Using Primitive Data Types

Objective

This lab will begin to introduce the Java syntax to the new Java programmer. You will create your first Java program and experiment with using different primitive data types.

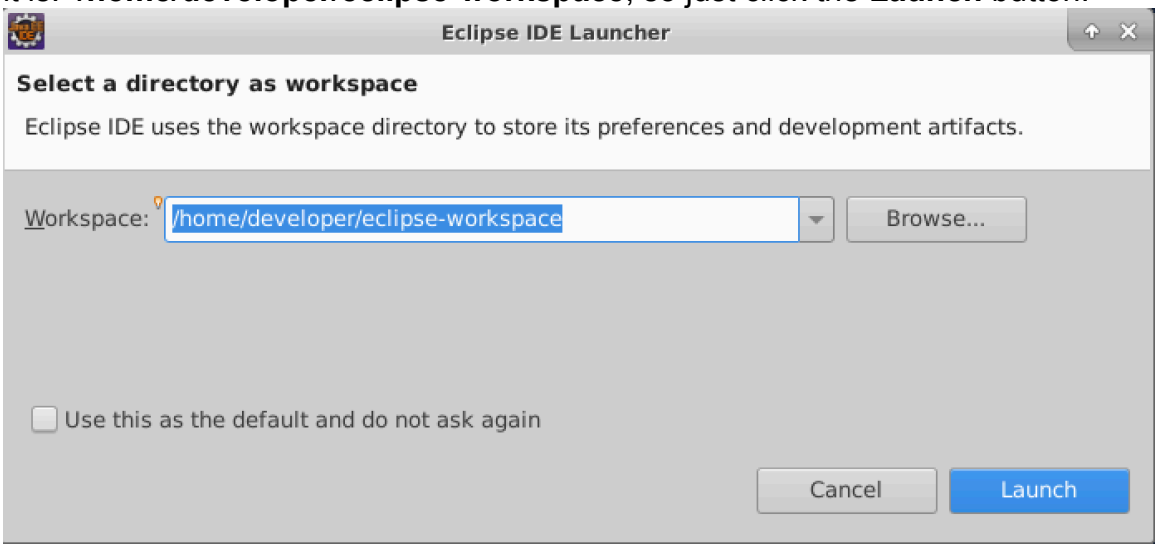
Overview

In this lab you will create a new Java project and your first Java class. While the details of object-oriented programming and Java class structure are yet to come, this initial Java class will allow you to begin writing some basic Java code.

Step by Step Instructions

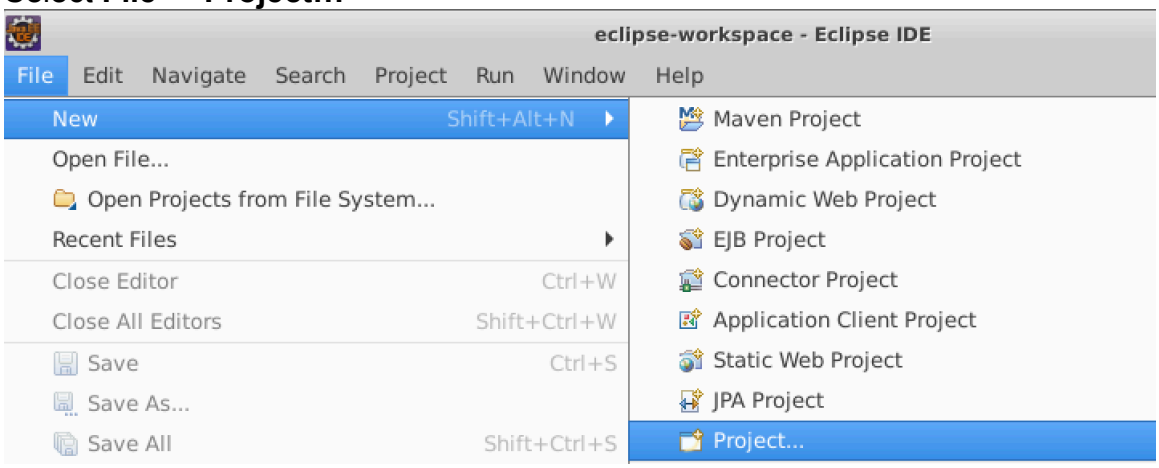
Create a Workspace, Project, and Package

1. Launch your IDE and use the default workspace for your lab exercises. The default workspace location should be left unchanged in this environment and it is: **/home/developer/eclipse-workspace**, so just click the **Launch** button.

