Creating Your First Java Projects in Eclipse

Introduction

About Java Packages

- Java Packages are used to easily identify classes without name conflicts, and to keep related .java files together
- There are many packages available in the Java API
- The Java API is a library of pre-written classes that are free to use and are available in the Java Runtime Environment
- You will also create your own classes that are best kept in a package that you create
- If you do not create a package for your classes, the default package will be used by NetBeans, Eclipse, or other IDEs
- **For the examples in this course, you will often use the default package to test coding examples using the instructions in Part 1**
- For **Part** 2, you will create a .java file inside a package called **mysecondproject**
- When you create a .java file inside a package called **mysecondproject**, you need to have the package declaration as your first line of code:
    - package mysecondproject;
- If you prefer to use the default package, a package declaration is not used.

Part 1: Create a Java Project and Add an Existing .java File

Overview

In this section, you will download the reference material HelloWorld.zip, Create a new project in Eclipse, and add an existing java file to the project.

Tasks

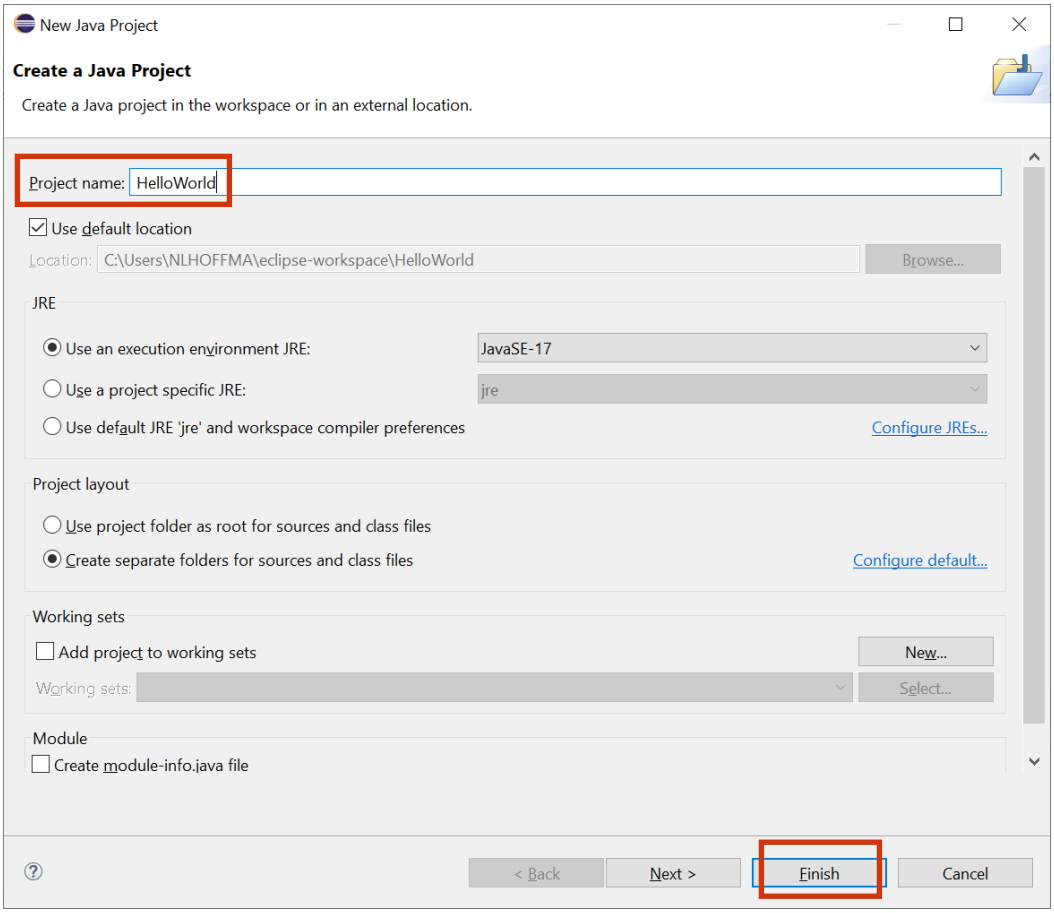
1. Download the file HelloWorld.zip from the reference materials for this practice

