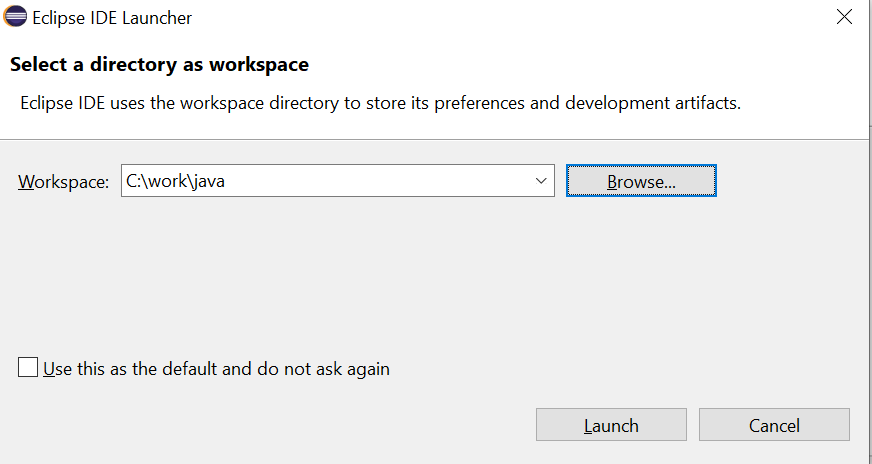
How to **edit**, **compile** and **run** your first Java program in Eclipse

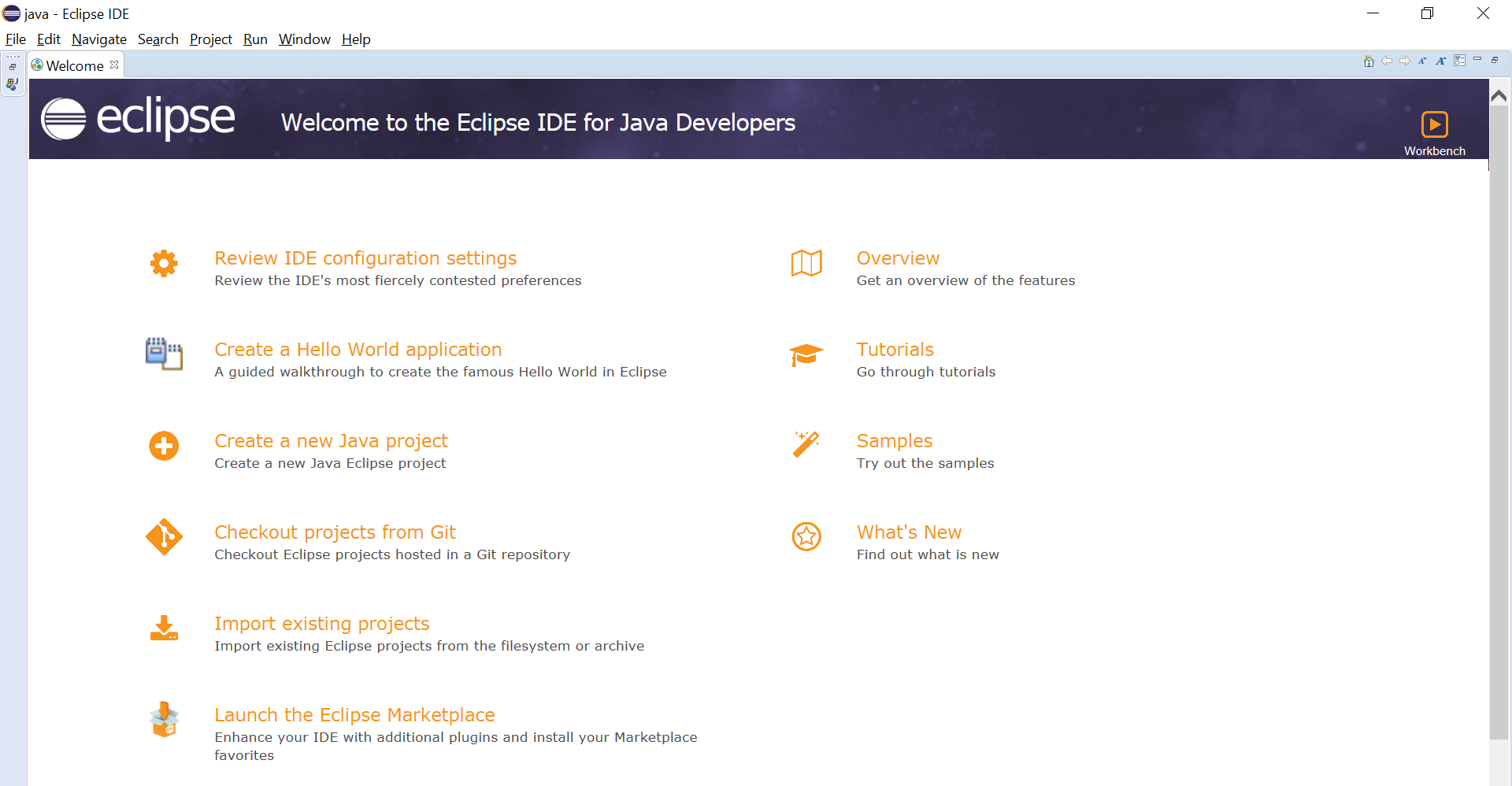
Before you can finish the task in the document, you need to install JDK and Eclipse first by following the instructions in the schedule document of this week. After that, you can work on the steps below.

Fire eclipse from the desktop shortcut you make, or click “eclipse.exe” directly from the extracted folder “eclipse”.

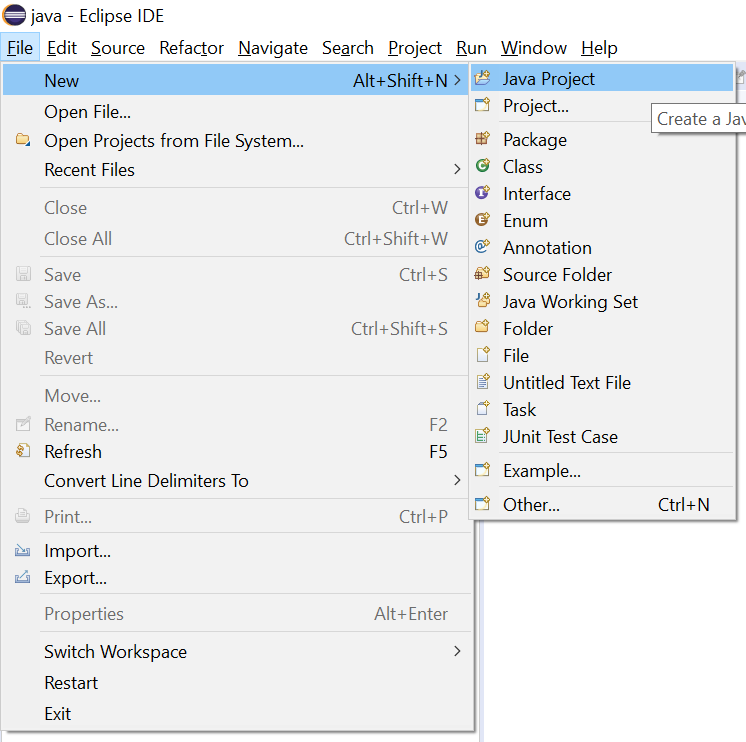
The first time you run Eclipse, it may ask you how to set the “workspace” folder, which is a folder that stores all the eclipse project files, including the source code that you will develop. You can click the “Browse” button and locate a local directory as the workspace folder. You may want to remember the path of this folder, so that you can refer to at all projects saved in this folder. For example, you can type in “c:\work\java” as workspace folder. To avoid being asked this question again when you run eclipse next time, you can check the box that says “use the workspace as default and do not ask again”. Then click “Launch”, as indicated below:

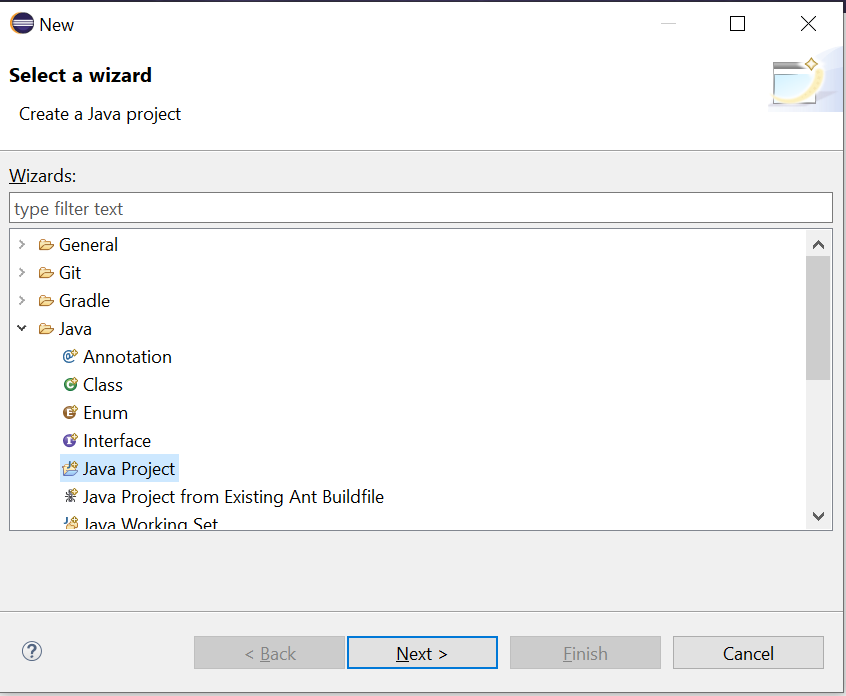


After eclipse starts, you can see the Welcome page as below. There is a “Tutorials” icon in the welcome page, and you can explore further when you have time. For now, just click the X mark in the “welcome” tab and close it.