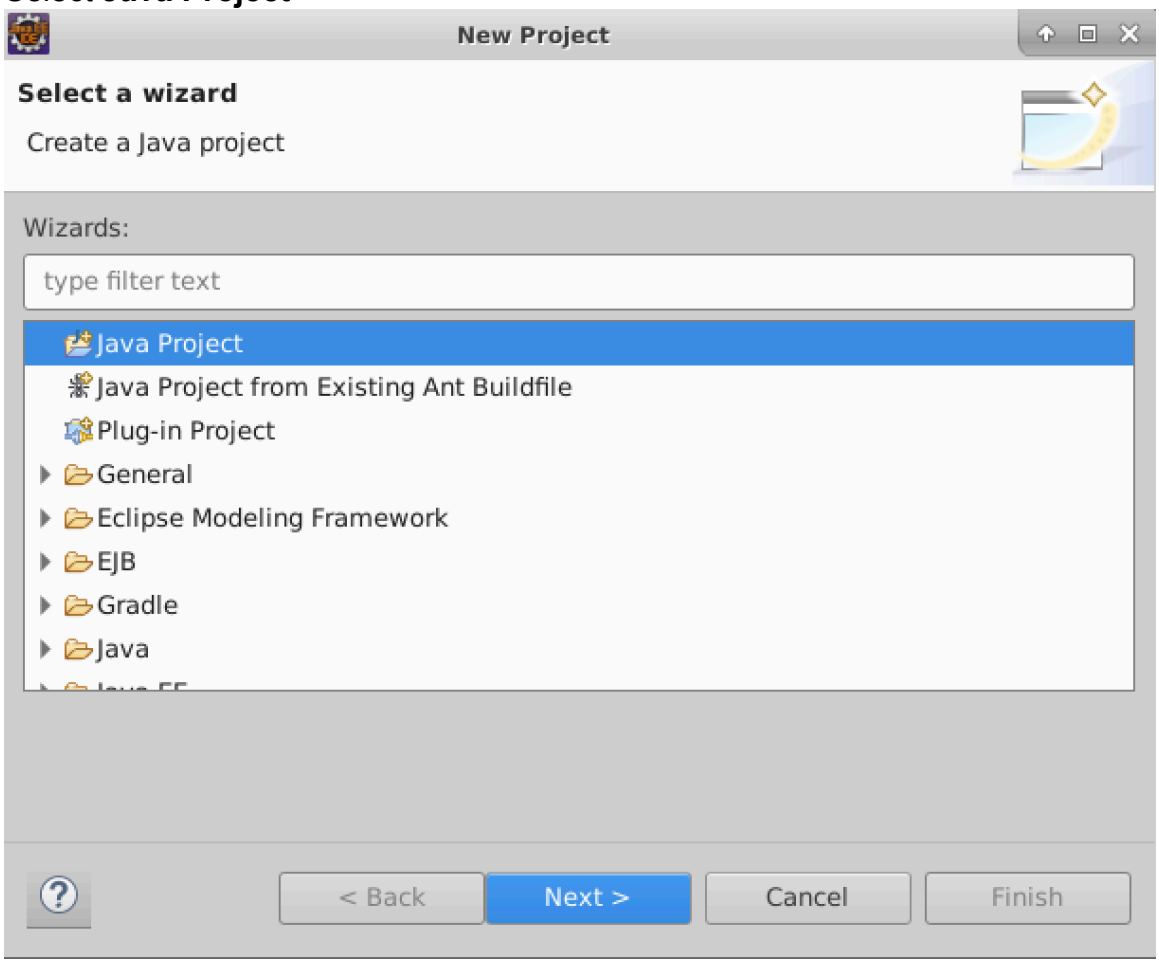


2. Create a new Java project for the first few exercises. Name the project **ClassExercises**.