2. Extract the zip file HelloWorld.zip, noting the location

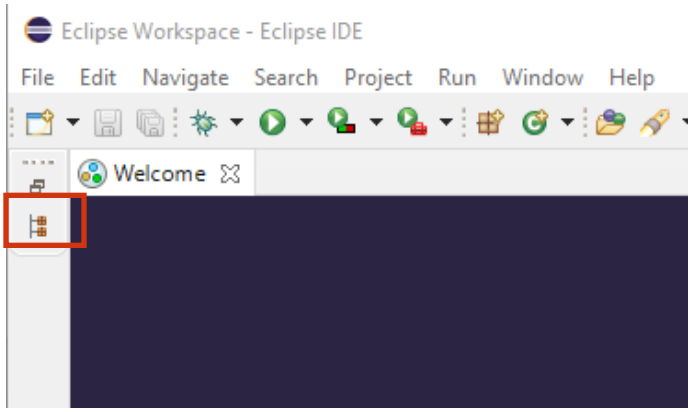
3. Launch Eclipse

4. Go to **File> New> Java Project**
    a. Add the project name – HelloWorld
    b. For this course, it is recommended that you name your projects the same as the .java file

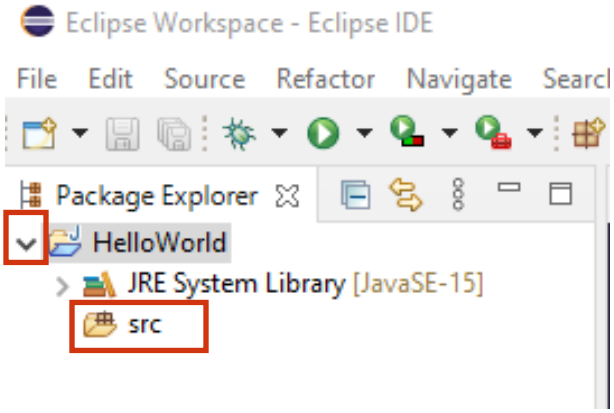
5. Click **Finish**



6. **Click the Restore** icon to see the Package Explorer window (if it is not open)

