

Using Eclipse for Java Programming

NOTE: Before using Eclipse for the first time, create a folder on your desktop named MyWorkspace. This folder will be used to store all of your Java projects.

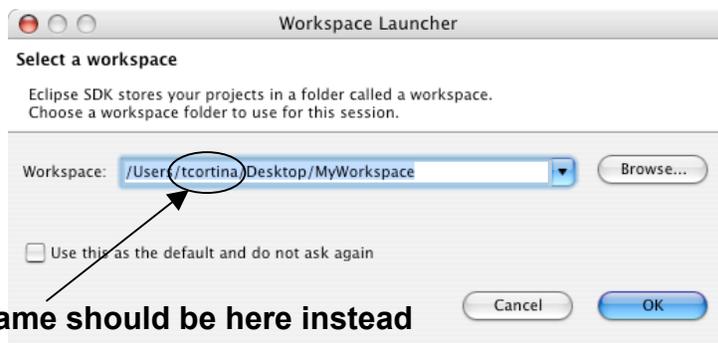


1. Start Eclipse. (Windows: Find it under All Programs in your *start* Menu. Macintosh: Find it under Go > Applications in the Finder.)
2. Select a workspace:



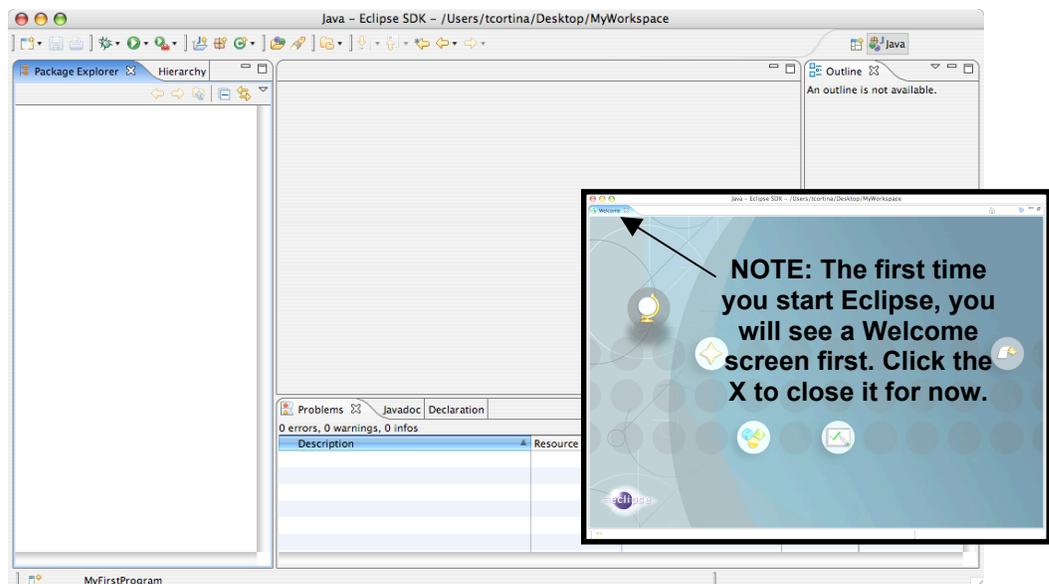
the Eclipse icon

- (a) Click on **Browse** and select the folder MyWorkspace from your Desktop. **DO NOT USE ANY OTHER WORKSPACE.**



your username should be here instead

- (b) Click **OK**. After some setup, you will see the Eclipse SDK window with a Package Explorer tab:



3. Each Java program we will write will be stored in its own project. From the Eclipse menu bar at the top of the screen, choose **File > New > Java Project...** You will see the following New Java Project window:



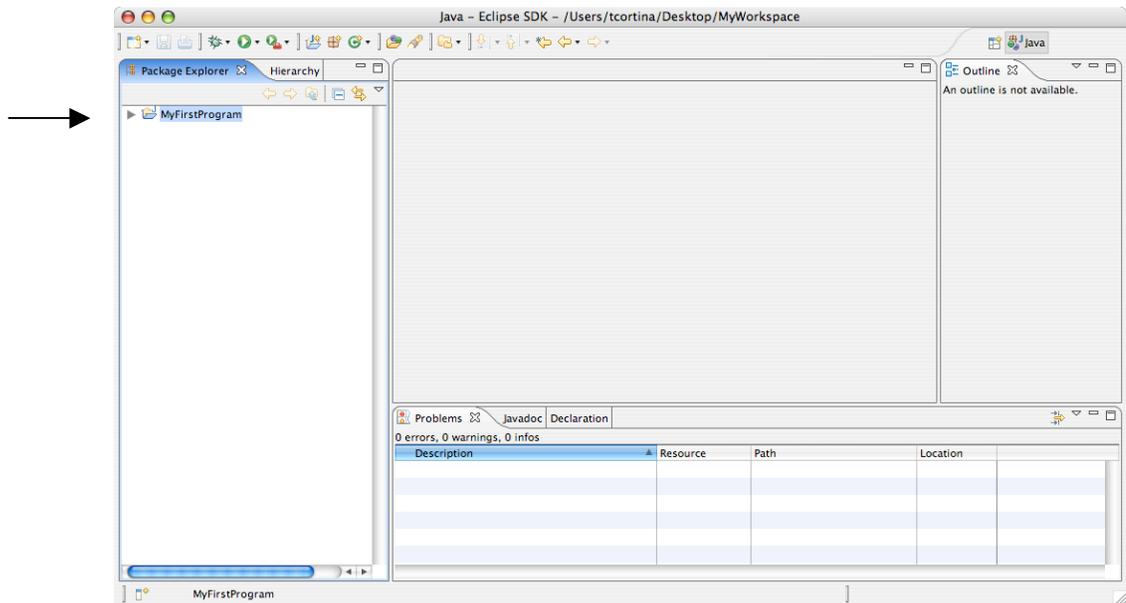
If you are creating a Java project from scratch, you will select the first option and go to Step 4.

If you are creating a Java project from files given to you, you will select the second option and go to Step 6.

4. If you are starting a **brand new Java project** without any initial program code:

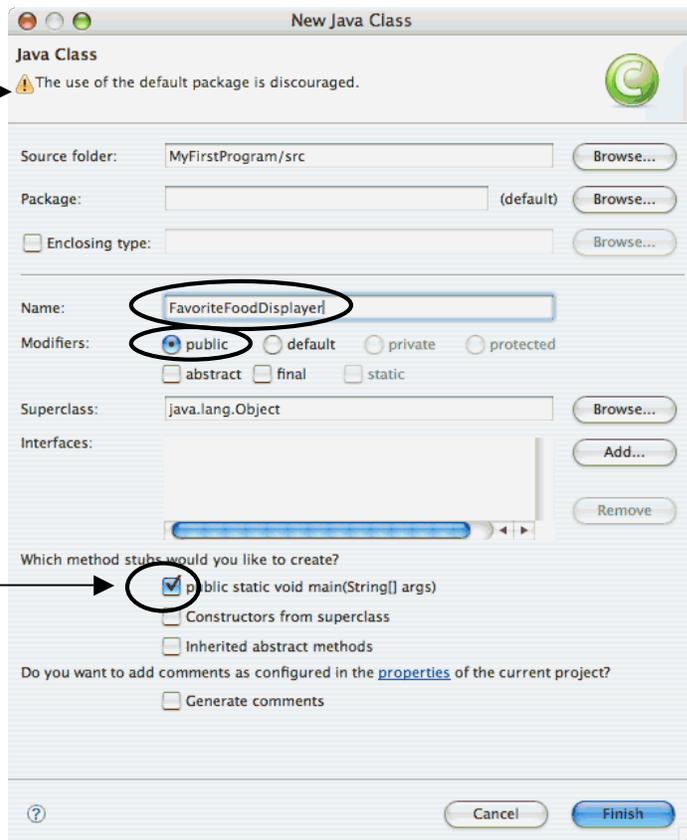
- In the **New Java Project** window, give the project an appropriate name that's easy to identify. For example, type in **MyFirstProgram** as the **Project Name**. You will notice that the **Location** is automatically updated to be a folder in your workspace with the same name.
- For the "Contents" option, make sure you select "Create new project in workspace". Generally, you do not need to change any other options in this window. You should be using Java compiler 1.6 (or a subsequent release) for your work.
- Click **Finish**. You will see the SDK display your project in the left column, as shown in the picture on the next page.

If you are starting a Java project using a folder with Java code that is already written for you, go to Step 6 instead.



5. Each program has one or more classes, one of which must contain a `main` method. To add a class to a project:
 - (a) Be sure that the name of the project folder is highlighted (e.g. see picture above).
 - (b) Then, from Eclipse's menu bar at the top of your desktop, select **New > Class**. (See next page for a picture.)
 - (c) Type in the name of the class in the space provided (e.g. `FavoriteFoodDisplayer`).
 - (d) Select the **public** modifier.
 - (e) If this class will contain a `main` method, check off the appropriate box under "Which method stubs would you like to create?" (In this example, our class will have a `main` method.)