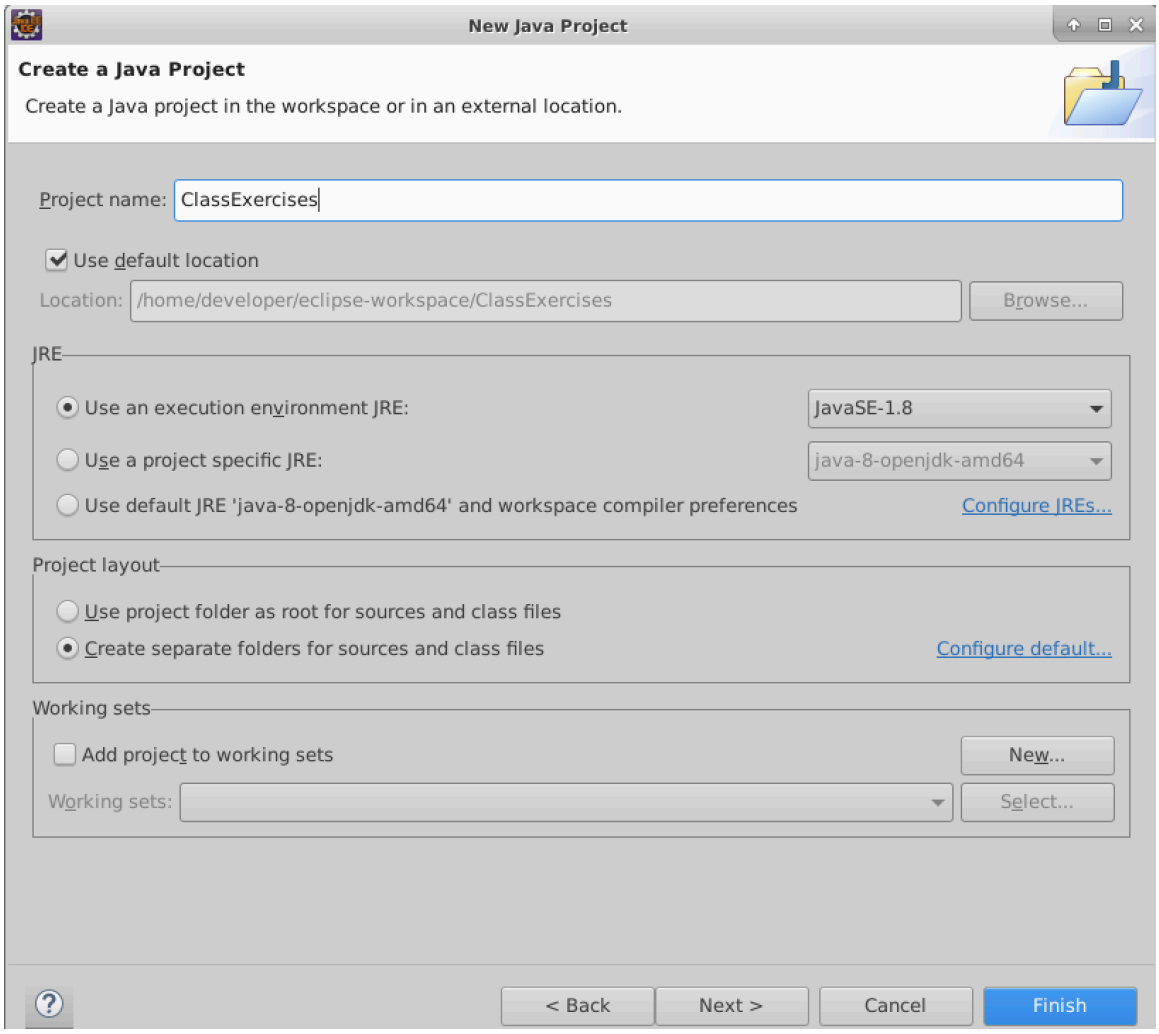
   a. Select **File** -> **Project…**



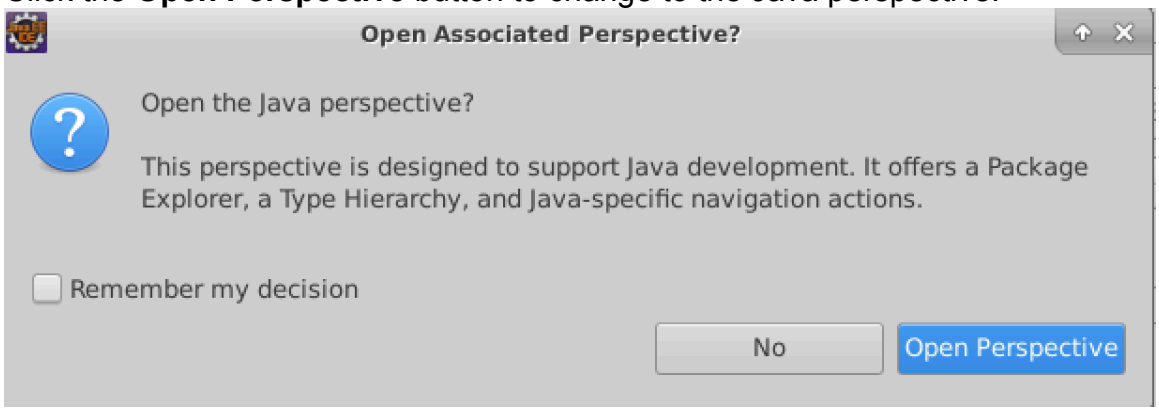
   b. Select **Java Project**



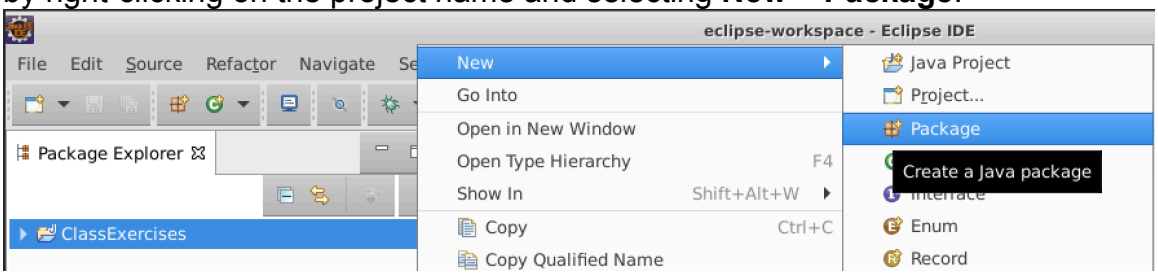
c. Enter **ClassExercises** as the Project Name and click the **Finish** button.