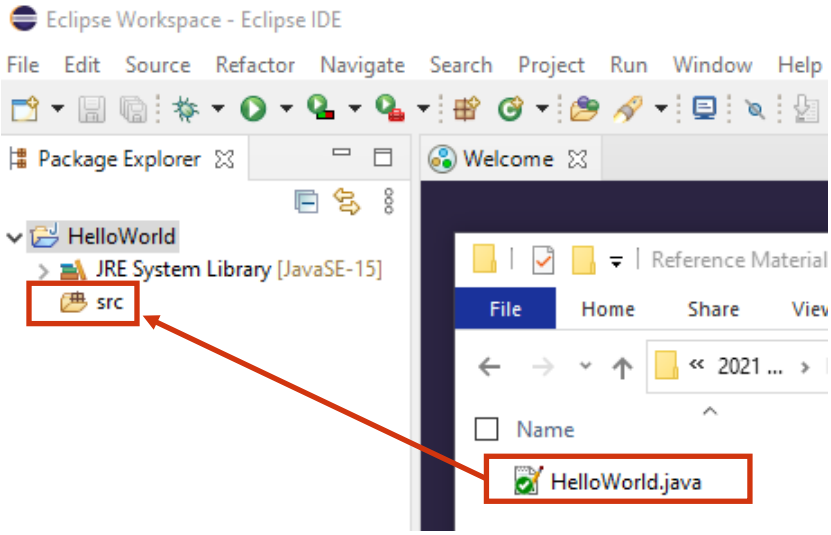


7. In the Package Explorer window, **click the arrow** to expand the project and show the src package

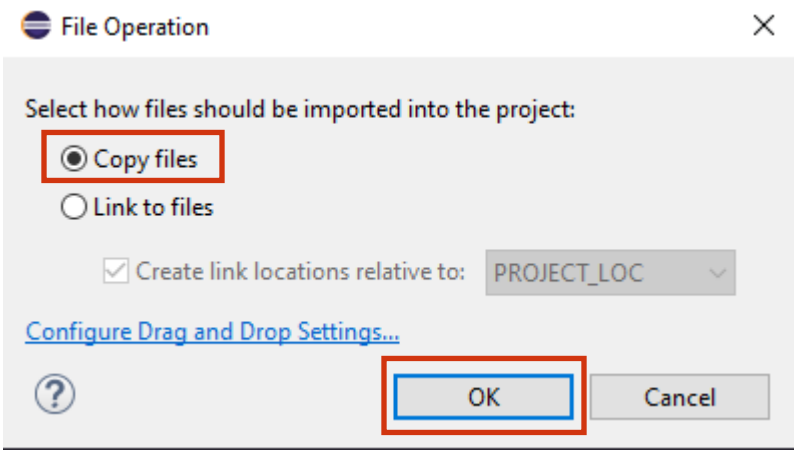


8. **Navigate to the folder** in which you extracted the HelloWorld.java file in step 2

9. **Drag** the HelloWorld.java file from the folder on your device and **drop** it on the **src** folder

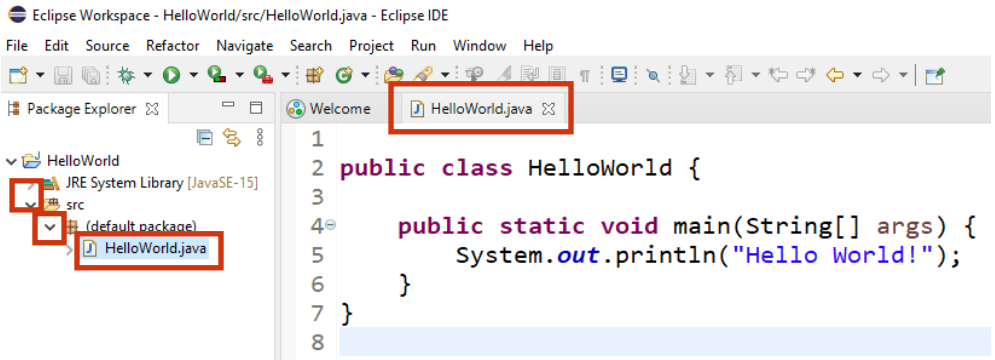


10. When prompted, **select Copy Files, click OK**

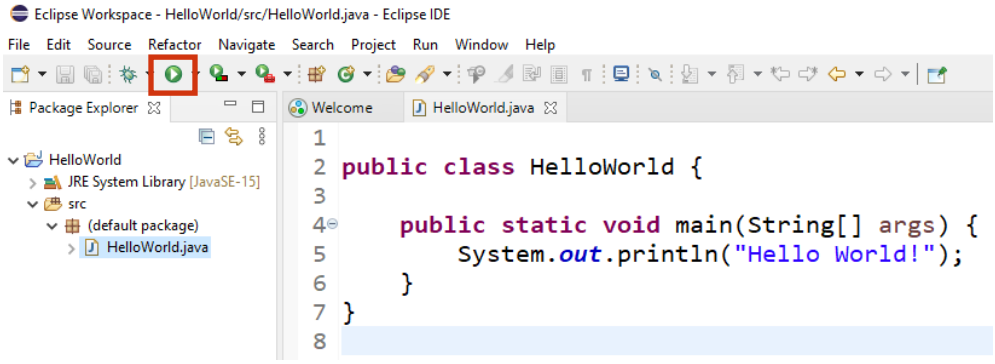


11. **Click the arrow** next to the src folder then **click the arrow** next to (default package) and you will see the HelloWorld.java file in the default package

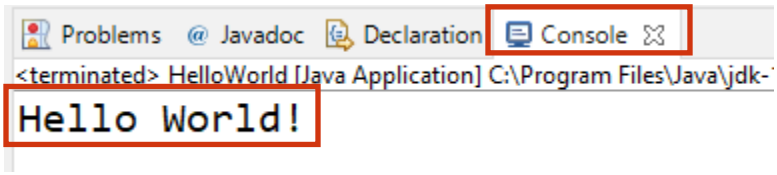
12. **Double click** the .java file and it will open in a new tab in the code editor window