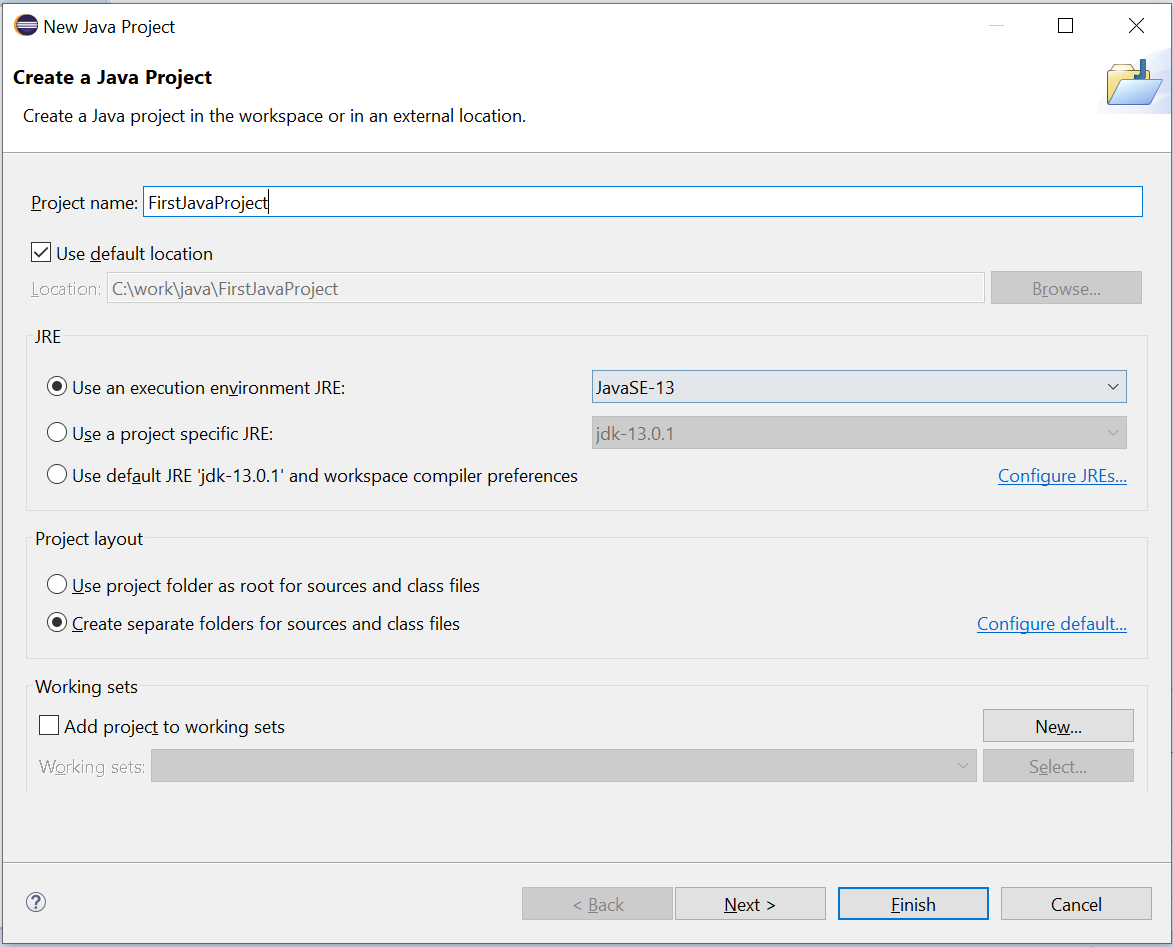
Then in Eclipse IDE, either click **File -- > New -- > Java Project** directly; or, if “Java Project” is not an option under “New”, then click **File -- > New -- > Other…**, and then in the popup dialog box, click folder “Java”, and you will see “Java Project”, then select it, and click next. Please refer to the two screen shot pictures below:





After Java Project is selected, the window below will pop up. Type in *FirstJavaProject* into “Project name” box.

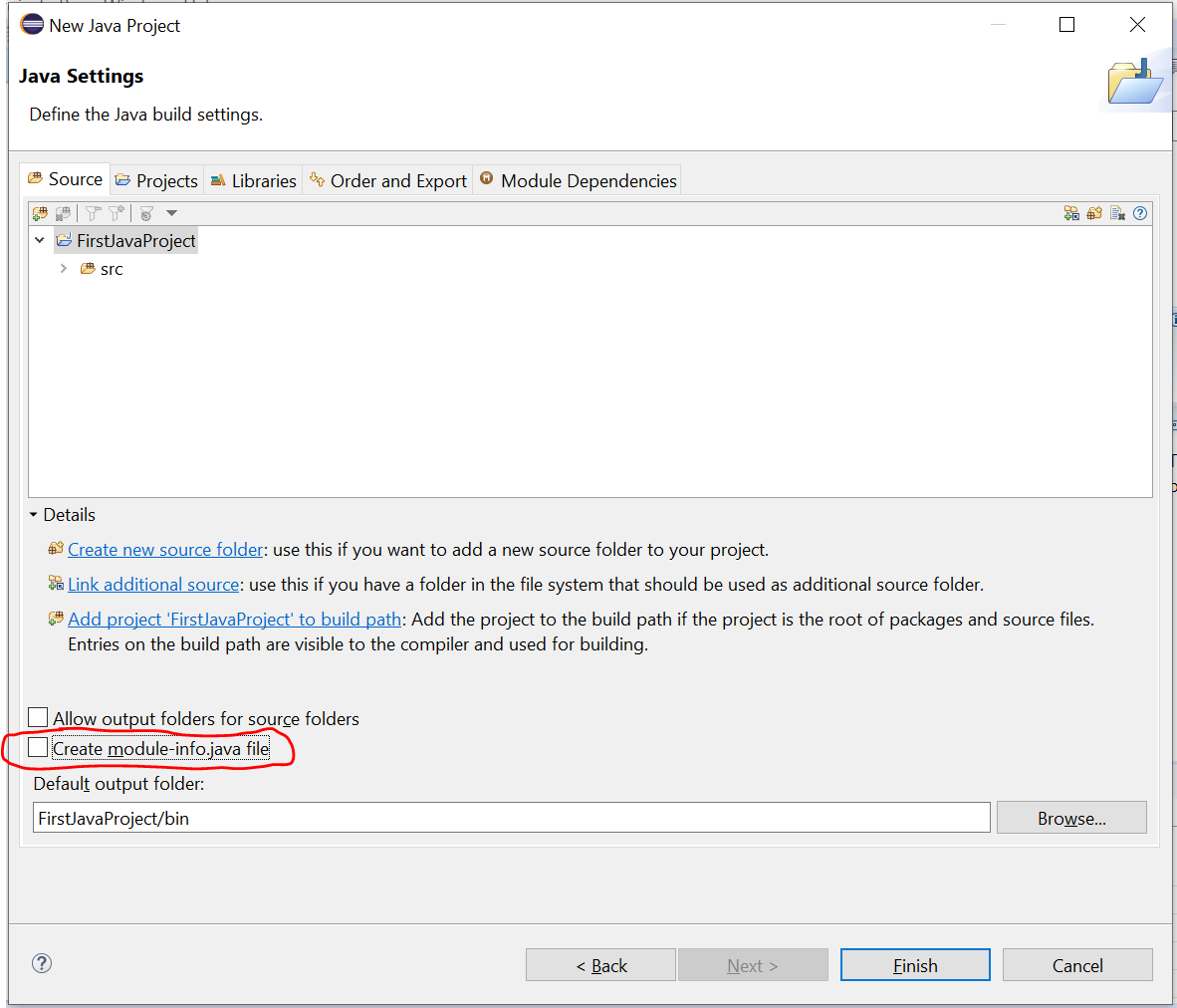
Double check the location field, and make sure the project is saved in folder “c:\work\java\FirstJavaProject”. If the default location is not “C:\work\java\FirstJavaProject”, then you need to uncheck the “Use default location” box, and type in the location where you want to store the project.



The “location” value in the window above may be different in your case, depends on where you set your default directory for your workspace when you first run eclipse (refer to page 1 of this document for detail). For example, if your default workspace is configured as **c:\school\programming**, then you will see this in the location box: **c:\school\programming\FirtstJavaProject**

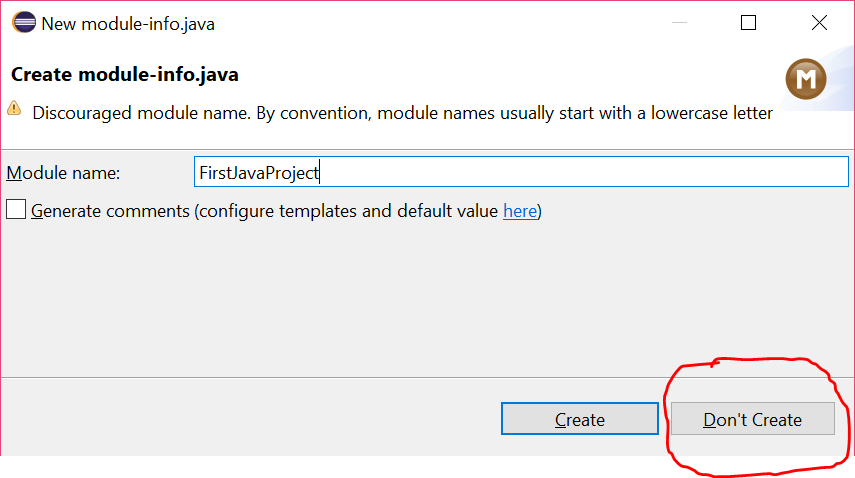
The ***default JRE is currently jdk-13.0.1***, because I installed jdk-13.0.1 earlier, and your most recent installation of jdk may have a different number in the last digit, which is the minor version. For example, you may have 13.0.2 or even later version, if that version is available when you install jdk.