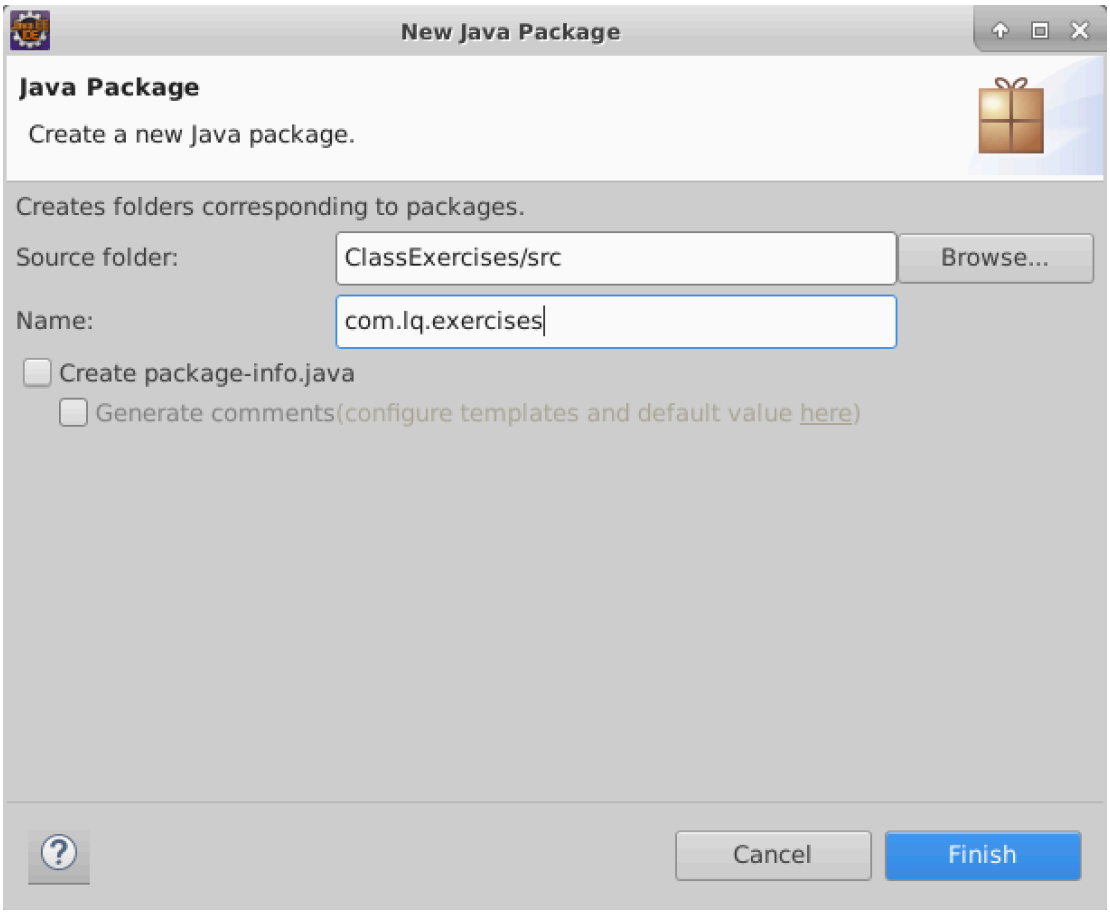


d. Click the **Open Perspective** button to change to the Java perspective.

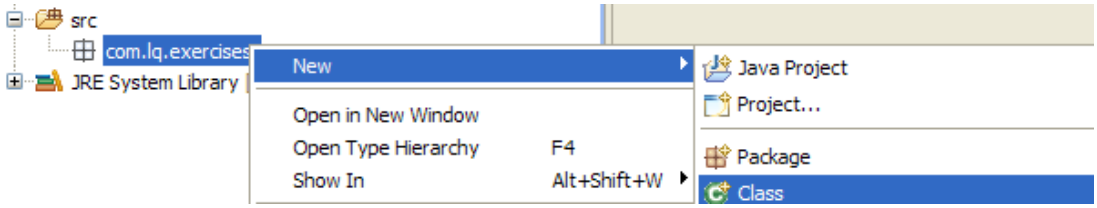


3. Create a package named **com.lq.exercises**. This can be done in most IDEs by right-clicking on the project name and selecting **New > Package**.