13. **Click the Run button** to test



14. In the Console window below the Code Editor, you should see the message “Hello World”



Part 2: Create a Package and Project in Eclipse with a main class

Overview

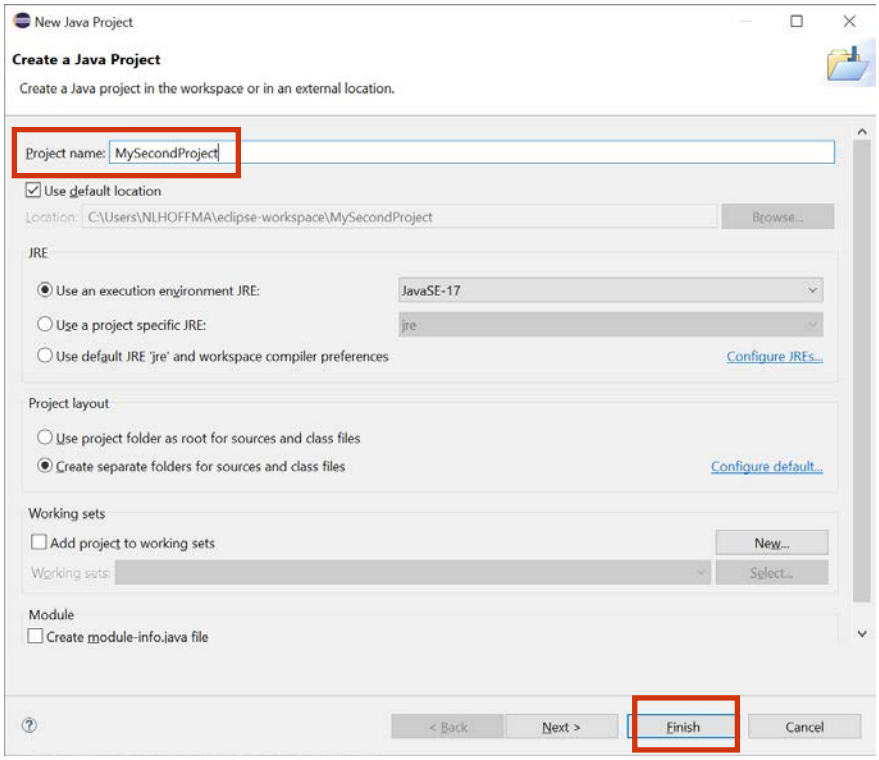
In this section, you will create and run an Eclipse java project inside a package, with a main Java class

Tasks

1. Launch Eclipse if not opened

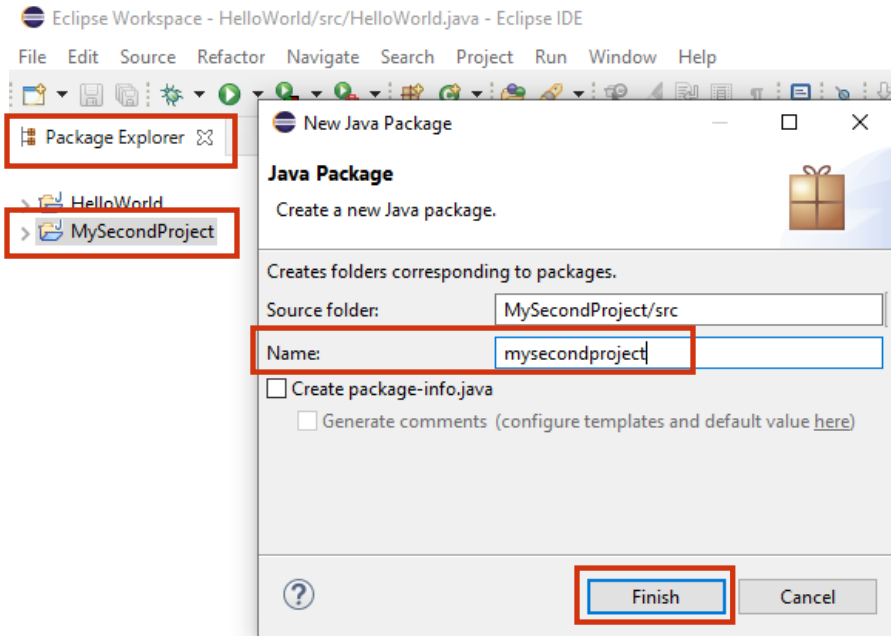
2. Go to **File> New> Java Project** and select the following:
    a. Add the project name - MySecondProject