Verify the ***Location*** value in the window above is correct, which should be your workspace location followed by your project name. Leave everything else untouched in the window above, then click the ***Next***> button at the page end. Now you are in this page:

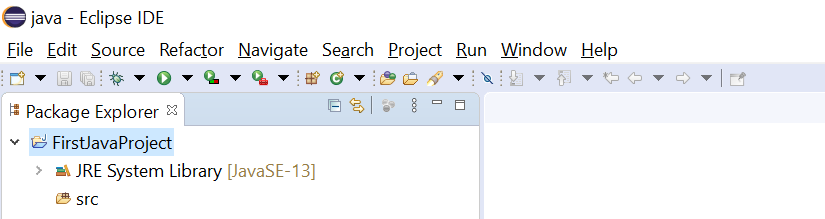


Then in the above window, please uncheck box *“Create module-info.java file”*, then click the Finish button.

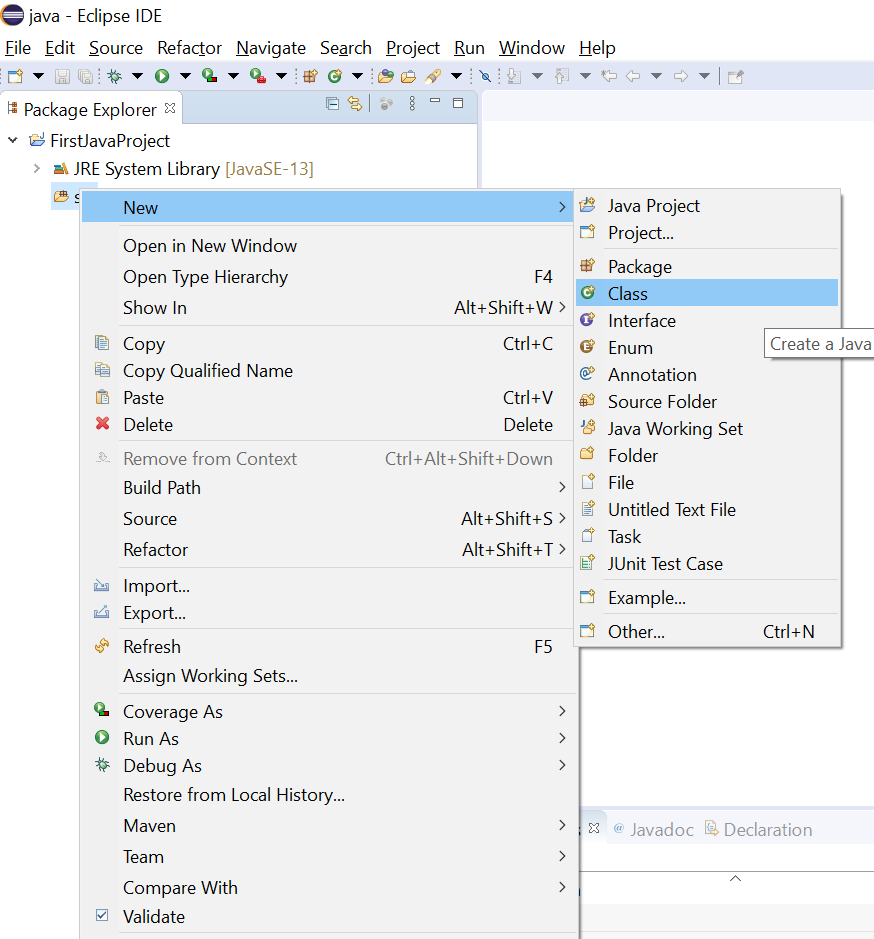
If you forget to uncheck this option, then this pop-up window will show up asking you whether to create a module file, please select “Don’t Create”.



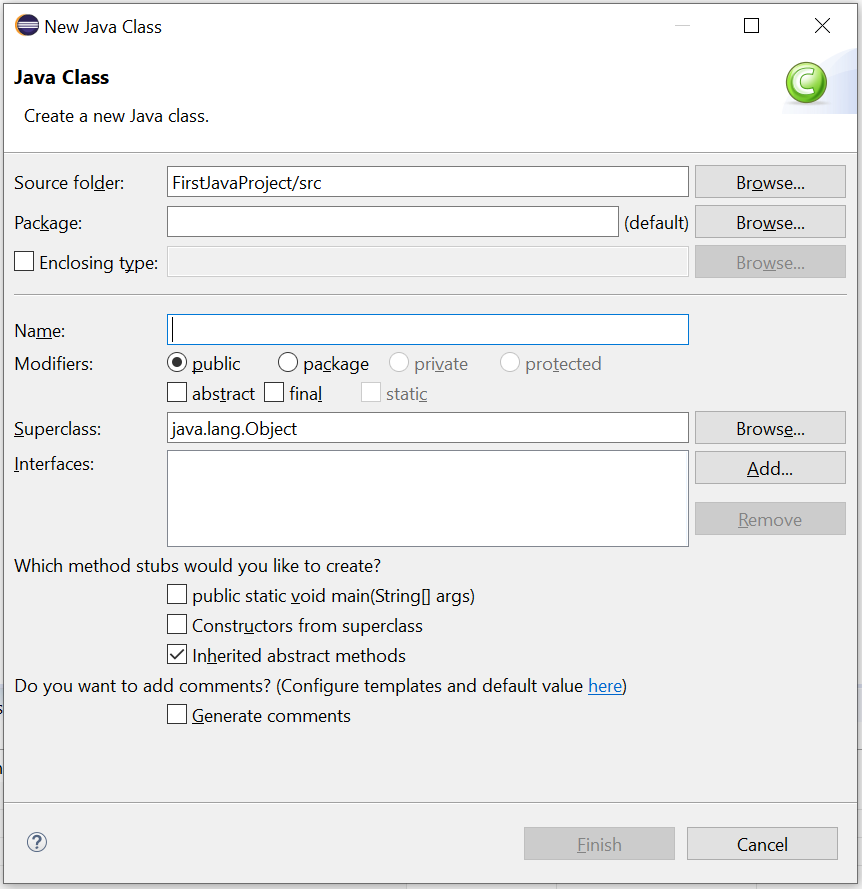
Then you are directed to the window below. If you don’t see the “JRE System Library” item as below, that means you may need to click the expand arrow to the left of the project name “FirstJavaProject” to expand the folder and see the content of folder. Even if you create the modu-info.java in the previous step, you can still right click the java source code file in the src folder and delete it. Currently, the “FirstJavaProject” folder has one ***src*** subfolder for storing java source code, and the other subfolder is JRE System Library that contains Java Runtime Environment (JRE) library files.



Now we need to create a new source code file in this project. Right click the **src** folder, and in the context menu, select **new 🡪 Class** , as indicated below:

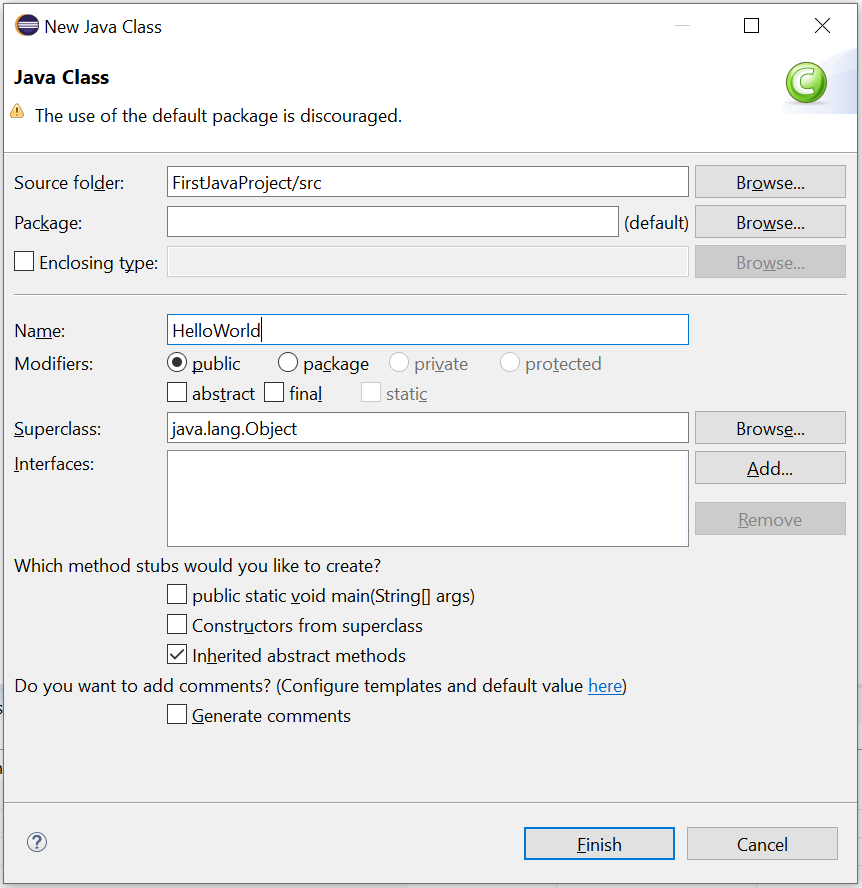


Then you will see this window:

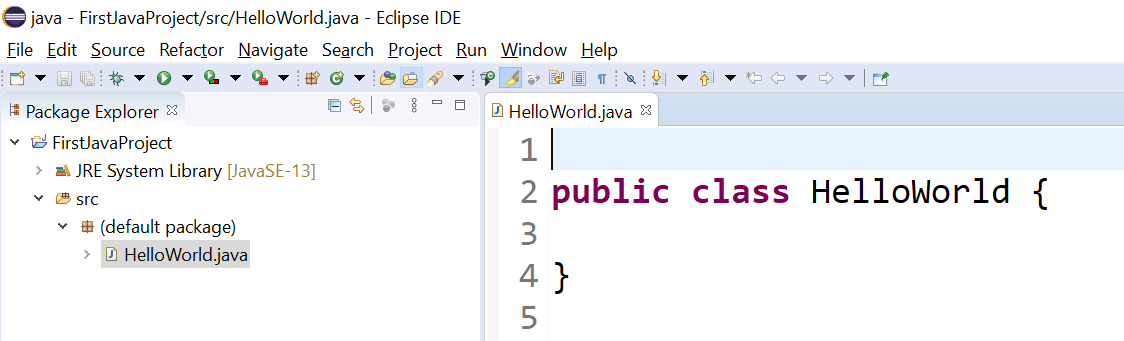


In the **Name** textbox above, you need to type in HelloWorld, and leave everything else untouched. Especially, please **leave the Package textbox empty**, because we use the default package. Don’t worry about the warning “The use of the default package is discouraged”. We just use the default package all throughout this course.

