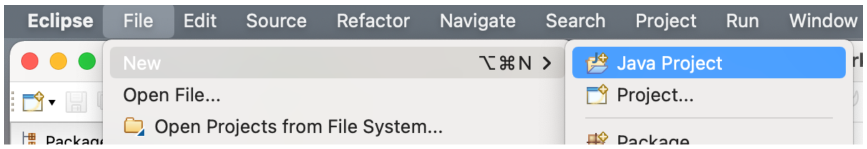
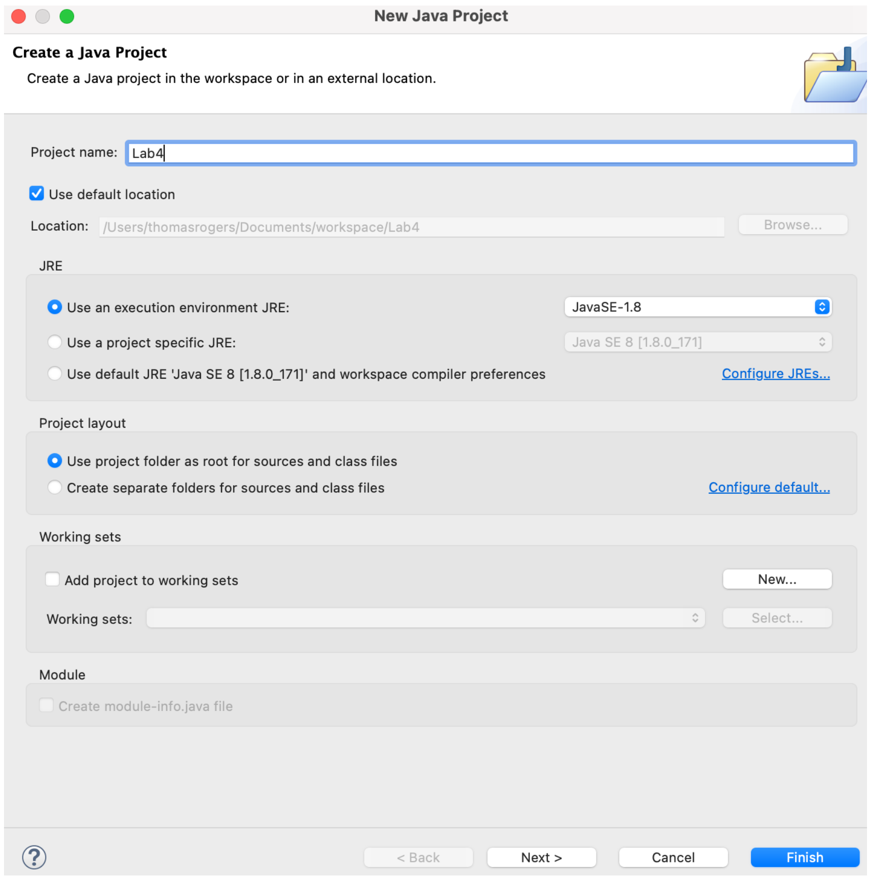
Millersville University Eclipse Best Practices

This document gives a brief overview of best practices when using the Eclipse interactive development environment (IDE) for developing CSCI 161 and 162 programs.

1. **Creating a Project:** Within Eclipse select *File->New->Java Project* as so:



Once the "New Java Project" form comes up, enter the lab or assignment name as shown below, when done click the "Finish" button (see helpful notes after the form):



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: Lab4

☑ Use default location

Location: /Users/thomasrogers/Documents/workspace/Lab4 Browse...

JRE

● Use an execution environment JRE: JavaSE-1.8

○ Use a project specific JRE: Java SE 8 [1.8.0_171]

○ Use default JRE 'Java SE 8 [1.8.0_171]' and workspace compiler preferences Configure JREs...

Project layout

● Use project folder as root for sources and class files

○ Create separate folders for sources and class files Configure default...

Working sets

☐ Add project to working sets New...

Working sets: Select...