3. Click **Finish**

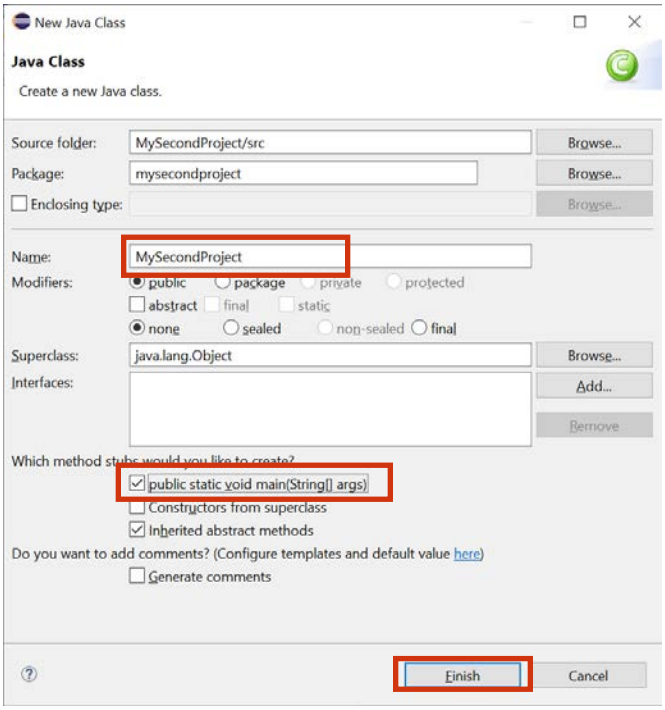


4. In the Package Explorer **right click** on the project **MySecondProject> click new> Package**

5. Enter the name **mysecondproject,** and then click **Finish**



6. In the Package Explorer **right click** on the package **mysecondproject > click new> Class**
7. Enter the name **MySecondProject, and then check the box to create public static void main**, **click Finish**