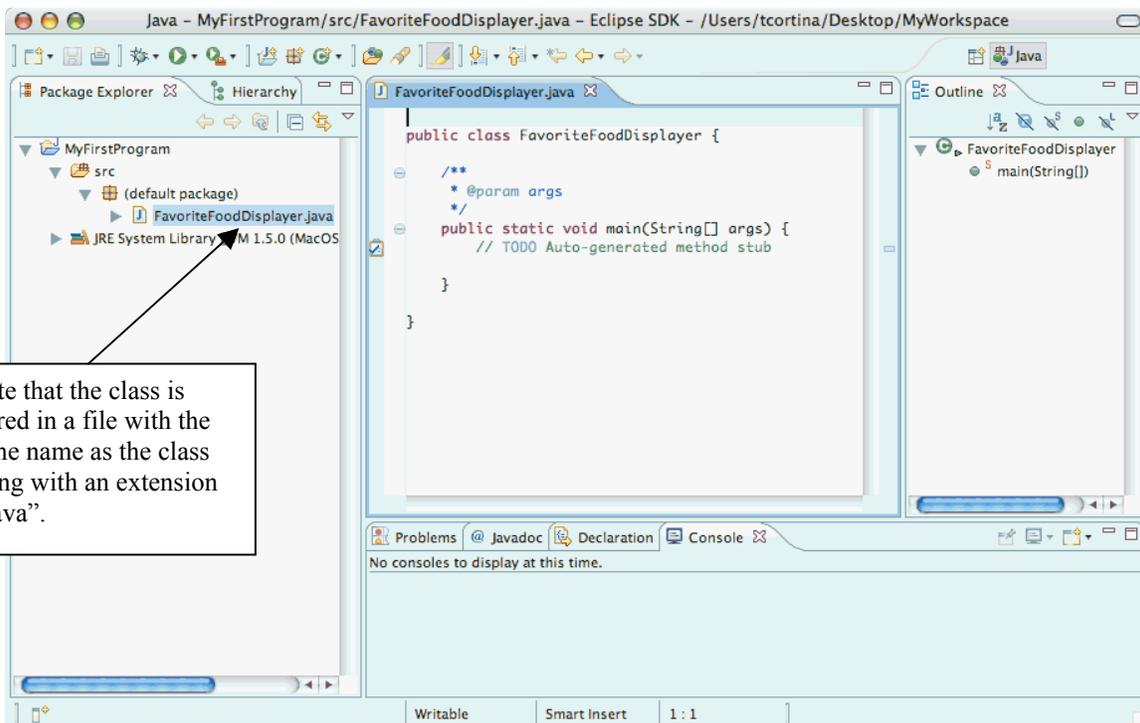
This warning can be ignored for your class.



Check this box only if your class requires a main method.

(f) Click Finish. You will see this class appear in the Package Explorer in your default package.

Note that the class is stored in a file with the same name as the class along with an extension "java".



- (g) Enter the following highlighted Java code exactly as it appears in the main method as shown below:

```
/**
 * @param args
 */
public static void main(String[] args) {
    // TODO Auto-generated method stub
    String favoriteFood = "pizza";
    System.out.print("My favorite food is ");
    System.out.println(favoriteFood);
}
```

This line is an automatically-generated comment. You may erase this line if you wish.

If you make any syntax errors, you will see a red X appear to the left of the line with the error. Click on the X to get more info about the error if you need help.

For example, if you leave the quotation marks out from the code above, you will see something like this:

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    String favoriteFood = pizza;
    System.out.print("My favorite food is ");
    System.out.println(favoriteFood);
}
```

Syntax error

- (h) Periodically, save your work using File > Save.
- (i) If your Java program requires more than one class, you can add additional classes to your project, repeating the instructions of this step.
- (j) When you are ready to run your program, continue with step 7.

6. If you are **creating a project using Java files that are already stored in a folder** for you (for example, the folder is named MySecondProgram):

If you are starting a Java project from scratch, go back to Step 4.

- (a) **IMPORTANT: Make sure the folder with the Java files is stored inside your MyWorkspace folder before you go on to the following steps.**
- (b) For the “Contents” option, select “Create project from existing source”. Click the **Browse** button to select the folder that contains the Java files that you want to use to create the new project.
- (c) In the **New Java Project** window, enter the name of the same folder as the **Project Name**. (A message will appear as you are typing that will warn you to use the same name as the project folder.)
- (d) Click **Finish**. (See pictures on next page.)