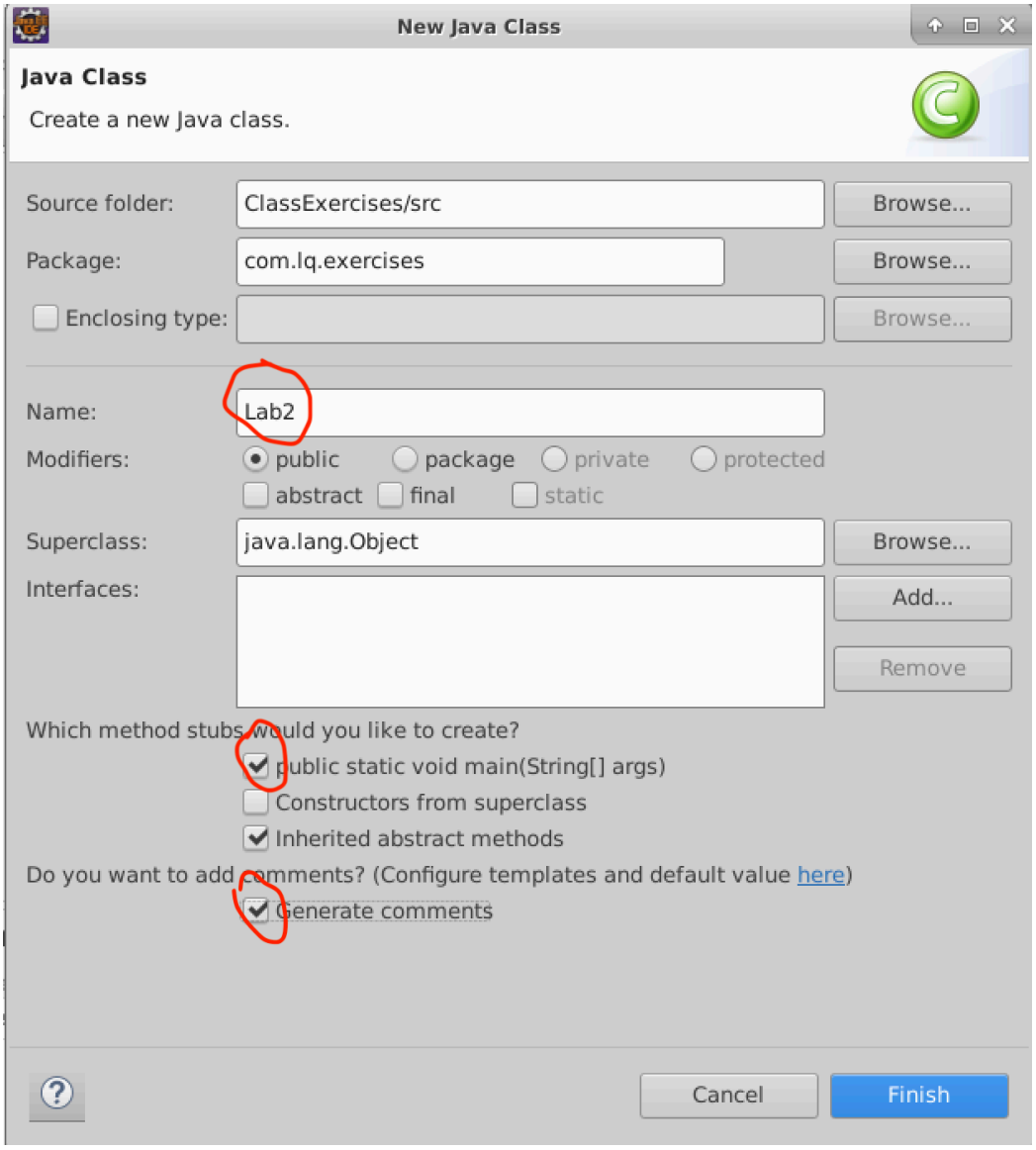




4. Create a new class named **Lab2** in the **com.lq.exercises** package. This can be done in most IDEs by right-clicking on the package and selecting **New > Class**.



The new class wizard will appear. Enter the name of the class, check the box for a main method, and check the generate comments method. Click the **Finish** button.



5. The **Lab2** class will open in the editor. It will contain a beginning curly brace '{'and an ending curly brace '}' for the class definition. Proceed with the lab exercises that follow.