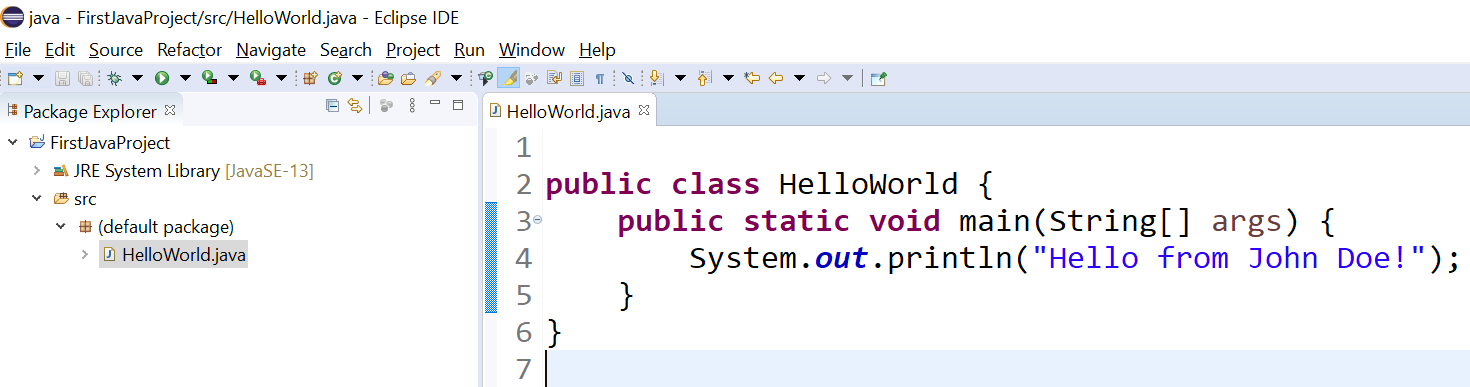
Java is case-sensitive, so please use the EXACT uppercase and lowercase letter combination for the class name: HelloWorld



Click Finish button and you will see the next page below:

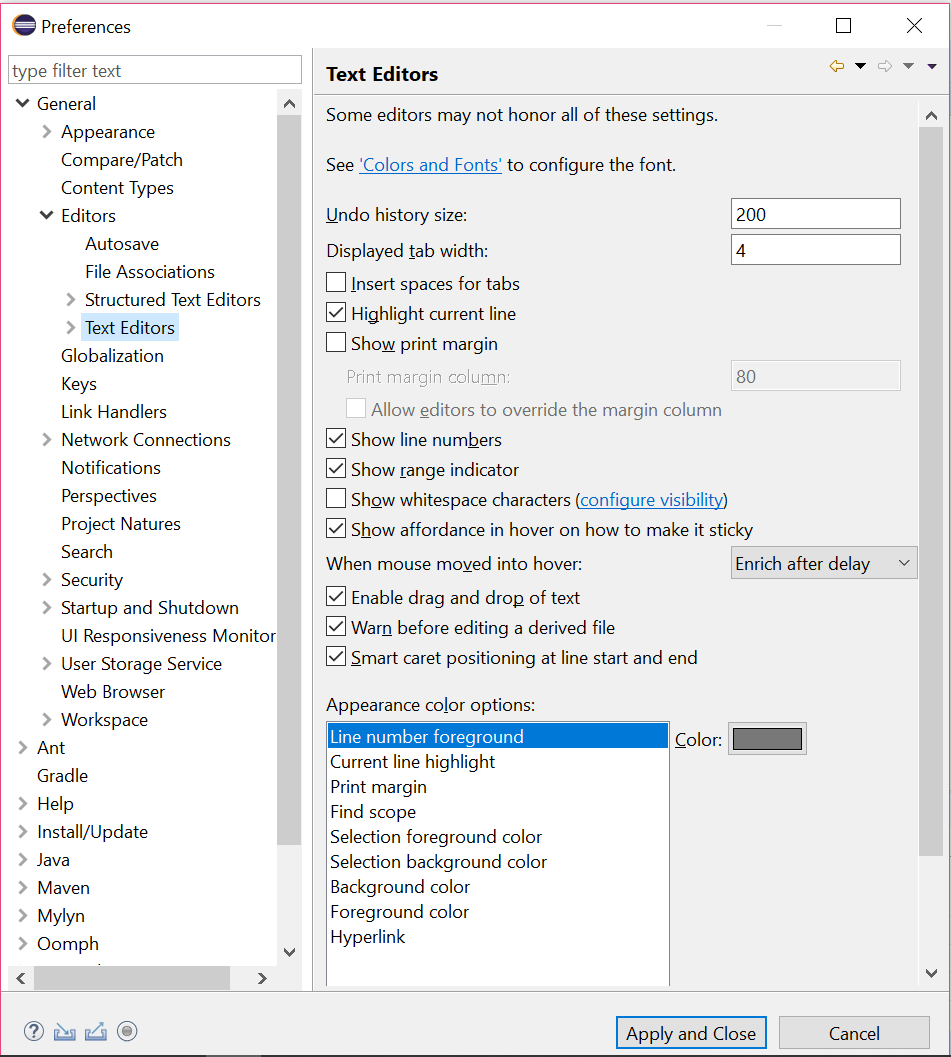


Then type in the source code below for in file HelloWorld.java, and after you finish typing, don’t forget to click the “save all” icon which is the 3rd icon from left in the menu icon bar.

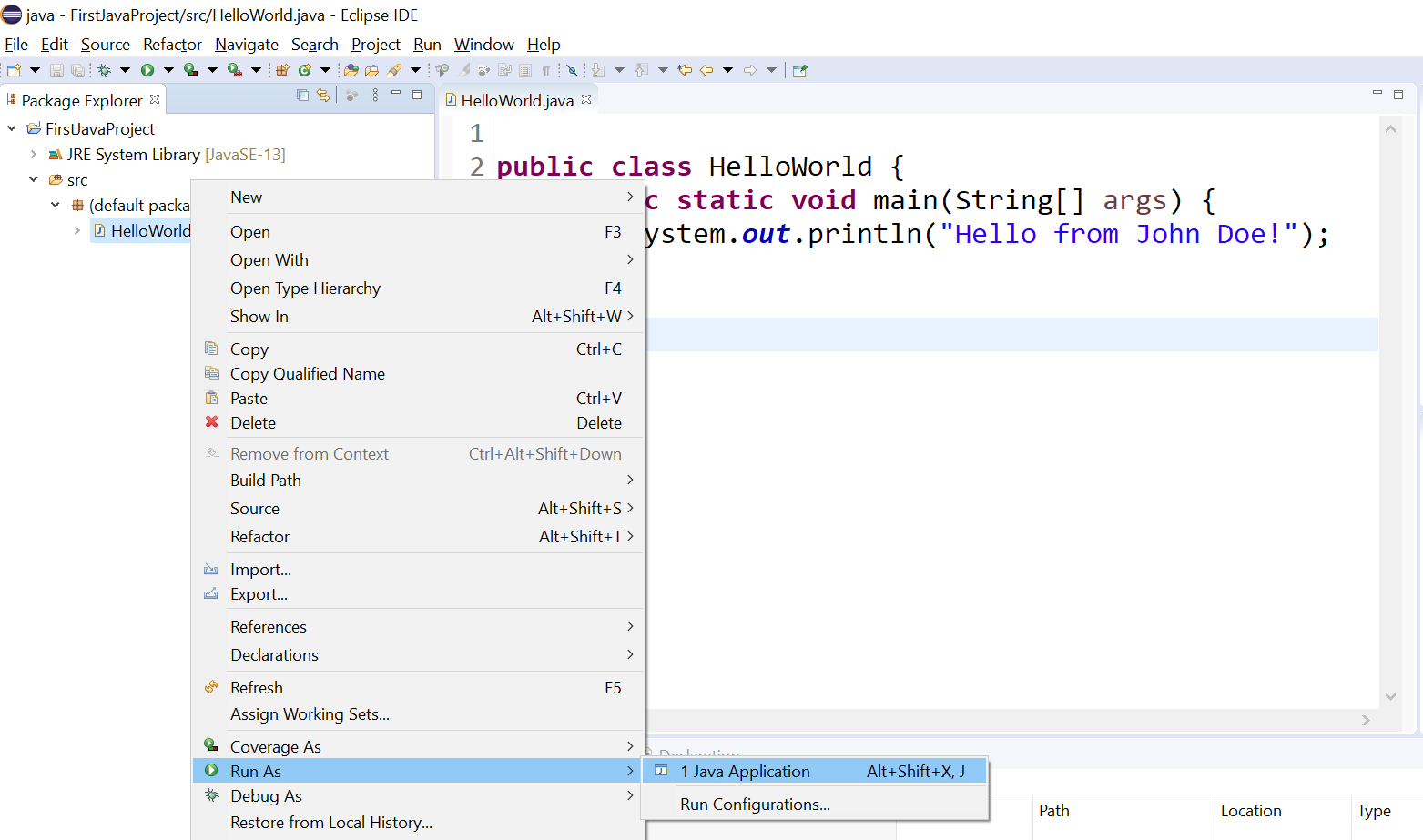


Again, Java is case sensitive, so please follow the source code above carefully and type it in. Currently, the place where you can exercise your freedom is to replace John Doe with your name.

