

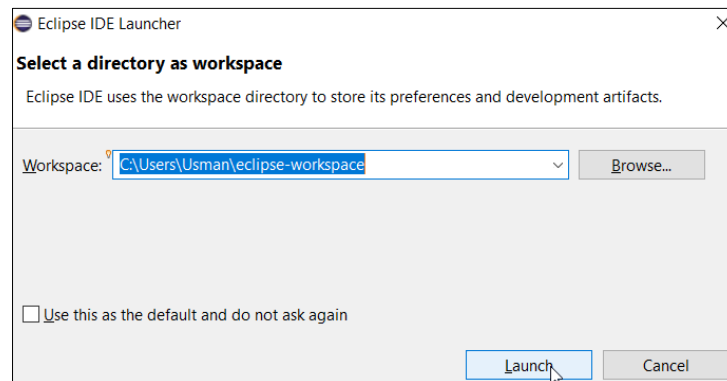
#AIM: Install Selenium Server(Selenium RC) and demonstrate it using a script in Java.

PRE-REQUISITES(*here are w.r.t. Windows 10(64 bit), so choose accordingly w.r.t. your specs):

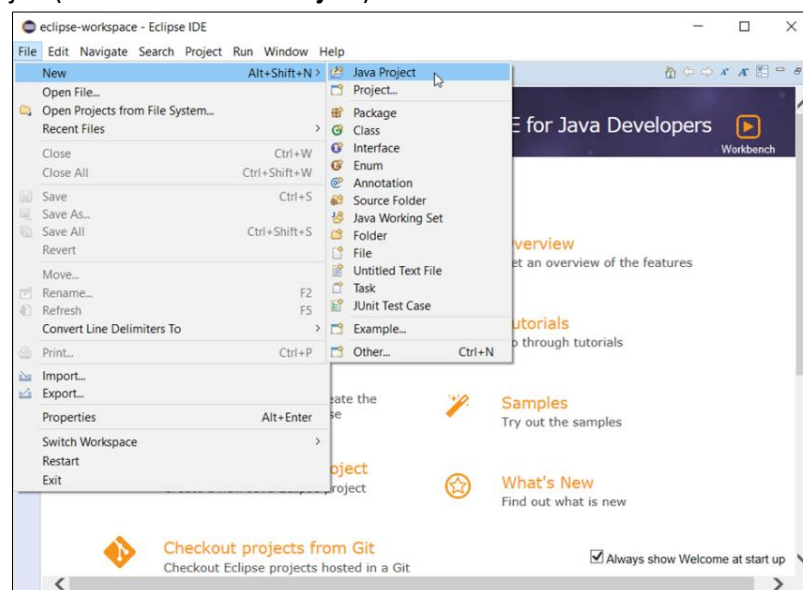
- 1) To Download "JDK":
 - Visit <https://www.oracle.com/technetwork/java/javase/downloads/jdk12-downloads-5295953.html>
 - Download this file "jdk-12.0.2_windows-x64_bin.exe" and install it.
- 2) To Download "Eclipse IDE":
 - Visit <https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2019-06/R/eclipse-inst-win64.exe>
 - Click "Download".
 - Installation:
 - ◆ Open application.(click OK if errors occur like "could not find java.dll" & "could not find Java Runtime Environment SE")
 - ◆ Select "Eclipse IDE for Java Developers".
 - ◆ It will automatically locate the JDK. Choose path, and click "Install".
 - After installation, click "Launch" or open Eclipse from START menu in Windows.
- 3) To Download "Selenium Server Driver and Client Driver(JAR files)":
 - a) For "Selenium Server Driver":
 - Visit <https://www.seleniumhq.org/download/>
 - Under section "Selenium Standalone Server", click download version "3.141.59"
 - You'll get the executable jar file(selenium-server-standalone-3.141.59)
 - b) For "Selenium Client Driver":
 - Visit <https://www.seleniumhq.org/download/>
 - Under section "Selenium Client & WebDriver Language Bindings", download the "3.141.59" version of Java.
 - Extract the file and you'll see two jar files. From them, we'll be using this executable jar file(client-combined-3.141.59)
- 4) To Download "Gecko Driver":
 - Visit <https://github.com/mozilla/geckodriver/releases>
 - Under section "Assets", download "geckodriver-v0.24.0-win64.zip" file.
 - You'll get the application file "geckodriver".

