# Lab 1: First Java Project in Eclipse IDE

Objective(s):

* Create a simple Java program which reads the user’s name and outputs a greeting
* Learn to create Java projects, Java class files and work with Eclipse

# 1 Creating your first project

You should be at the Welcome page in Eclipse. Choose “Create a new Java project” on the Welcome page. You can also navigate to File > New… > Java Project **or** navigate to the on the task bar (directly under File).

Graphical user interface, text, website

Description automatically generated

This will open another window that will let you create a new project. Name the project “FirstProject” The default location for projects is in the workspace directory you specified earlier. Deselect the module-info.java option. Keep the other options which are selected by default and choose “Finish”

You will see the Package Explorer on the left. The center window is the editor. The outline on the right will give you Outline of the file you have open in the editor. The bottom displays any compiler warnings and errors. It will later host the Console window as well.

A screenshot of a computer

Description automatically generated

In the Package Explorer, expand your project using the arrow on the left. You will see a src folder and JRE System Library. Expand the src folder. You will find that you do not currently have any source files. Let’s add one. Navigate to File > New > Class or open the wizard selector window and choose Java > Class. This will open a window to create a new Java class.

Name the class Greeter and choose the public modifier and none or final.

Greeter will be our main class. You can choose the option to create the method stub “public static void main(String[] args).” If you choose not to, you will need to type out the method later.

A screenshot of a computer

Description automatically generated with medium confidence

You will receive a warning about using the default package. You can ignore this for now. Choose Finish.

Greeter.java will now be open in the editor. The outline on the right will have updated to include the name of the class, Greeter, and the methods in the class, main(String[]).

If you don’t see the editor, try closing the Welcome page. If it is still not active, navigate to  on the top-right of Eclipse and open the Java perspective.  should now be available on the top-right so you can quickly navigate between perspectives.

Add the following code to the main method.



A screenshot of a computer

Description automatically generated

Notice that Scanner is now underlined red. The Problems window at the bottom also shows the errors “Scanner cannot be resolved to a type.” Hover your mouse cursor over either Scanner in the editor. A dropdown will appear showing the same error in the Problems window and there will be a list of quick fixes. Any of these fixes will help remedy the error, but we specifically want to use the Scanner class from the java.util package. Click on the fix “Import Scanner (java.util).”

An import statement will be added to the top of the file, and we should no longer have the error.

Now we can run the program. Navigate to Run > Run.

A screenshot of a computer

Description automatically generated

The program will now be running in the Console window on the bottom. Try entering your name as input and press Enter.

The program should print the greeting and echo your name.

Let’s add another method to the Greeter class.

This method is called greet. It takes a String called name as a parameter and prints the greeting message.

A screenshot of a computer

Description automatically generated

Change the greet method to be non-static i.e., delete the static keyword from the line. You should get an error on this line now. Fix the error by hovering over the underlined text and choosing the fix.

Try running the code again. It should run the same as before.

# 2 Command line compilation

You will probably work with an IDE when you are working on any programming projects, but an IDE is just a text editor packaged together with useful tools for development. It’s useful to know how to build your programs without the use of an IDE, specifically with a terminal. We can use Eclipse for this as well.

Right-click on your FirstProject and navigate to Show In > Terminal.

This will open your project directory inside a terminal window where the Console and Problems are located. If you’re on Windows, it probably opened inside a command prompt terminal. Use dir to list the contents of the directory. Linux and macOS can use the command ls.

Use the command cd to move into the src folder if you are not already there e..This folder contains the Java source files, it should only contain “Greeter.java” now.

The compile command for Java programs is javac <list of src files>. Run the command javac Greeter.java.

Run dir or ls. You should see now that there is a new .class file in the src folder. Class files are generated from compilation. You can run your Java program by using the java command and listing the main class.

