDriver was created by Simon Stewart It was the first cross-platform testing framework

2006
Selenium IDE by Shinya Kasatani of Japan – a record and play option for FireFox

2005
Seeing the potential – it was renamed as 'Selenium Core' and made 'open-source

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

2004
Selenium initial version built by Jason Huggins at Though Works as 'set of JavaScript libraries'. He called "JavaScript Test Runner"

Prior to 2004
WinRunner and QTP by Mercury Software after that – acquired by HP in 2006.

Selenium WebDriver - simplified

UNICAL ACADEMY

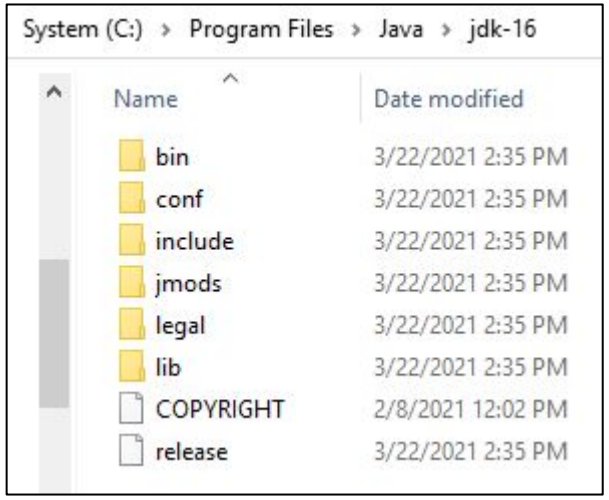
1 **Our Java, C#, Python programs run here**

3 **This is the actual browser in which the scripts run**

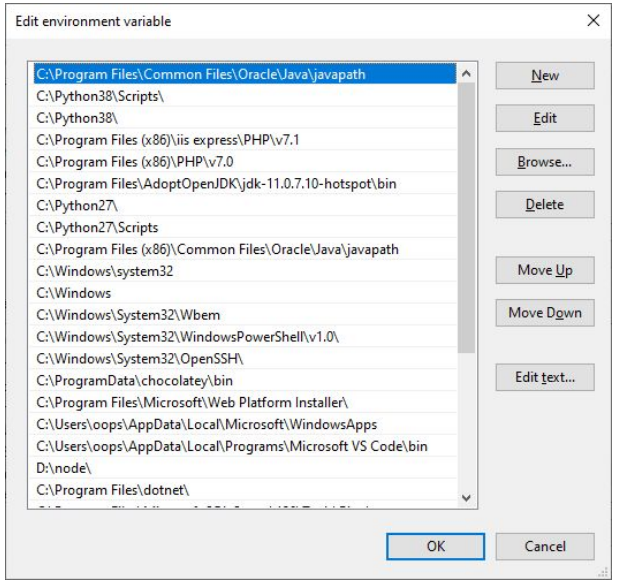
2 **Browser specific executable that has to be downloaded & used - as 'web driver'**