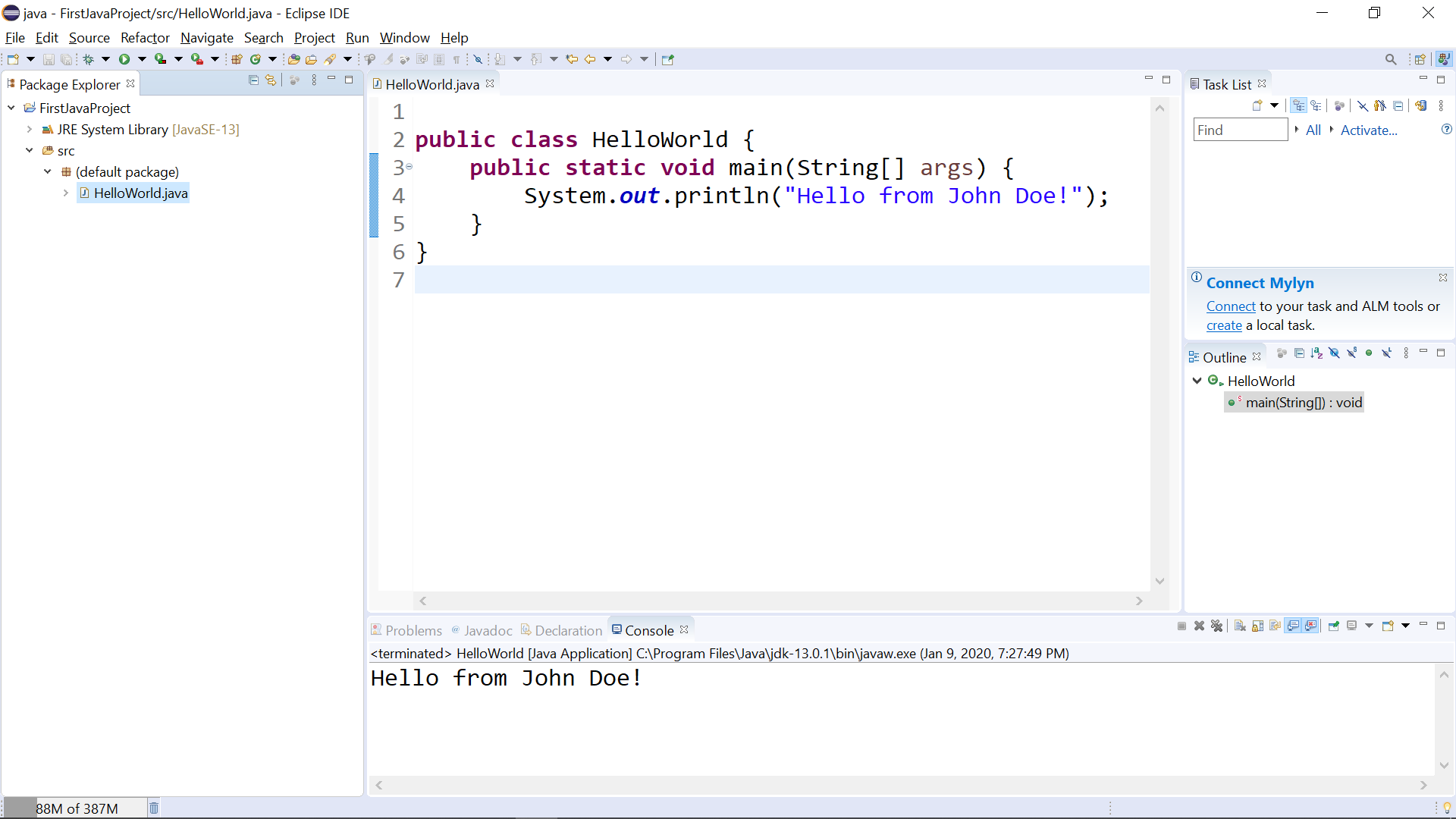
If the line number does not show up automatically in the Eclipse editor, it is recommended to show the line numbers, and you can do this by selecting Eclipse menu “**Windows**” > “**Preference**” > “**General**” > “**Editors**” > “**Text Editors**” , check on the “**Show line numbers**” option. Similar to this picture below:



Now it is time to run this application. **Right click** “HelloWorld.java” file in the “Project Explore” window on the left pane of the Eclipse, then choose menu item **Run As** --> **Java Application** , as indicated in the picture blow:

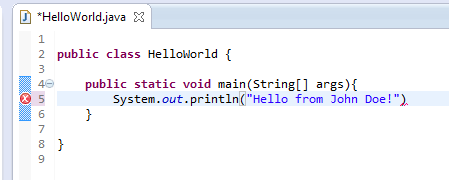


Once you click **Run As --> Java Application** , the program will run, and the result will be displayed in the console tap below the source code editor of eclipse, as indicated in this picture below:

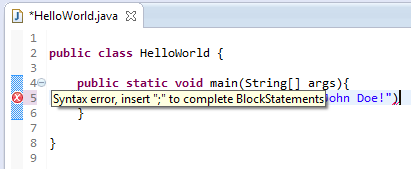


And you can see the “Hello from John Doe!” output item in the console tab at the lower portion of the Eclipse window. Now you are done with your first Java program’s edit and run!

**Trouble shooting:**

If you are not able to run the Java application, most likely you have errors in the Java source code. For example, I deliberately create an error in the Java source code, and here is what Eclipse looks like with this error:

See the little red check box on the left border of the source code editor? That indicates an error, and you have to clear out all these error indicators, before you can run the application. But how? You can look at the code more closely to see what is wrong with the code. And you can also move the mouse cursor to the red check box, and there will be a hint showing up, which can help you identify the error, as shown below:



Apparently, the error is that I forgot to put an semicolon ; symbol at the end of the output statement line.

If you have the red circle check mark in source code, you have to fix this compilation error before you proceed.

Sometimes there will be a **yellow** triangle with an **exclamation mark** and that is a warning in the Eclipse java compiler. You can ignore the warning and proceed to run the java program.

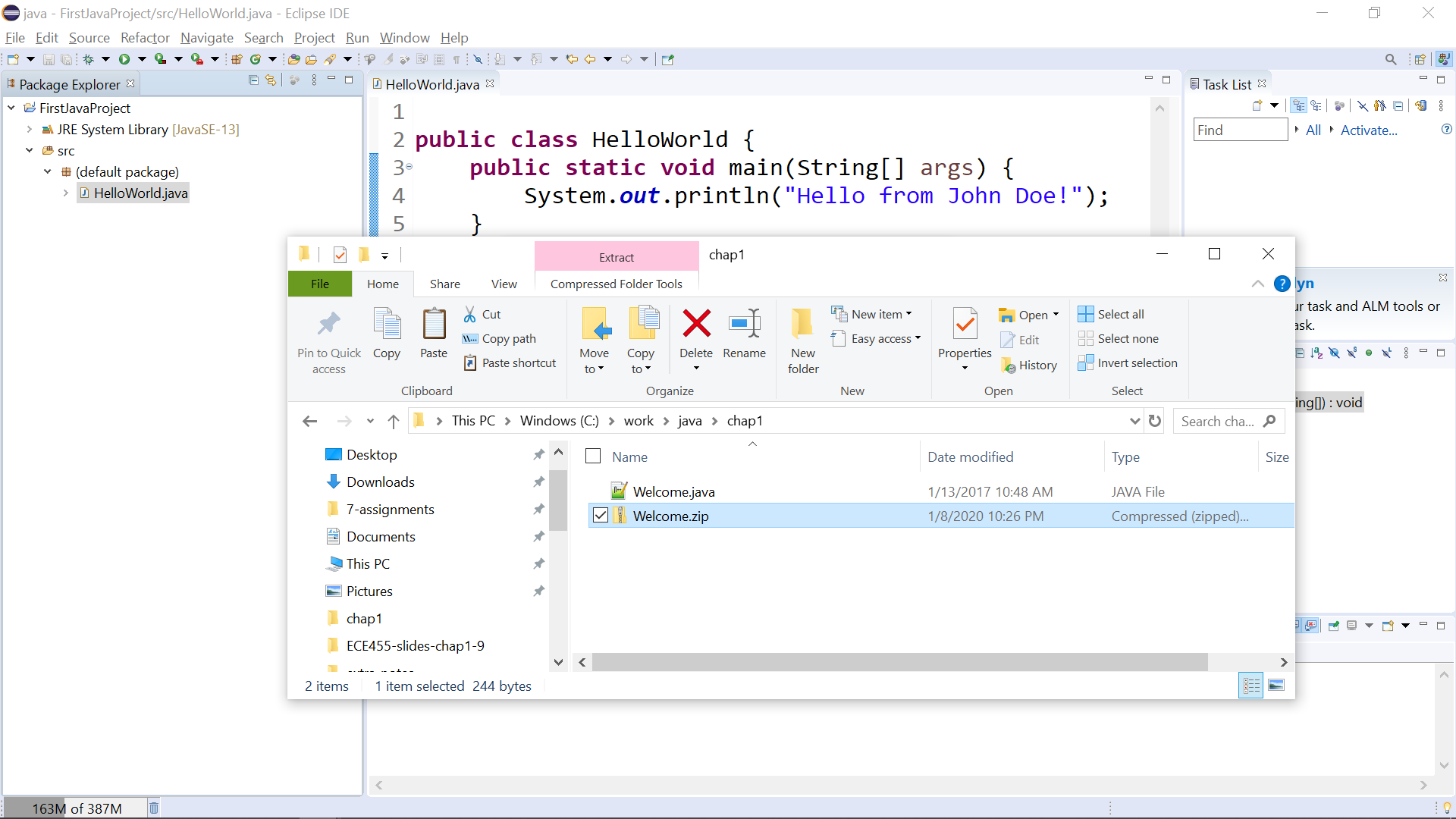
Till now, you should have a basic idea on how to run Java application in Eclipse.