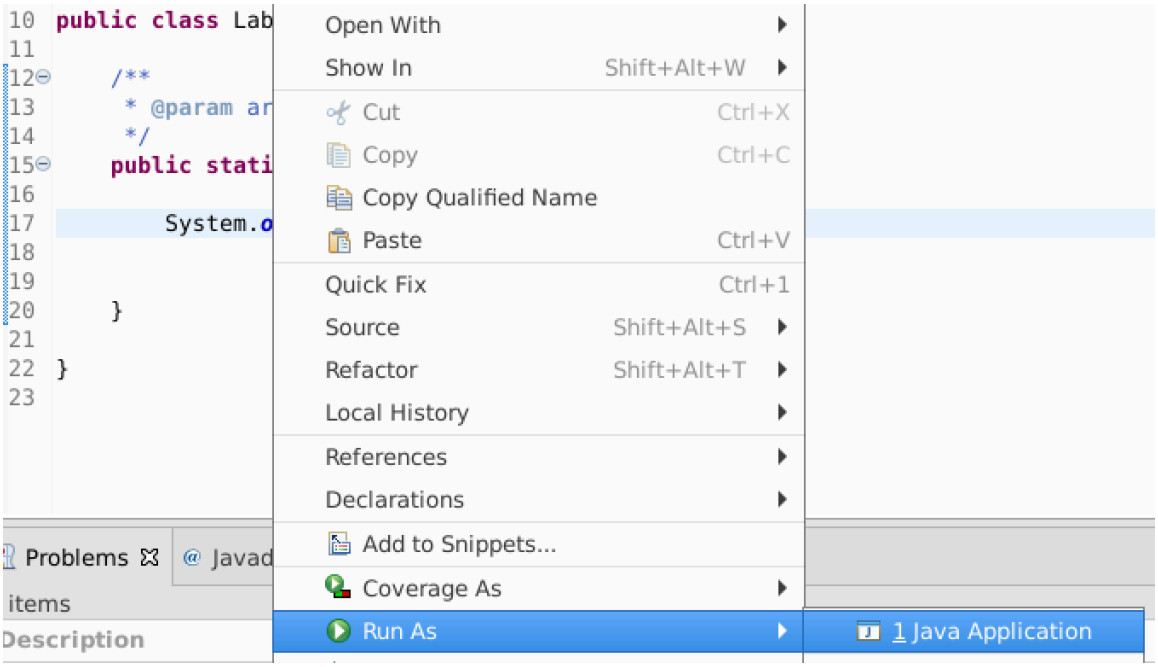
<u>Implement the main() method</u>

6. Remove the TODO comment. It is there only to remind you that the method was automatically created for you and that you must implement what the method is to do.

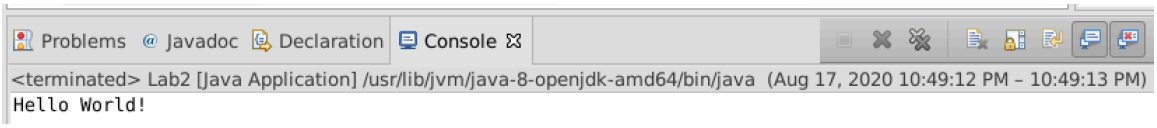
7. Add a print statement to your `main()` method that says, "**Hello World!**". We will cover the details of how this all works later but for now it will look like the following:

```
1  /**..
4  package com.lq.exercises;
5
6  /**
7   * @author developer
8   *
9   */
10 public class Lab2 {
11
12     /**
13      * @param args
14      */
15     public static void main(String[] args) {
16
17         System.out.println("Hello World!");
18
19
20     }
21
22 }
23
```

8. Save your changes and execute your small Java program. In most IDEs this can be accomplished by right-clicking on the class name and selecting **Run As > Java Application** or by clicking the run button on the toolbar.



9. View the results in the console view. It should be similar to the following:



Define Primitive Data Types

10. At the beginning of the main() method, declare the following variables with the specified characteristics:

    a. An int with the name **width** and no initial value.

b. An int with the name **height** and no initial value.
c. An int with the name **area** and no initial value.
d. A double with the name **radius** and an initial value of **10.0.**
e. A double with the name **pi** and an initial value of **3.14.**
f. A boolean with the name **result** and an initial value of **true.**

In the body of the main() method, perform the following variable assignments:

g. Assign a value of **8** to **width.**
h. Assign a value of **12** to **height.**
i. Assign a value of **96** to **area.**

11. Near the bottom of the main() method, print the values of each variable using `System.out.println()`. Use the following as a guide:

```
System.out.println("The value of width is " + width);
```

12. Execute your program. Your output should be similar to the following:

```
Problems  @ Javadoc  Declaration  Console
<terminated> Lab2 [Java Application] /usr/lib/jvm/java-8-
Hello World!
The value of width is 8
The value of height is 12
The value of area is 96
The value of radius is 10.0
The value of pi is 3.14
The value of result is true
```

<u>Define Arrays</u>

13. Near the top of your main() method, define the following arrays:

j. An array of 12 **ints** named **daysInMonths**.
k. An array of 12 **String** references named **monthNames** – initialize this array at the time it is declared with the names of the 12 months (refer to your course book for the syntax).

14. Write 12 lines of code to assign the number of days in each month to the **`daysInMonths[]`** array elements (do not worry about leap year!). See output in step 16 below for days in months.

15. Write a print statement for each month that will display the name of the month and the number of days it contains. Use the following as a guide:

```
System.out.println( monthNames[0] + " has " + daysInMonths[0] + " days." );
```

If you are already familiar with loops, you may use them. If not, then use 12 output statements.

16. Execute your program. Your output should be similar to the following:

```
Problems  @ Javadoc  Declaration  Console
<terminated> Lab2 [Java Application] /usr/lib/jvm/java-8-
January has 31 days.
February has 28 days.
March has 31 days.
April has 30 days.
May has 31 days.
June has 30 days.
July has 31 days.
August has 31 days.
September has 30 days.
October has 31 days.
November has 30 days.
December has 31 days.
```