Let's start with Installation

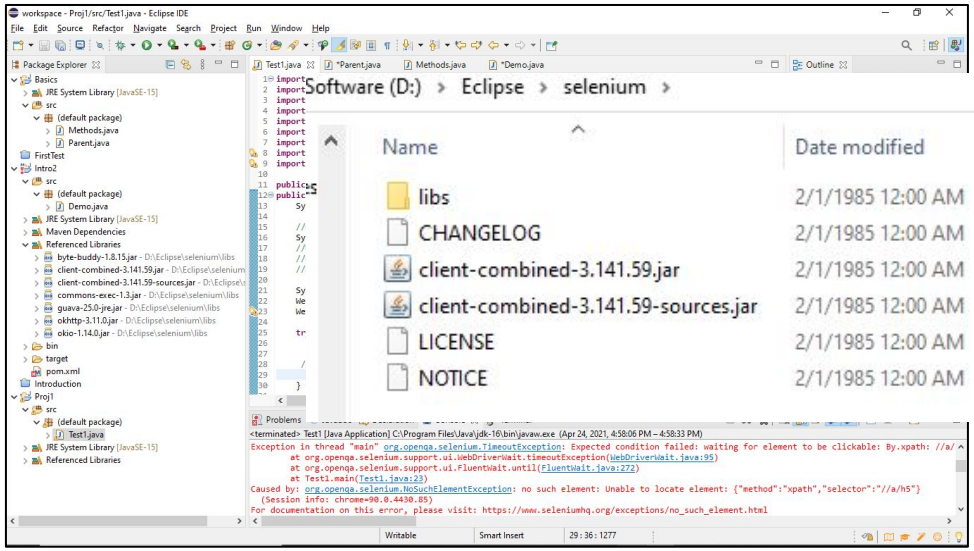
UNICAL ACADEMY



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 –

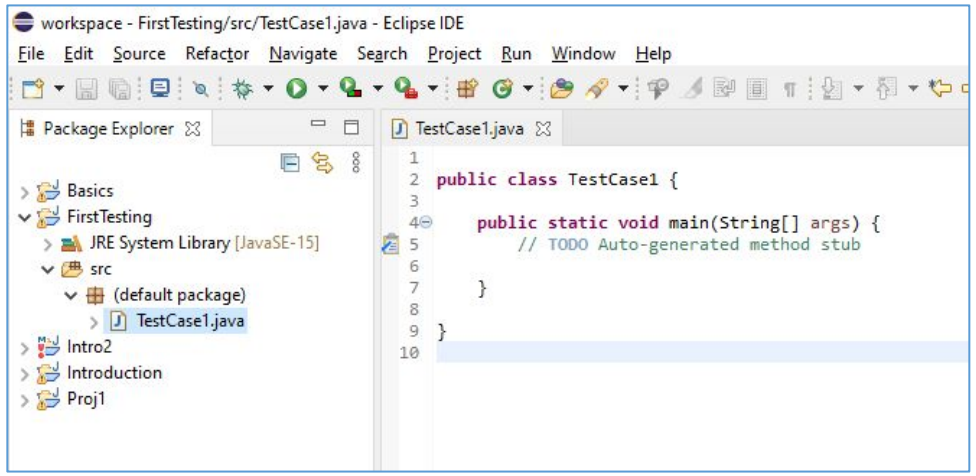
1. Choose a browser to run
2. Create a driver object based on the chosen browser
3. Run the test program
4. Work on the advanced testing – data-driven testing, framework based testing etc.