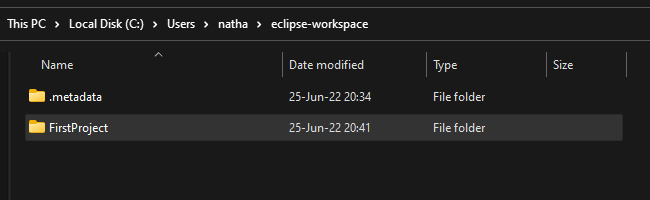
Try running your program now using the command java Greeter.

It should run similarly to when it was run through Eclipse’s run button.

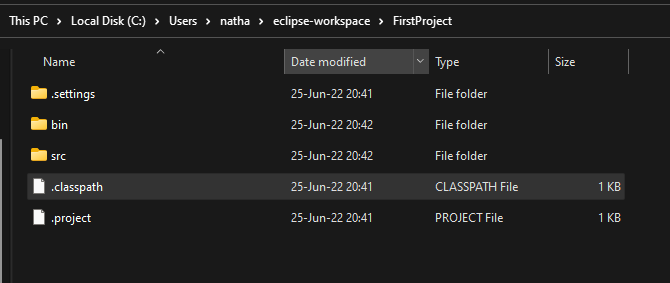
# 3 Submitting programs for labs and assignments

You will be asked to submit your Java programs for grading. The purpose of this section is to show you how you can prepare your files for submission.

Right-click on your FirstProject and navigate to Show In > System Explorer. This should show your project directory in your computer’s file explorer.



Open the folder with the FirstProject name. You should find contents like this. We’ll go through what these files are.



* The .project file describes the project.
  + It will allow the grader to correctly recreate the project settings that you developed with.
  + You should include this file in your submissions.
* The .classpath file includes the settings necessary to run your program from within Eclipse.
  + It specifies the src directory as the location of source files, and the bin directory as the location of output class files.
  + You should include this file in your submissions.
* The src folder holds all the source code as .java files.
  + You should include the entire directory in your submissions.
* The bin folder contains all the class files as .class files.
  + These files are generated from compiling the java files.
  + You should NOT include this directory in your submissions.
* The .settings folder contains .prefs preference files which describe the preferences of your project.
  + The information in this folder is for project specific settings. These can be used to ensure your project is built using the compiler settings you want when it is submitted.
  + This directory can be optionally included in your submission.
  + If you do not include it, then we will generate the settings using the LTS version of Java in good faith.

You will be submitting your projects as .zip files.



Highlight all the required files and folders using ctrl + left-click or dragging the selection box over them.

* **On Windows**, right-click on the files and choose “Compress to zip file.” If you don’t have this option, you will need to right-click and navigate to Send to > Compressed (zipped) folder.
* **On Linux**, right-click on the files and choose Compress...
* **On Mac**, Command + click each required file and folder, then control-click the highlighted files and choose Compress from the shortcut menu.

The naming convention for submissions will be “lab/assg#\_firstName\_lastName” as shown above.

So, in this case name your zip file “lab1\_<firstName\_lastName>”.

Mac and Linux users:

By default, your system’s file explorer may hide dot files like the .classpath, .project, and .settings folder. Mac users can show dot files in Finder using the shortcut (shift + command + . (dot)). Settings to show hidden files should also exist for Linux users, but their location and any shortcuts depend on the file manager you are using.

Submission notes for Task 1:

* Submit the entire project folder which includes all files from your project including src folder containing .java files, .classpath file, and .project file. The .settings folder containing .pref files can optionally be included.
* Zip the project folder and name it as “Lab1\_firstName\_lastName”, where the cslogin is your   
  login ID for the computers at the Department of Computer Science at ODU.
* Submit the zipped file to the respective Canvas link.
* **Instructions for Mac users**
  + By default, Finder hides dot files like .project, .classpath, and the .settings folder. You can use the shortcut (shift + command + . (dot)) to show hidden files in Finder.
* **Instructions for Linux users**
  + By default, your file manager may hide dot files like .project, .classpath, and the .settings folder. Your file manager should have a setting that will show hidden files. It’s location and any shortcuts depend on the file manager you are using.

|  |
| --- |
| Grading Rubric |
| Task 1:   * Compilation (5) * Submission of required files (5) |