**Now let’s demo how to include an existing java source code file into a java project,** such as the java source code provided in the zip file “**Welcome.zip**” in Moodle folder “chap 1”. You can create a new java project now, or you can use the existing project **FirstJavaProject**. In the demo below, I will use the existing project.

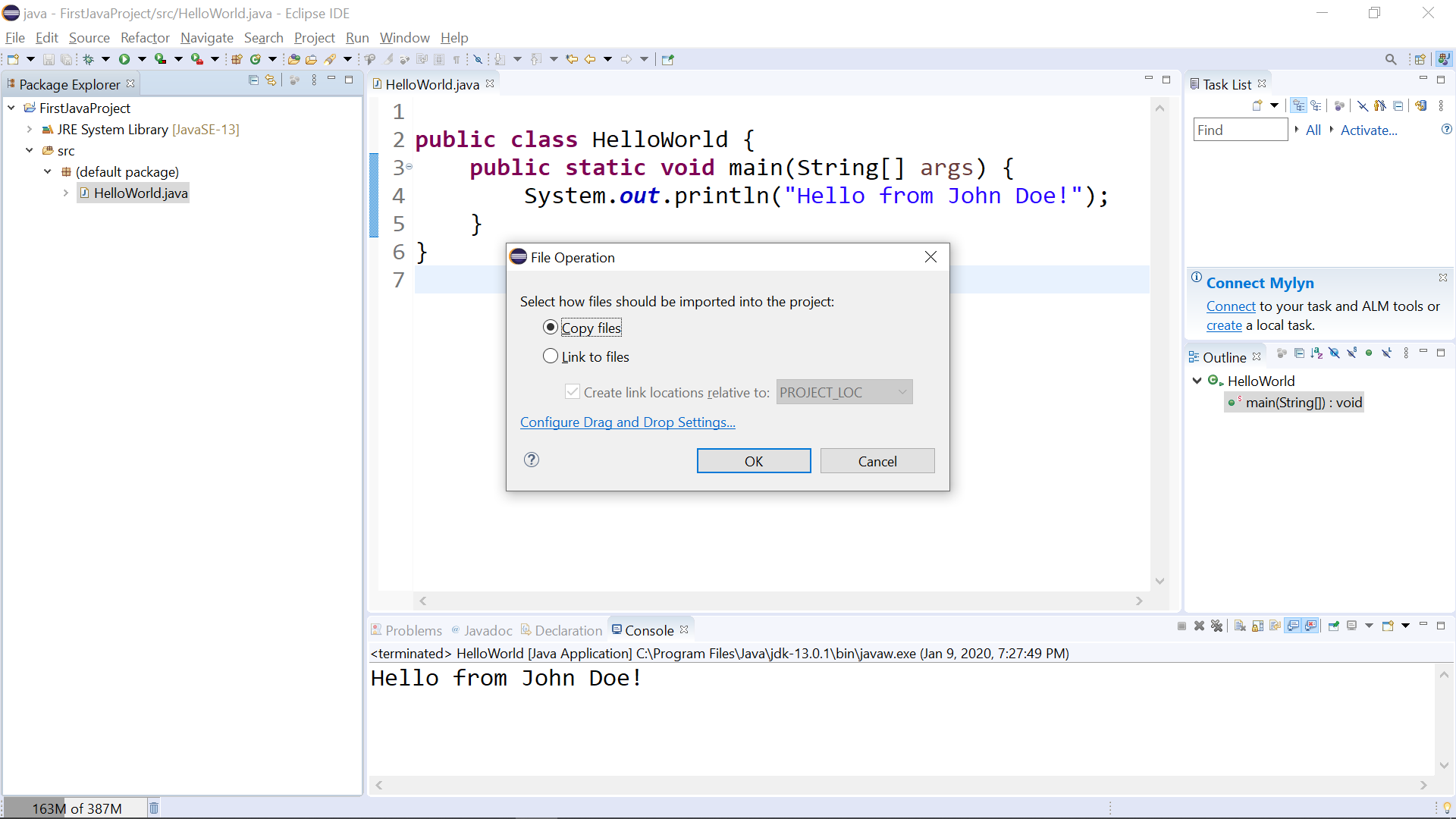
Download zip file “**Welcome.zip**” from Moodle “chap 1” folder. After unzip it, you will see the java source code file ***Welcome.java***. Assume this ***Welcome.java*** source code file is stored in this local folder c***:\work\java\chap1\ .*** Locate this file in the correct folder in “File Explorer” in Windows, or “Finder” in Mac.

Arrange the Eclipse window and File Explorer window, so that they show up in the same screen with File Explores window in the foreground, as shown below:

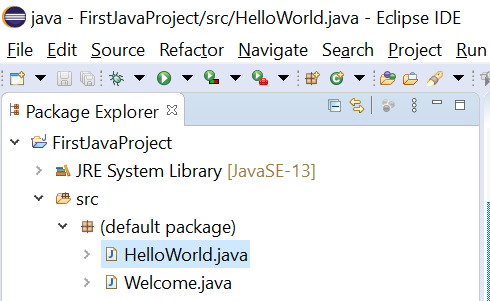
Then use mouse to select file **Welcome.java**, and then **drag and drop** this file from its file explorer folder into the **src** folder in “Project Explorer” tab in Eclipse.

**!!!!!** 

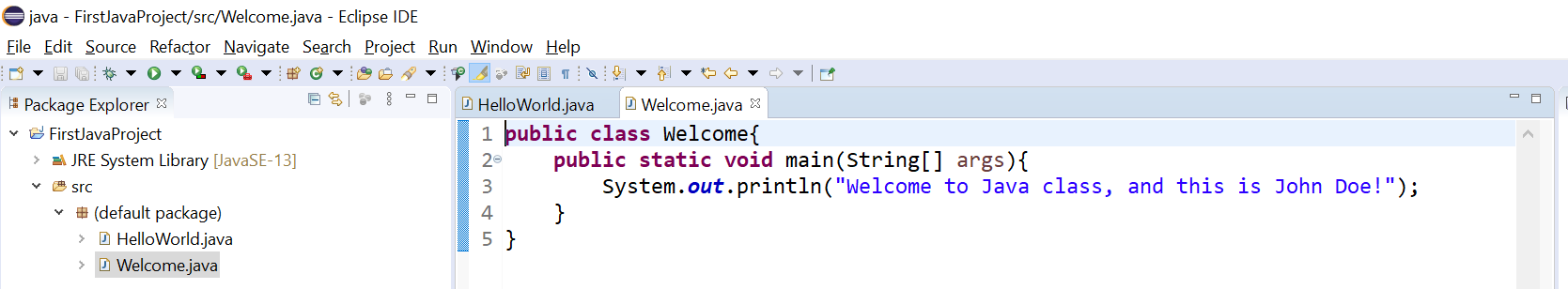
Then a popup window appears asking you whether you want to copy the java source code file to the **src** folder.



In the picture above, click the **OK** button to say yes. Now the Eclipse “Project Explorer” looks like this:

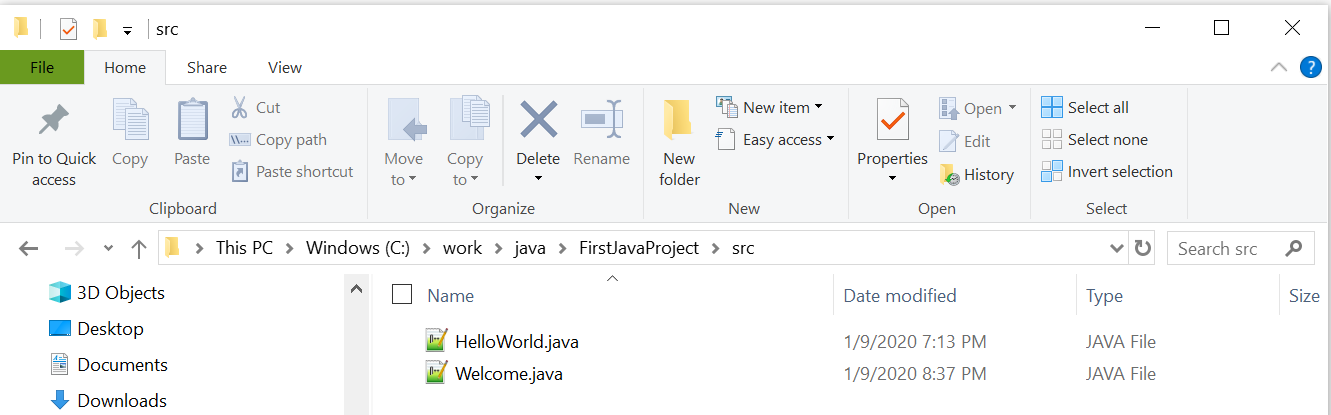


As you can see that, there are two Java files in the src folder, and the new one is the Welcome.java you copy from an existing file. Now you can left click the Welcome.java so that its source code is shown in the editor window, as indicated by the picture below:



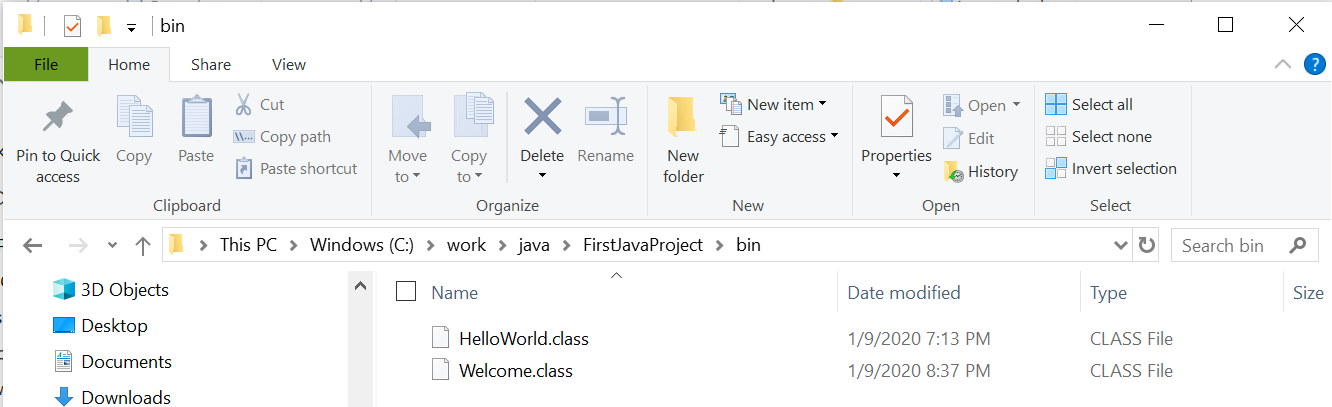
We use the same approach to run Welcome.java as running HelloWorld.java: right click **Welcome.java** in the **src** folder, and then in the context menu, selecting **Run As -> Java Application**.

Before we finish this demo, you need to locate the eclipse project folder where the java source code locates.



The above picture is the source file location in my computer. Pay attention to the folder where the two java source code file locates, and it is **C:\work\java\FirstJavaProject\src** , why is that? Because **C:\work\java** is my workspace folder, and **FirstJavaProject** is my project folder, and **src** is where the source code locate. Similarly, if you create another java project named **MyWebProject**, then the java source code will locate in **C:\work\java\MyWebProject\src ,** so on and so forth. Or if you switch to a different workspace, say, d:\company\client, and the project name is finance, then the source code is in **d:\company\client**\**finance**\**src** .

Behind the scene, Eclipse invokes the JDK that we installed to compile the java source code .java file into .class file automatically. Here is where the .class files locates, and it is **C:\work\java\FirstJavaProject\bin** in our case.



If in your Windows File Explorer, you cannot see the file extension .java or .class, then the file extension is hidden by the Windows Explorer. On how to show file extensions, refer to these links:

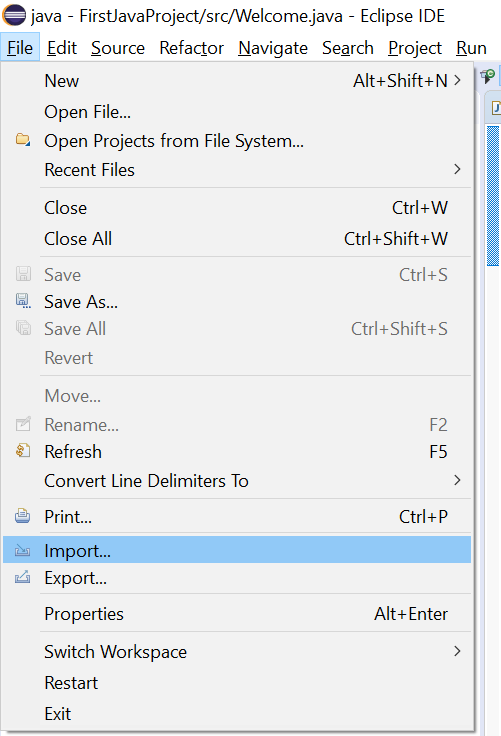
<http://kb.winzip.com/kb/entry/26/> is for Windows

<https://www.mactrast.com/2018/01/show-file-extensions-macos-finder/> is for Mac

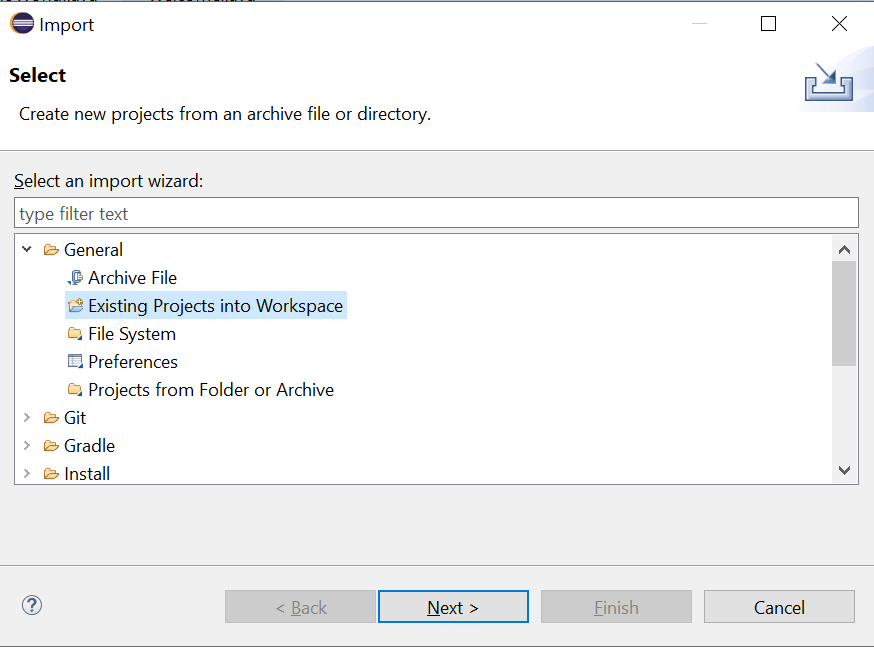
Now we can demo how to import an existing Eclipse project into Eclipse IDE.

Suppose you have a downloaded zip file that contains a complete Eclipse java project folder, and then you have unzip this zip file into an folder. Let’s assume you this Eclipse project folder is c:\test\JohnDoeHw5, now I am going to import this project into the Eclipse IDE.

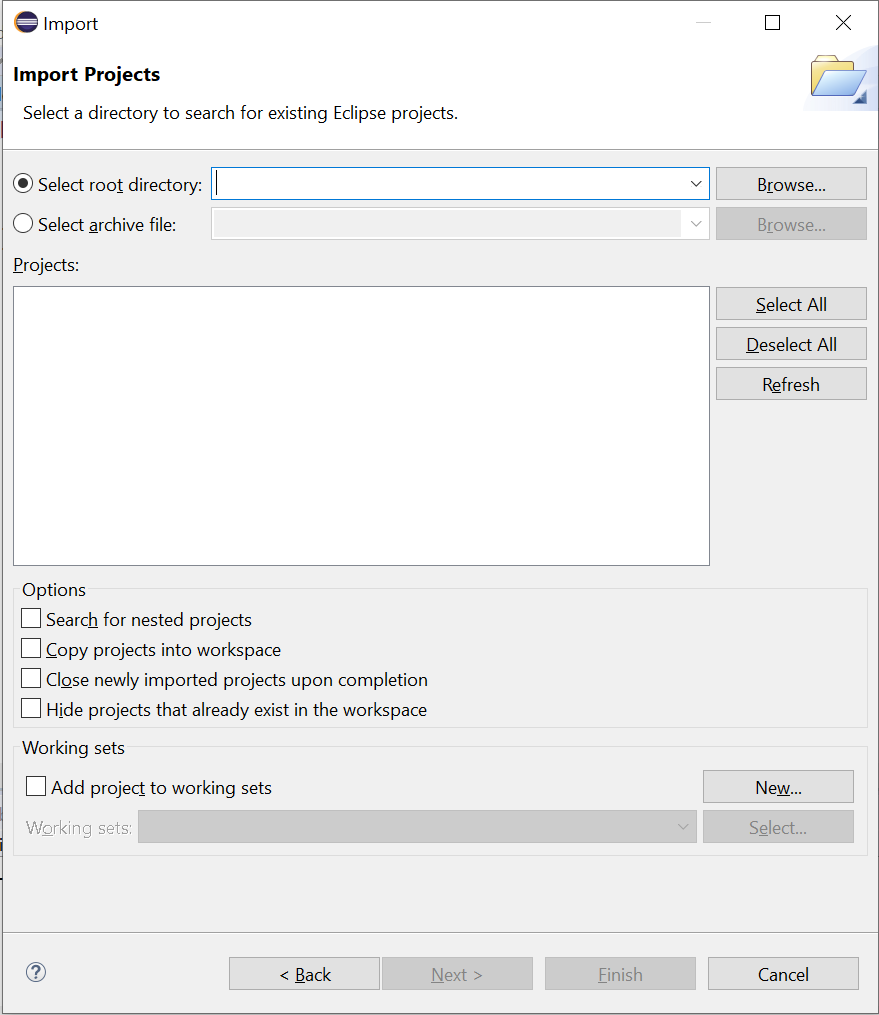
In Eclipse IDE, click menu item File -- > Import