STEPS:

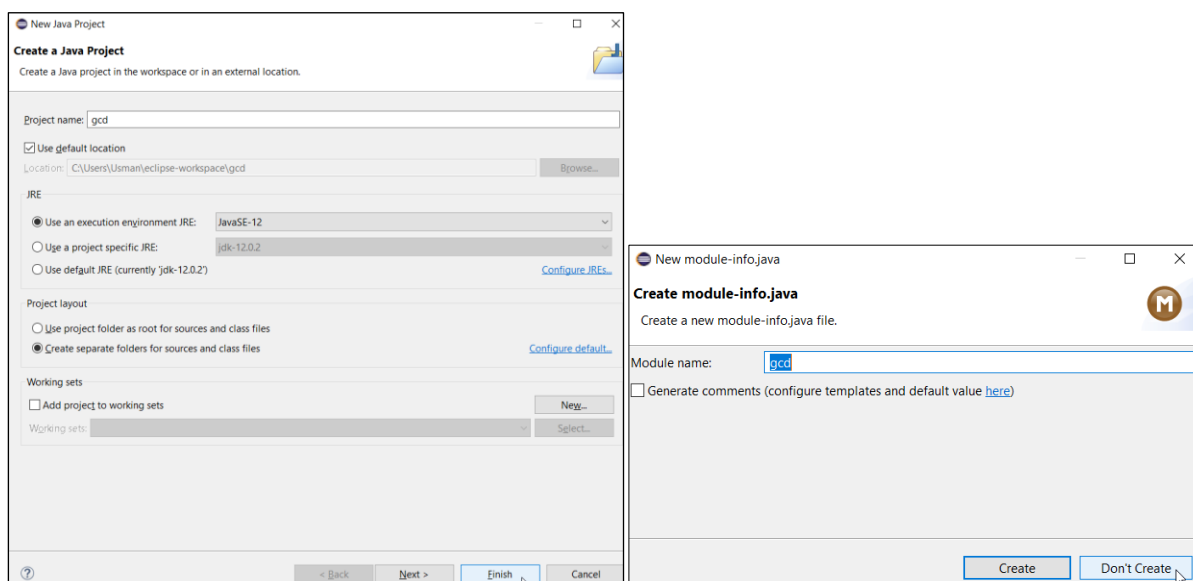
- 1) **Open Eclipse.** Select your workspace directory. Click **Launch**:



- 2) **Create a Project (File > New > Java Project):**

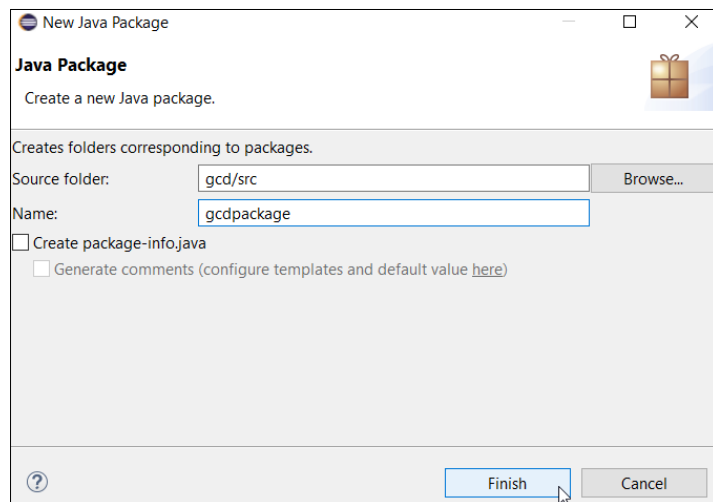
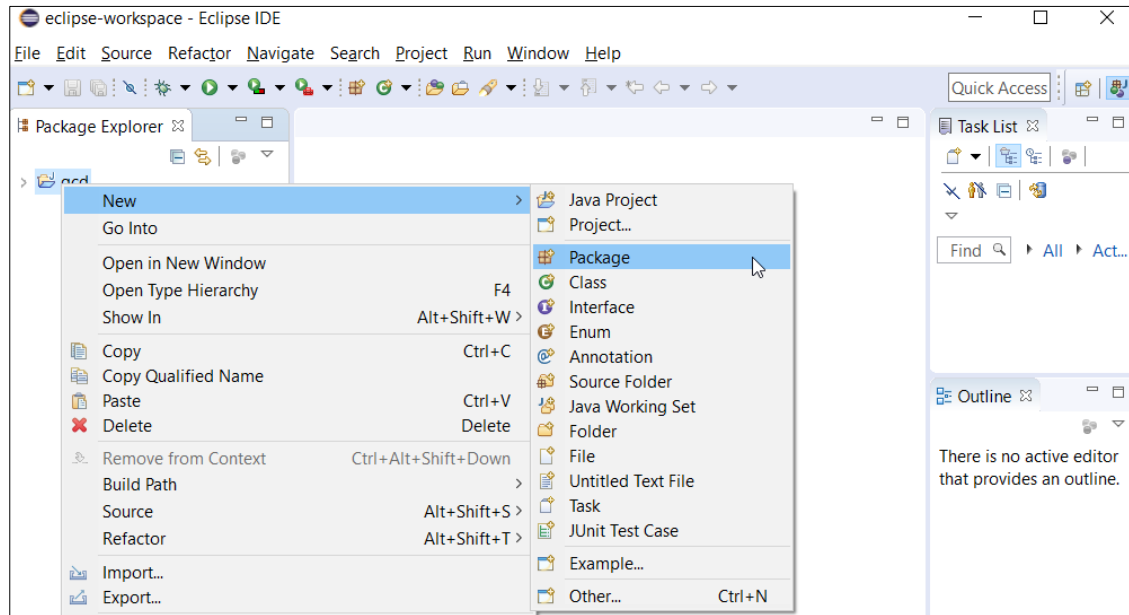


- 3) **Name the project as “gcd” > click Finish > click Don’t Create module:**

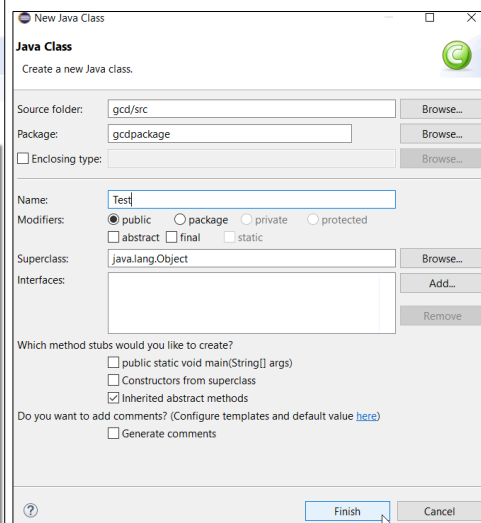
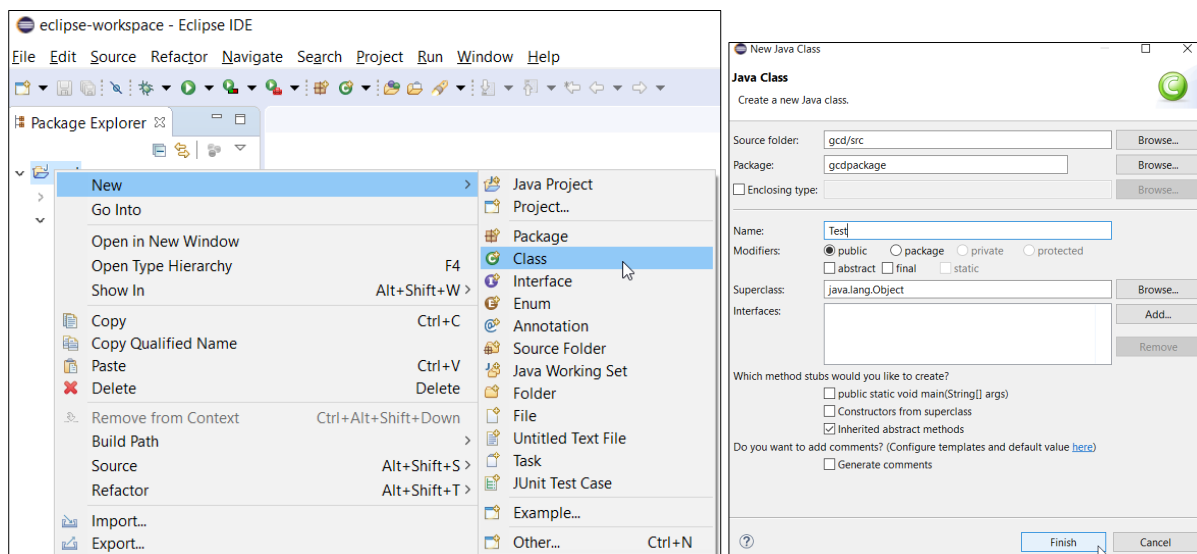