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

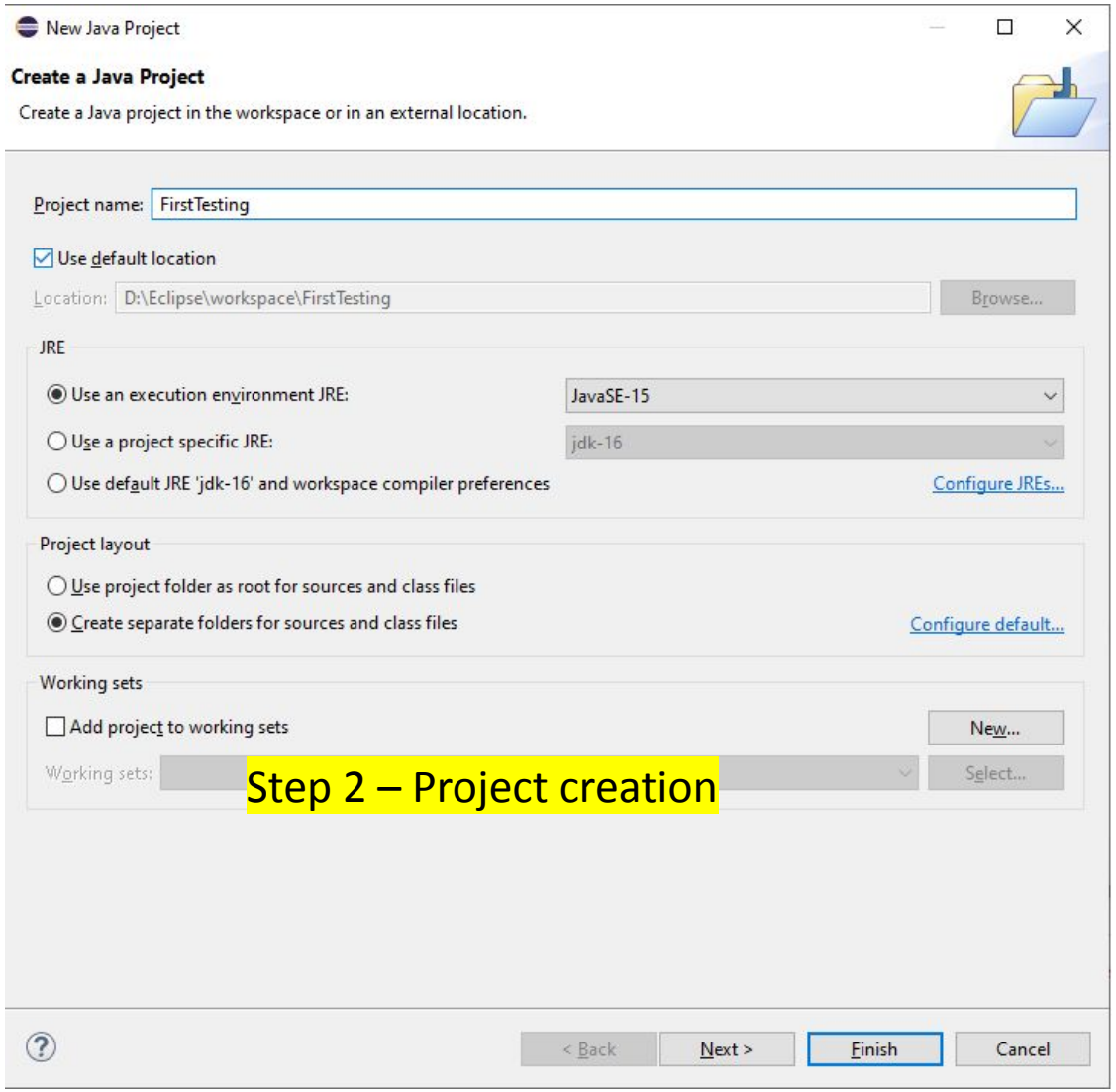
Hay - our first program (1 of 4)