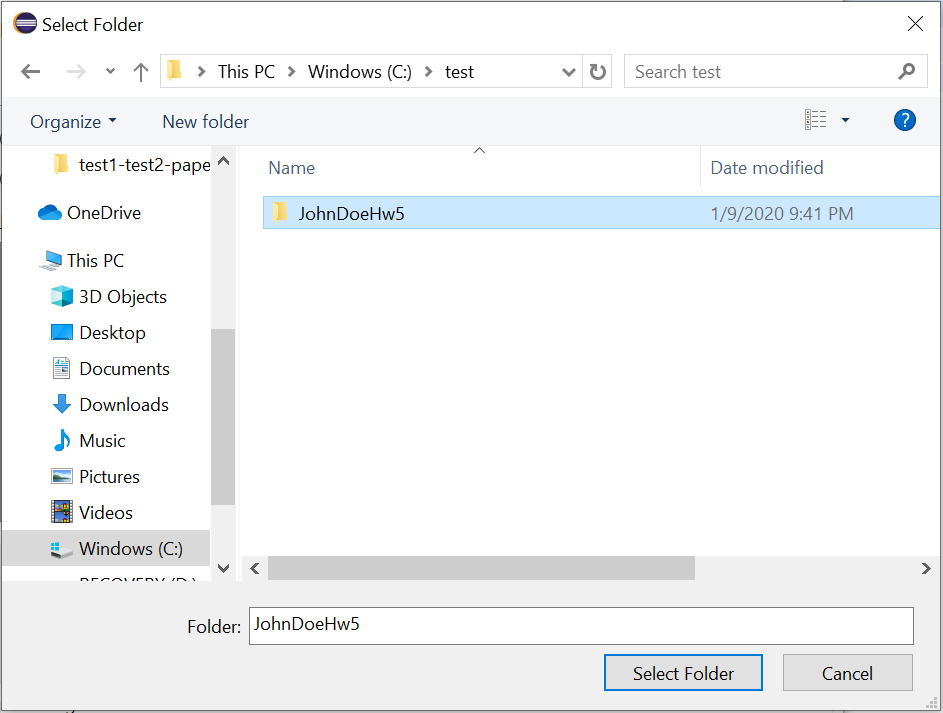
Then in the dialog window, expand “General”, and choose “Existing Projects into Workspace”, then click Next.



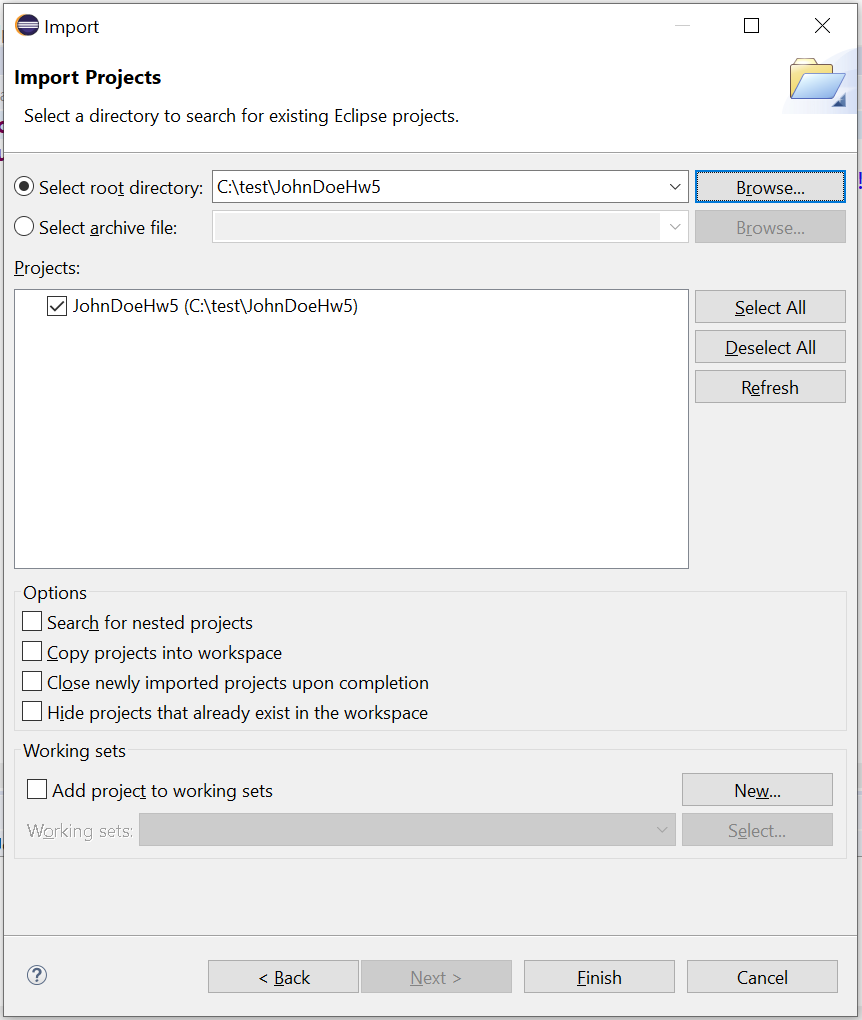
Then you will see this screen below.



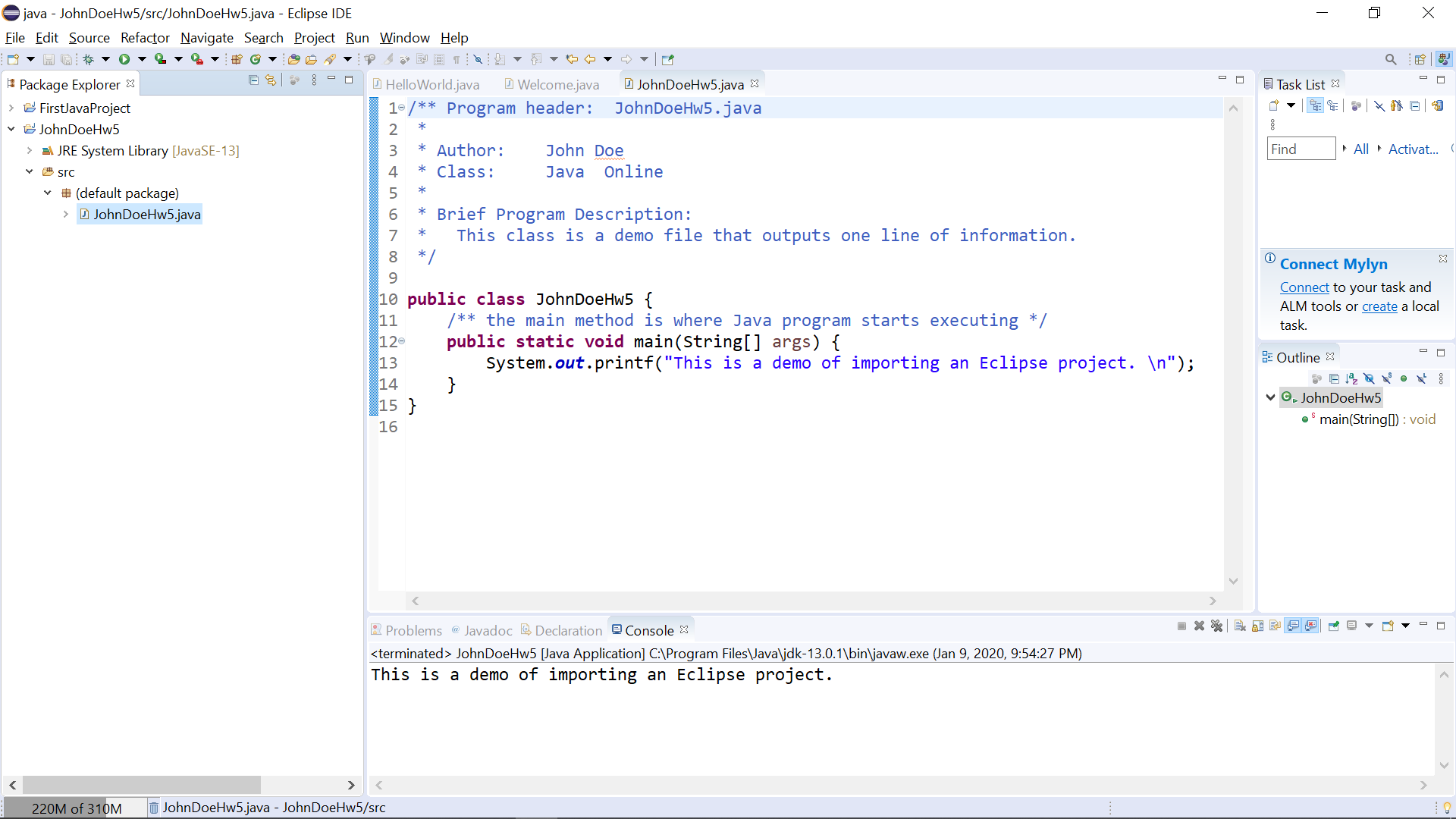
Use the Browse button to browse to the parent folder where the Eclipse project folder is, in this example, it is the C:\test\ folder, and the select the c:\test\JohnDoeHw5 folder, like this:



and then click the “Select Folder” button, and this will show up:



Then click the Finish button, and the Eclipse will be imported. You can run this project use the same method as we discuss earlier for Eclipse project “FirstJavaProject”:



In summary, in this document, we have demonstrated the following:

* how to create a new Java project in Eclipse
* how to create a new Java source code file
* how to type in source in the source code editor
* how to debug (correct any error) and run a Java application
* how to locate the .java source code file and the .class file
* how to include an existing Java source code file into a project
* how to import an existing Eclipse Java project folder

Below is another link that explains how to use Eclipse to run a simple Java program, and you can use it as a reference. If you successfully finish all the steps above, the content in the link below will be familiar to you.

<http://www.wikihow.com/Start-and-Compile-a-Short-Java-Program-in-Eclipse>