- 4) **Close the “Welcome” tab.**

5) Create a Package(right-click on Project Name > New > Package > Name it > Finish):

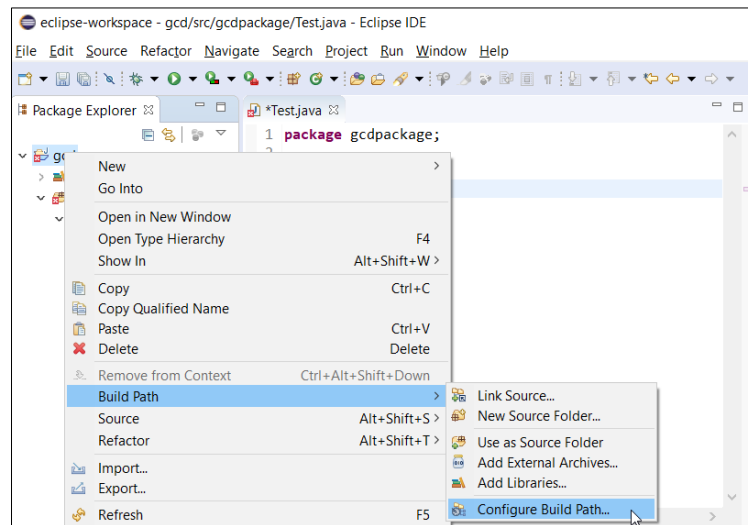


6) Create a Class(right-click on Project Name > New > Class > Name it > Finish):

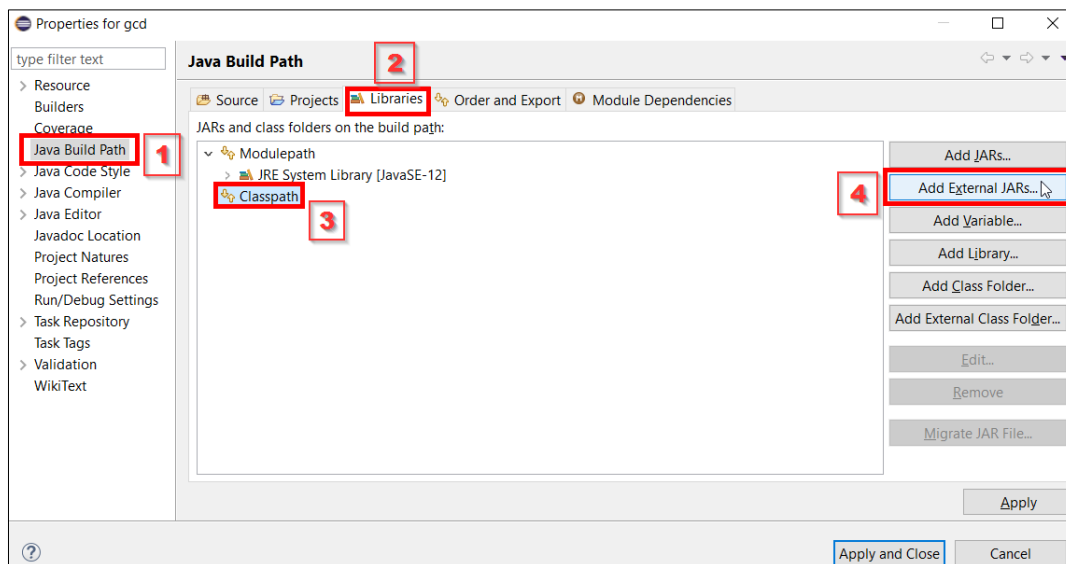


7) Adding “Selenium Server Driver and Client Driver(JAR files)” in Eclipse IDE:

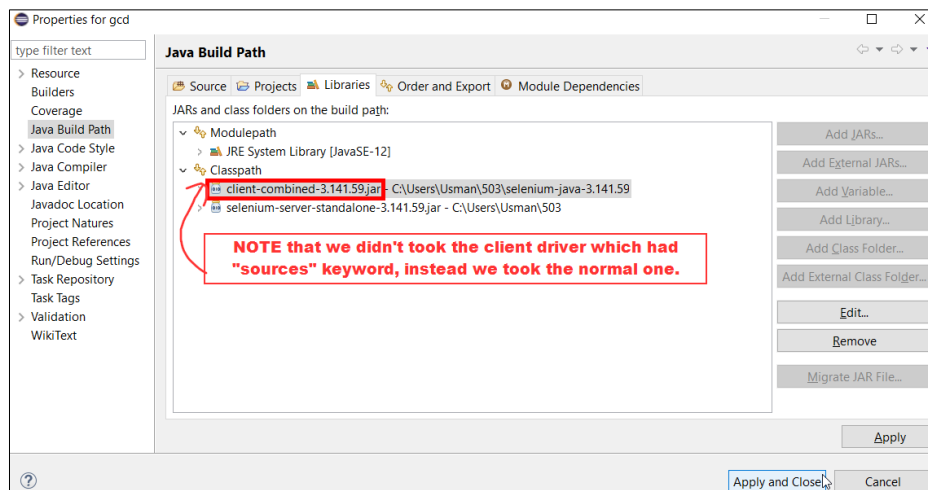
- right-click on Project Name > Build Path > Configure Build Path...



- now go under: Java Build Path > Libraries > Classpath > click Add External JARs...



- Browse and add JAR files > click Apply and Close :



8) Creating a link for HTML file(wherein calculation part is present):

(NOTE that this file will be run by the '**script in JAVA**'(which we'll create later))

- **Create a Notepad file** with the following code and save it as "**gcdhtml.html**":
---(gcdhtml.html)---

```
<html>
<head>

<script type="text/javascript">
function gcd()
{
    var x,y;
    x=parseInt(document.myform.n1.value);
    y=parseInt(document.myform.n2.value);
    while(x!=y)
    {
        if(x>y){x=x-y;}
        else{y=y-x;}
    }
    document.myform.result.value=x;
}
</script>
</head>

<body>
<center>

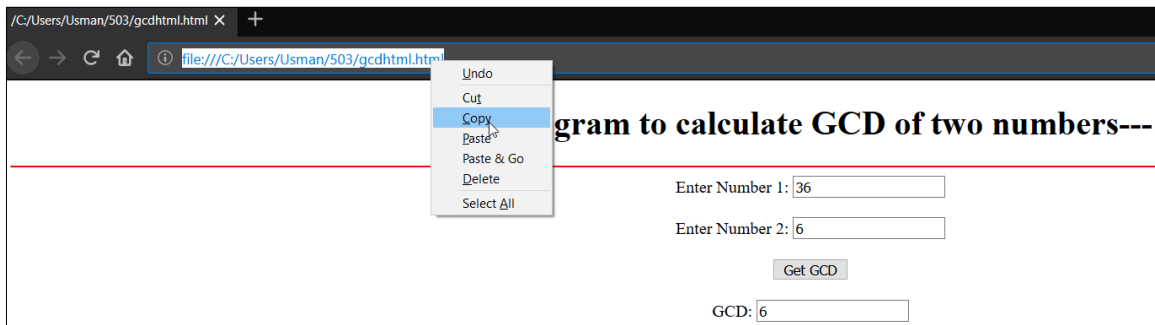
<h1>---Program to calculate GCD of two numbers---</h1>
<hr color="red">

<form name="myform">
    Enter Number 1: <input type="text" name="n1" value=""> <br> <br>
    Enter Number 2: <input type="text" name="n2" value=""> <br> <br>
    <input type="button" name="btn" value="Get GCD" onClick="gcd()"><br><br>
    GCD: <input type="text" name="result" value="">
</form>
</center>
</body>

</html>
```

- **Close the file.** Then **right-click > Open with > Firefox Browser**

- Copy URL from the webpage:



9) Creating the *script* in JAVA:

(NOTE that this **script** will be run by Eclipse IDE)

(In simple words, it's like we are

-ordering Eclipse to run a script or to do a job

-of opening the HTML file

-and putting the values in the textboxes with the help of Selenium Drivers

-and to show the result.