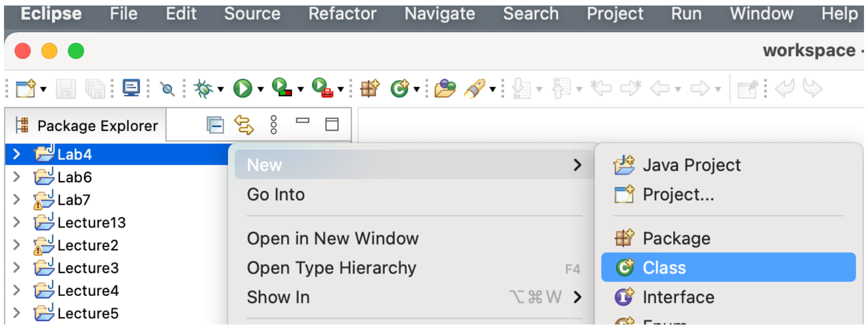
Module

☐ Create module-info.java file

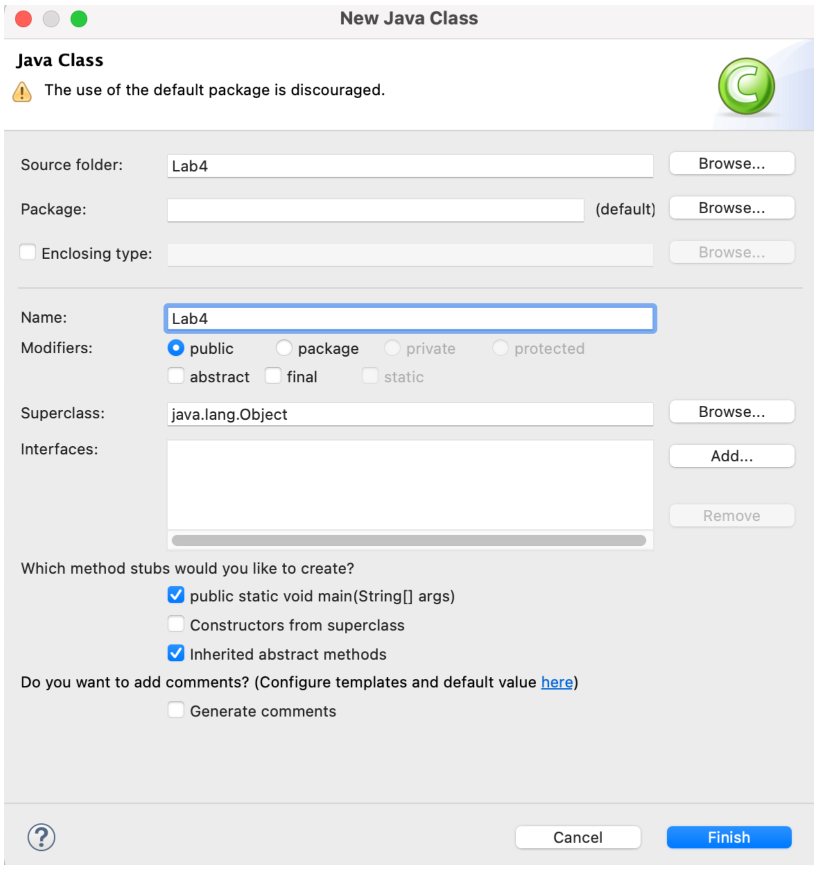
< Back Next > Cancel Finish

- Your project name *should not* include spaces and should start with an uppercase letter (e.g. Lab4, not "Lab 4", not lab4, etc.).
- In the "Project Layout" section click the "Use project folder as root for sources and class files" radio button so that your Java file created is *not* in a "src" subfolder (easier to find).
- By default, the "Create module-info.java" file checkbox should be disabled. If it is not disabled then make sure NOT to have it checked.

2. **Adding a Program Class to a Java Project:** Once you have a Java Project created you must add your program class (the .java file with your main method) to the project. The easiest way to do this is to select your new project from the "Package Explorer" pane on the left then right-mouse-click on it in order to bring up the menu choices and selecting New->Class as shown below:



The "New Java Class" form should come up and you then enter the Name which is typically the same as the Project name. Then you should check the "public static void main(String[] args)" checkbox so that the main method stub is created, then click "Finish":



New Java Class

Java Class

The use of the default package is discouraged.