UNICAL ACADEMY

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

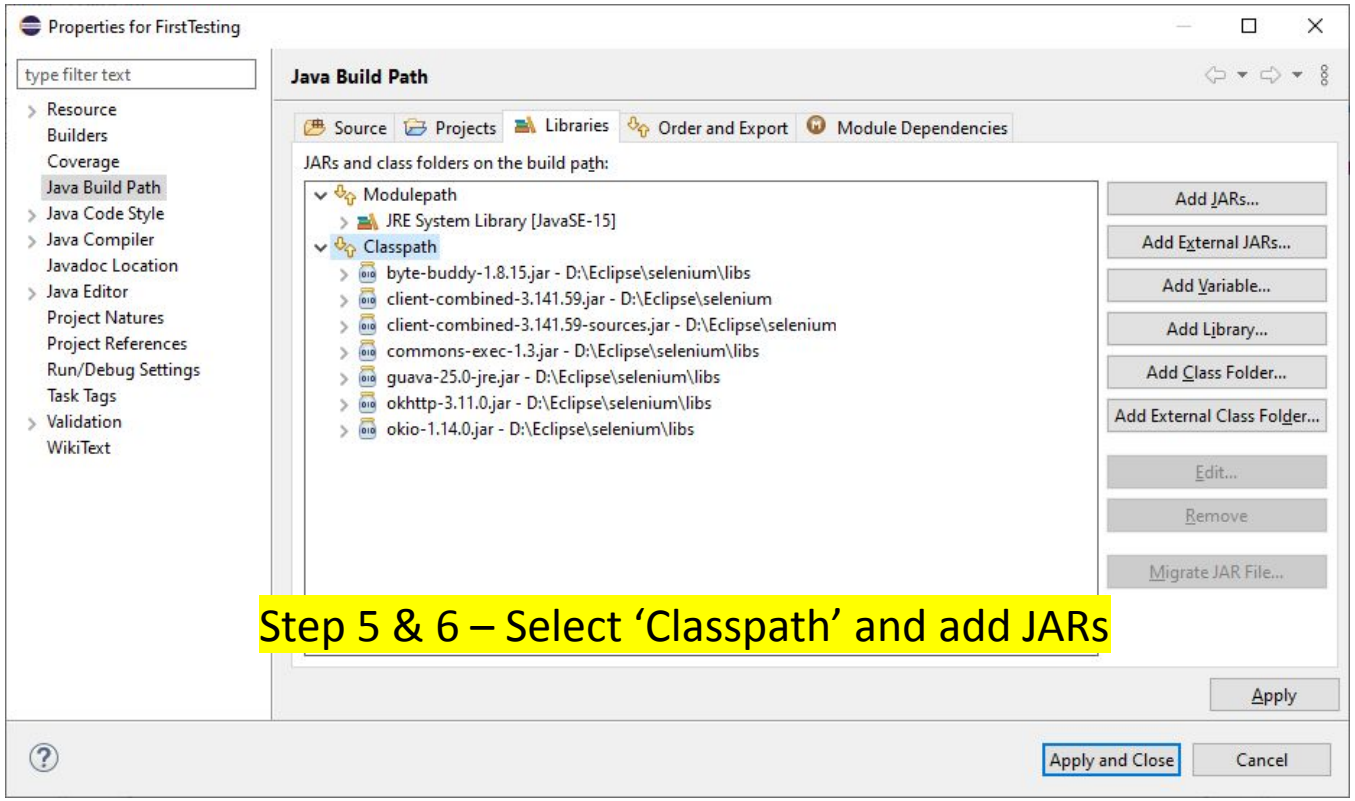


Step 2 – Project creation

Add Selenium libraries (2 of 4)

UNICAL ACADEMY

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’