-Hence automating the work in browser)

- Now we'll put the path of "geckodriver" in a String **driverPath**
- And we'll **paste the copied URL in the .get()** method of the WebDriver class

---(Test.java)---

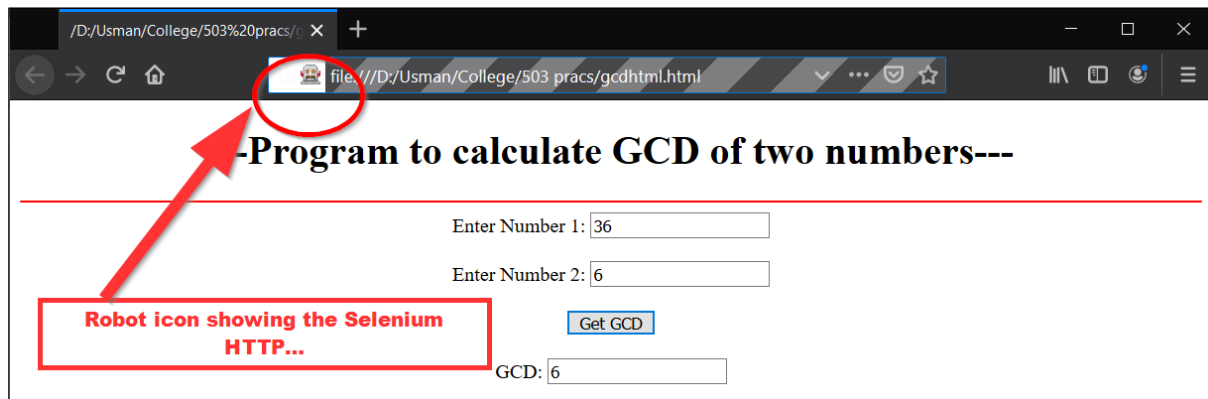
```
package gcdpackage;
```

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxOptions;
import org.openqa.selenium.firefox.FirefoxProfile;
```

```
public class Test {
    static String driverPath = "C:\\\\Users\\\\Usman\\\\503\\\\geckodriver.exe";
    public static void main(String[] args) {
        System.setProperty("webdriver.gecko.driver",driverPath);
        //DesiredCapabilities capabilities = DesiredCapabilities.firefox();
        //capabilities.setCapability("marionette",true);
        //ProfilesIni allProfiles = new ProfilesIni();
        FirefoxProfile fp = new FirefoxProfile();
        fp.setPreference(FirefoxProfile.PORT_PREFERENCE,"7055");
        FirefoxOptions options = new FirefoxOptions();
        options.setProfile(fp);
        WebDriver driver=new FirefoxDriver(options);//(capabilities);
        //WebDriver driver=new ChromeDriver();
        driver.get("file:///D:/Usman/College/503%20pracs/gcdhtml.html");
        driver.manage().window().maximize();
        driver.findElement(By.name("n1")).sendKeys("36");
        driver.findElement(By.name("n2")).sendKeys("6");
        driver.findElement(By.name("btn")).click();
        String
result=driver.findElement(By.name("result")).getAttribute("name=result");
        System.out.println("GCD="+result);
    }
}
```

10) Run the file from Eclipse IDE:

- **OUTPUT:**



11) Finish!

What is Gecko Driver:

-a WebBrowser engine inbuilt within Mozilla Firefox browser.

-acts as a proxy between Web Driver enabled clients(Eclipse, NetBeans, etc.) and Mozilla Firefox browser *or simply* acts as a link between Selenium Web Driver tests and Mozilla Firefox browser.

What is Selenium Remote Control (RC):

-a test tool that allows you to write automated web application UI tests in any programming language against any HTTP website using any mainstream JavaScript-enabled browser.

-Selenium RC comes in two parts.

- 1) A server which automatically launches and kills browsers, and acts as a HTTP proxy for web requests from them.
- 2) Client libraries for your favourite computer language.