Source folder: Lab4

Package: (default)

Enclosing type:

Name: Lab4

Modifiers: [✓ public] package private protected

abstract final static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

[✓] public static void main(String[] args)

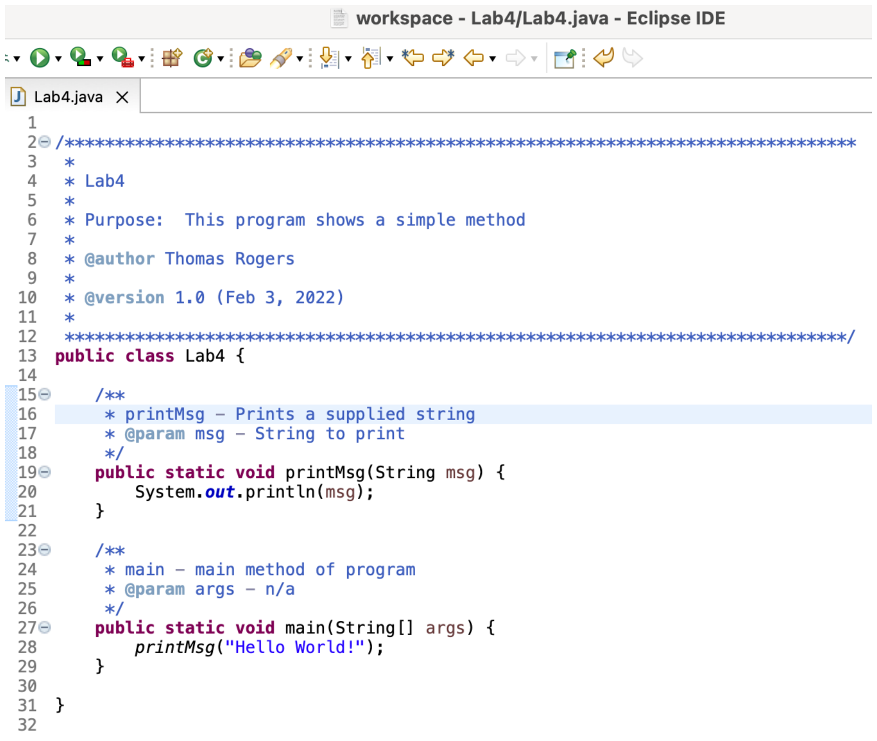
[ ] Constructors from superclass

[✓] Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

[ ] Generate comments

3. **Coding Conventions / Miscellaneous -** What follows are a number of coding conventions you should follow based on the simple program shown:



```

/******************************************************************************
 *
 * Lab4
 *
 * Purpose:  This program shows a simple method
 *
 * @author Thomas Rogers
 *
 * @version 1.0 (Feb 3, 2022)
 *
 ******************************************************************************/
public class Lab4 {

    /**
     * printMsg - Prints a supplied string
     * @param msg - String to print
     */
    public static void printMsg(String msg) {
        System.out.println(msg);
    }

    /**
     * main - main method of program
     * @param args - n/a
     */
    public static void main(String[] args) {
        printMsg("Hello World!");
    }

}

```

i. **Comment at top of the program** should include the assignment name, purpose, @author and @version information as shown.

ii. **Helper methods should be near the top of the module** and the main method should be last.

iii. **Add a comment block before each method** ideally in Javadoc format.
Helpful Hint: If you write the method first with all its parameters then add the comment above by typing in /** then hitting enter the IDE will fill in much of the comment block and parameters for you.

iv. **Method names** should be camel case (e.g. printMsg, not PrintMsg, not print_msg).

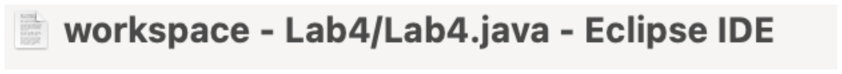
v. **Remove auto-generated comments** including:
```
// TODO Auto-generated method stub
```

vi. **Maintain proper indentation.** Note how the program above is indented and make sure of the following:

   1. Methods start one tab space over from the main class.
   2. Starting (open) curly braces should be at the end of lines and after you type then simply hit enter and Eclipse will insert a tab on the next line as should be included (do not "fight" the IDE).

3. Ending (closing) curly braces should line up with the start of the statement that included its matching open brace.
4. Statements within curly braces are indented one tab (see 2 above).

vii. **Project Folder, Program file Location** – You should always be able to find your program file on your system because at the top middle of the IDE window the title bar shows the workspace folder (workspace), the project/folder name (Lab4) and name of the program file currently active (Lab4.java) as so:


workspace - Lab4/Lab4.java - Eclipse IDE