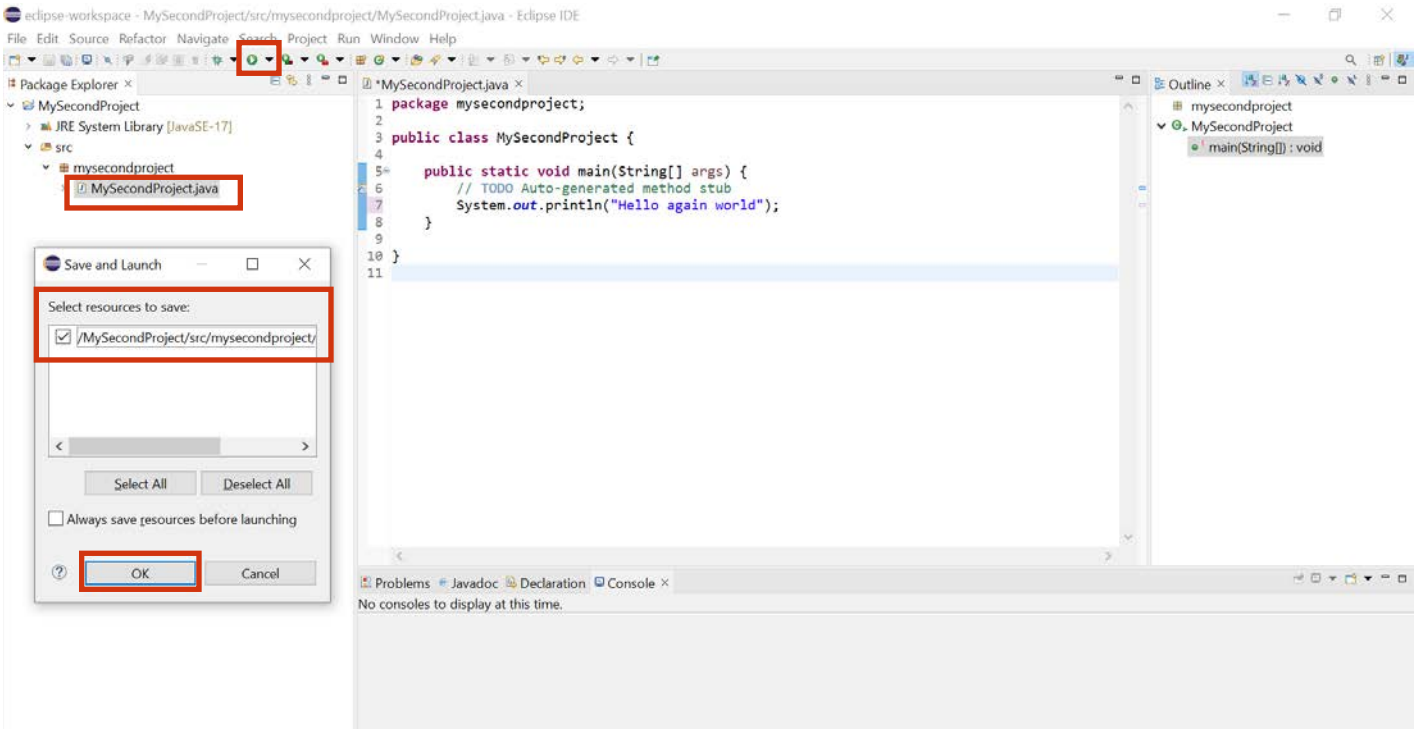


8. The newly created MySecondProject class will open in a new tab in the code editor

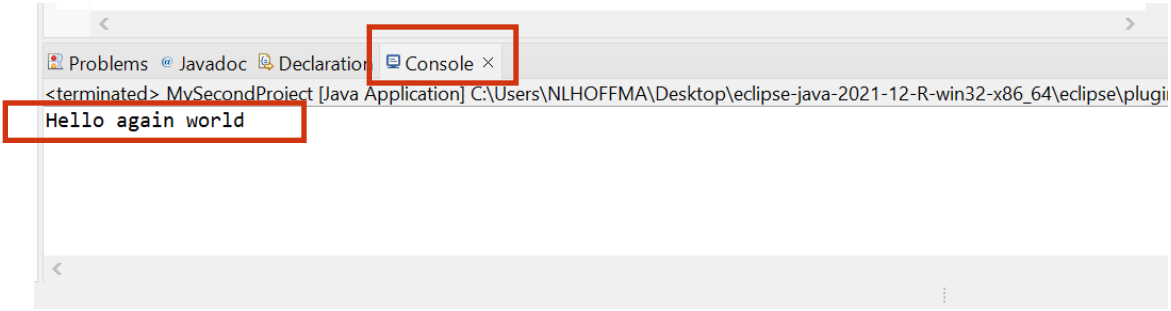
9. In the Code Editor, locate the main method of the MySecondProject class and **enter the line of code** as shown below:

   System.out.println("Hello again world");

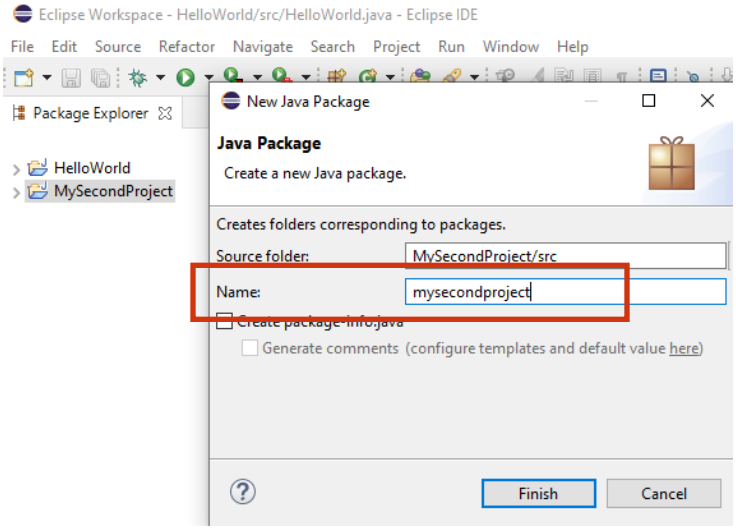
10. Ensure the class MySecondProject.java is highlighted in the Package Explorer, and then **click the run button** to test (or click the small down arrow at the right of the run button and select which java file you want to run) - If prompted **click OK** to save the resources