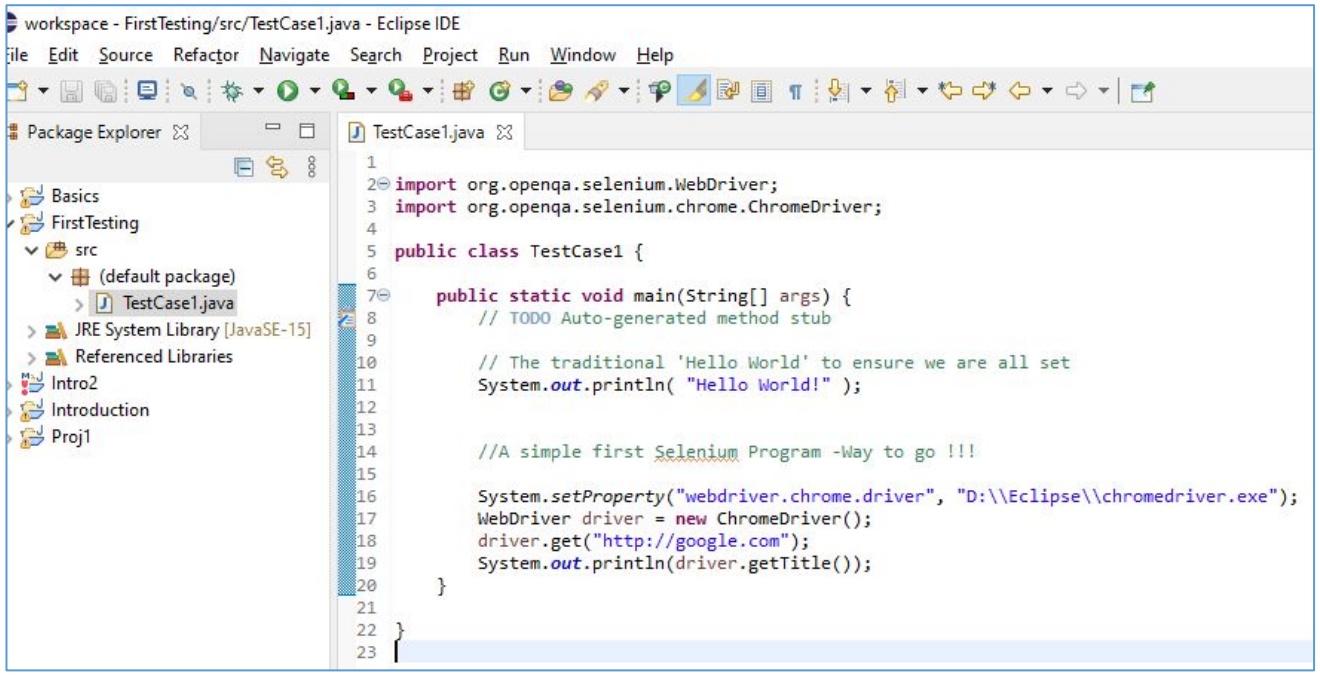


Step 5 & 6 – Select ‘Classpath’ and add JARs

Copy and Paste the Code(3 of 4)

UNICAL ACADEMY

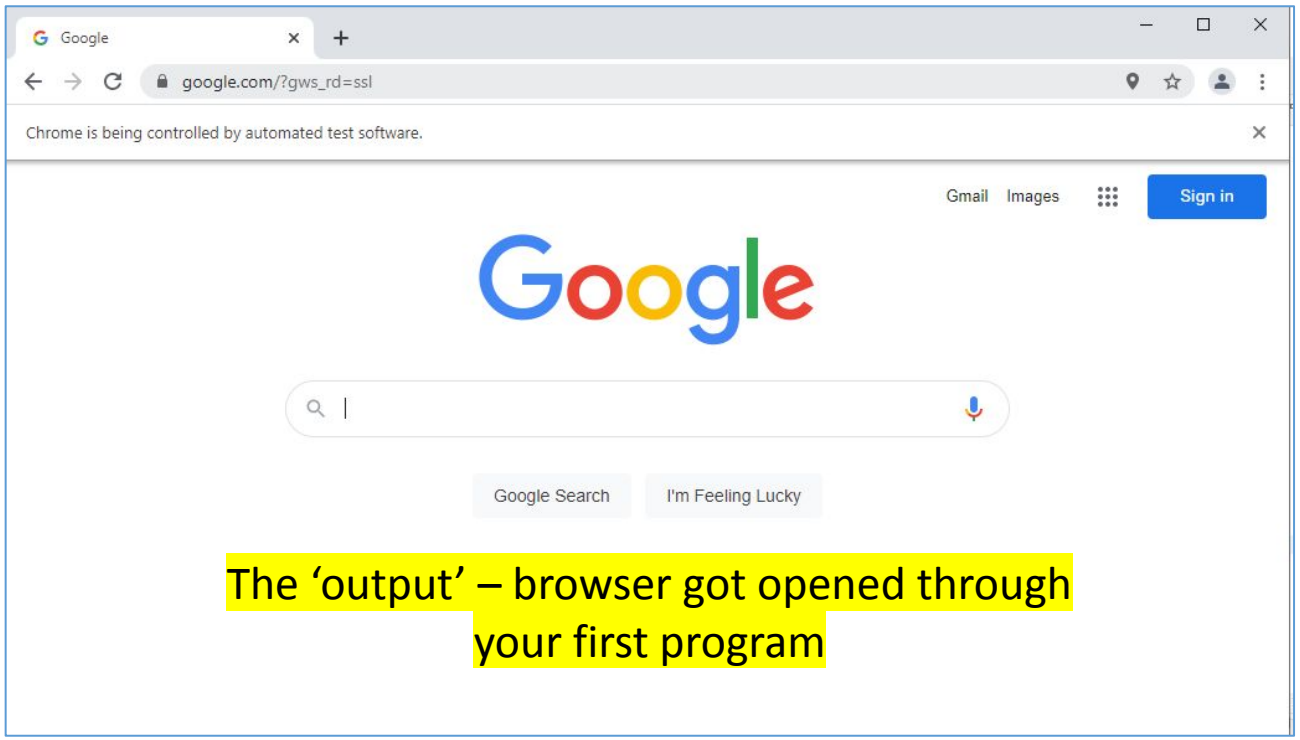
```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

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());
}
}
```



Copy and Paste the Code(4 of 4)

UNICAL ACADEMY

```
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());
}
}
```



The 'output' – browser got opened through your first program

Let's understand our code

UNICAL ACADEMY

```
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'

Session Recap

UNICAL ACADEMY