11. You should see the message “Hello again world” displayed in the Console output at the bottom of the IDE



Note an important difference between the first project and the second project:
In task 5 of MySecondProject, you added a package explicitly, and named it mysecondpackage.

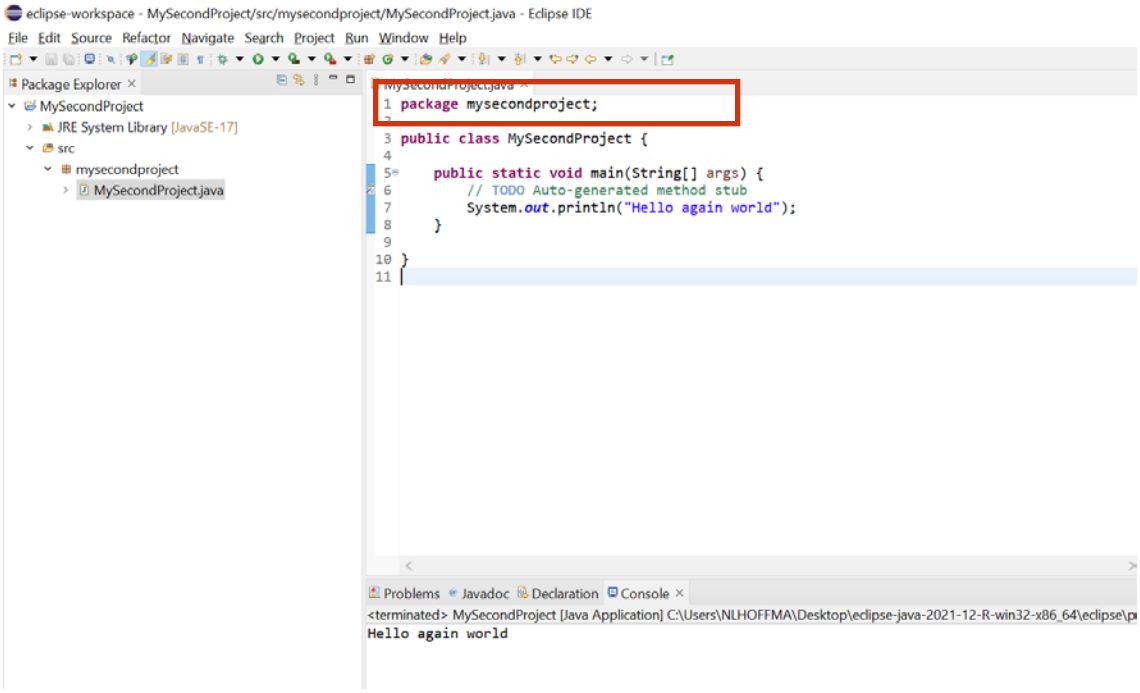


In Part 1, you did not create a Main Class you simply dragged an existing .java file into the src folder, the package was implicitly created and named (default package) by Eclipse

If a package has a name other than (default package), it is important to note that a package declaration must be included as the first line of code in the Java file

If you explicitly create a class in an existing package, Eclipse will automatically add the package declaration for you

In the code editor for your second project, scroll to the top of the code and you will see the package declaration



Observe that in your first project, as the package was (default package), if you scroll to the top of the java file in the code editor, you will see that there is no package declaration

If you add an existing Java file to a package that has already been named, you need to add the package declaration manually as the first line of code in the java file using the format:

package packagename;