

1.00/1.001/1.002 - Spring 2012
Introduction to Computers and Engineering Problem Solving
Problem Set 0
Due: 12 noon, Monday February 13, 2011

All 1.00/1.001/1.002 students must complete Problem Set 0, which introduces 1.00/1.001/1.002 software and procedures. Problem Set 0 will not be graded but must be completed to make sure your software installation and account on the course website function properly. If you do not understand or cannot complete any part of the problem set, please speak to a 1.00/1.001/1.002 staff member as soon as possible. Instructors, TAs and their email addresses are on the 1.00 Website as well as in the printed materials handed out at the first class.

Problem 1: Read Initial Handouts

All the initial handouts include

- Syllabus/FAQ
- Lecture Schedule
- Academic Honesty Guidelines
- How to Install Java
- How to Install Eclipse
- How to Submit Problem Sets

You are required to read the Guidelines on Academic Honesty. After reading them, you should understand that you can receive an F for your final grade in 1.00 by committing an academic honesty violation. In order to take the class, you must sign and return to the 1.00 staff a hard copy of the Guidelines on Academic Honesty. Those forms are handed out at the first lecture, but ask a TA if you need another form.

Problem 2: Sign Up for Recitation Section

All 1.00/1.001/1.002 students are required to sign up for a recitation section. Section assignments are on a first come first served basis, so please sign up as soon as possible, after 8pm on Wednesday February 8. For detailed section assignments, please go to the course website.

Recitations will last one hour and are held only on Mondays and Tuesdays. There is a maximum capacity of 12 students in each section.

Because recitation capacity is limited, sign up as soon as possible. Once you have joined a recitation section, you will be able to submit and download homework in that section.

You must attend the first class of recitation section you signed up, which is either on Monday, February 13 or Tuesday, February 14. If you decide to switch sections, you will only be able to do so through Wednesday, February 22.

Problem 3: Introduction to Eclipse

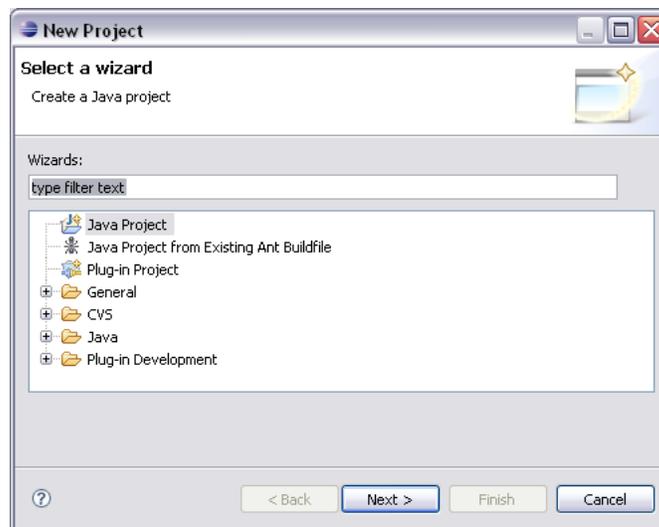
Please install Java, *then* Eclipse. For details, refer to the „How to“ documents, which can be found on the 1.00 website. Contact your TA for any installation problem.

Step 1 - Start Eclipse

If you followed the „How to“, you should have an Eclipse shortcut on your desktop. Double-click this icon and an Eclipse splash screen should appear while the program is loading. Or, you may double click on the Eclipse executable file. This file is located in the folder where you installed Eclipse. (You can create a shortcut to the Eclipse executable by right-clicking and dragging it to your desktop, where it will appear as an icon.)

Step 2 - Creating a new project

In Eclipse, projects are used in order to organize the files in a program. The project will include all the needed files for the program to run. To create a new project, choose **File** → **New** → **Project**. A „New Project“ screen will appear:



Courtesy of The Eclipse Foundation. Used with permission.

Make sure 'Java Project' is highlighted. After clicking 'Next', a 'New Java Project' screen will appear:



Courtesy of The Eclipse Foundation. Used with permission.

Under „Project name“, type „ProblemSet0“.

We **strongly recommend** that you choose the “Create project in workspace” option, which will place your work in the folder:

For Windows 7/ Vista: **C:\Users\\workspace**

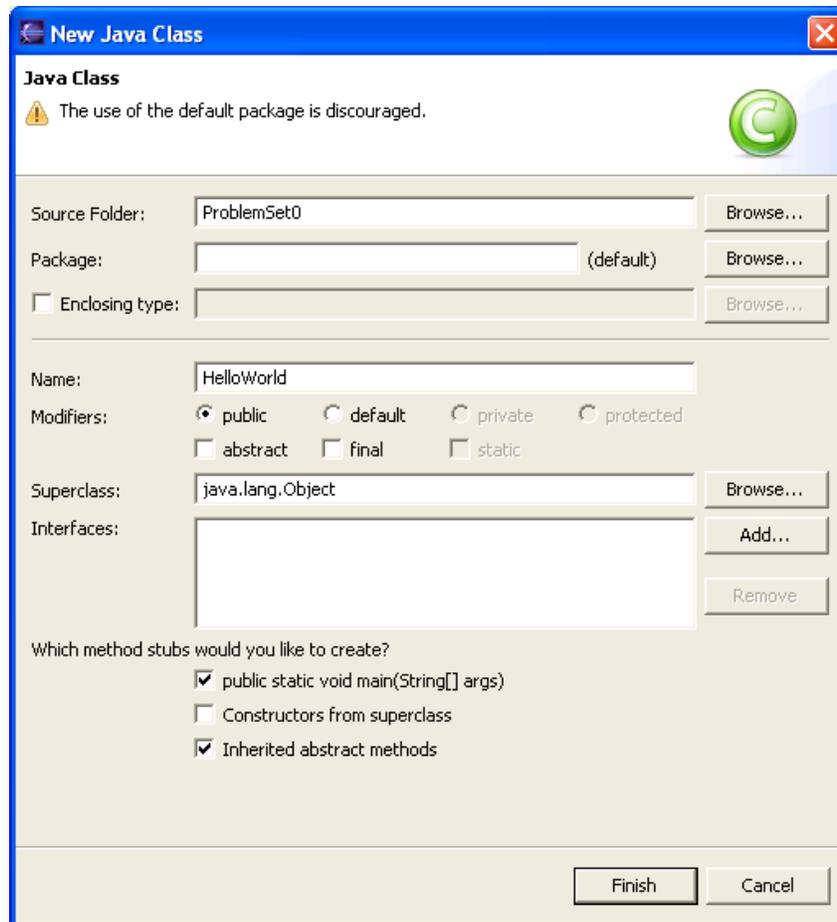
For Mac: **/Users/<username>/Documents/workspace/**

For Linux: **/home/<username>/Documents/workspace/**

When you are done, click „Finish“. If a dialog box appears, check the “Do not show this message again” box and hit „Yes“.

Step 3 - Creating your first class

Select **File** → **New** → **Class**, or click on the „NEW“ icon, , then select class and hit „Next“. On the „New Java Class“ screen, type the name of the class, HelloWorld, in the Name field and check the „public static void main(String[] args)“. Don't worry about other options on the screen for now.



Courtesy of The Eclipse Foundation. Used with permission.

Step 4 - HelloWorld.java

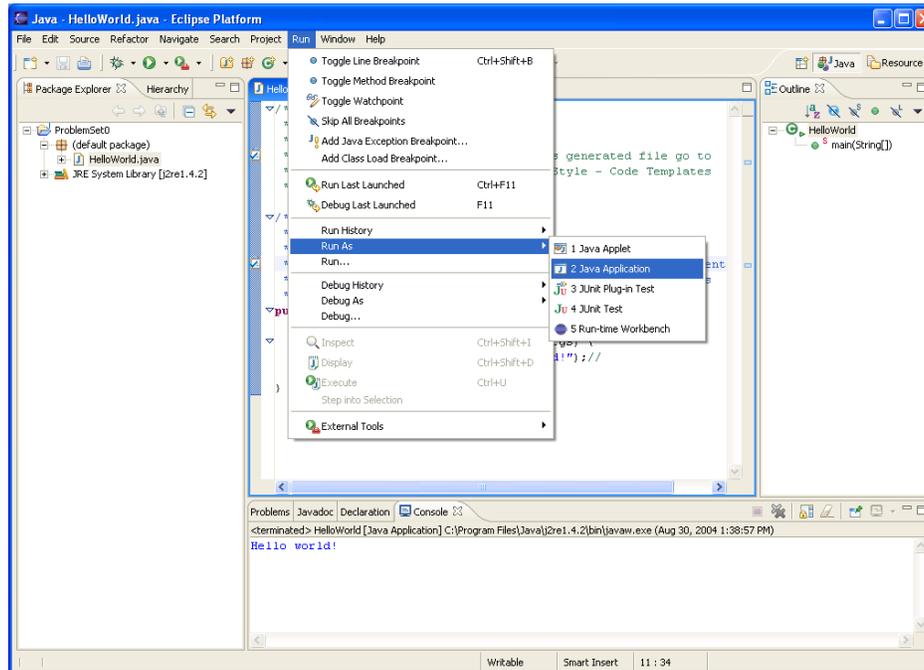
Eclipse will create and open a new file named „HelloWorld.java“. Add one line of Java code to the `main()` method so that it looks like:

```
public static void main(String[] args)
{
    system.out.println("Hello world!"); // Add this line
}
```

Save the edited version of `HelloWorld.java` by selecting `File → Save`, or by clicking on the save icon. **Eclipse automatically compiles the file when it is saved.** As you write longer programs in future problem sets, remember to save each file within the program as you write or edit it; otherwise other files that reference it will appear not to see it, or not to see your latest changes.

Step 5 - Running the HelloWorld class

- 1) From the Eclipse menu, Select **Run** → **Run As** → **Java Application**.
- 2) Because we've edited the program, Eclipse prompts us to save our changes before it runs the program. Click Ok.
- 3) The Console displays our program output.



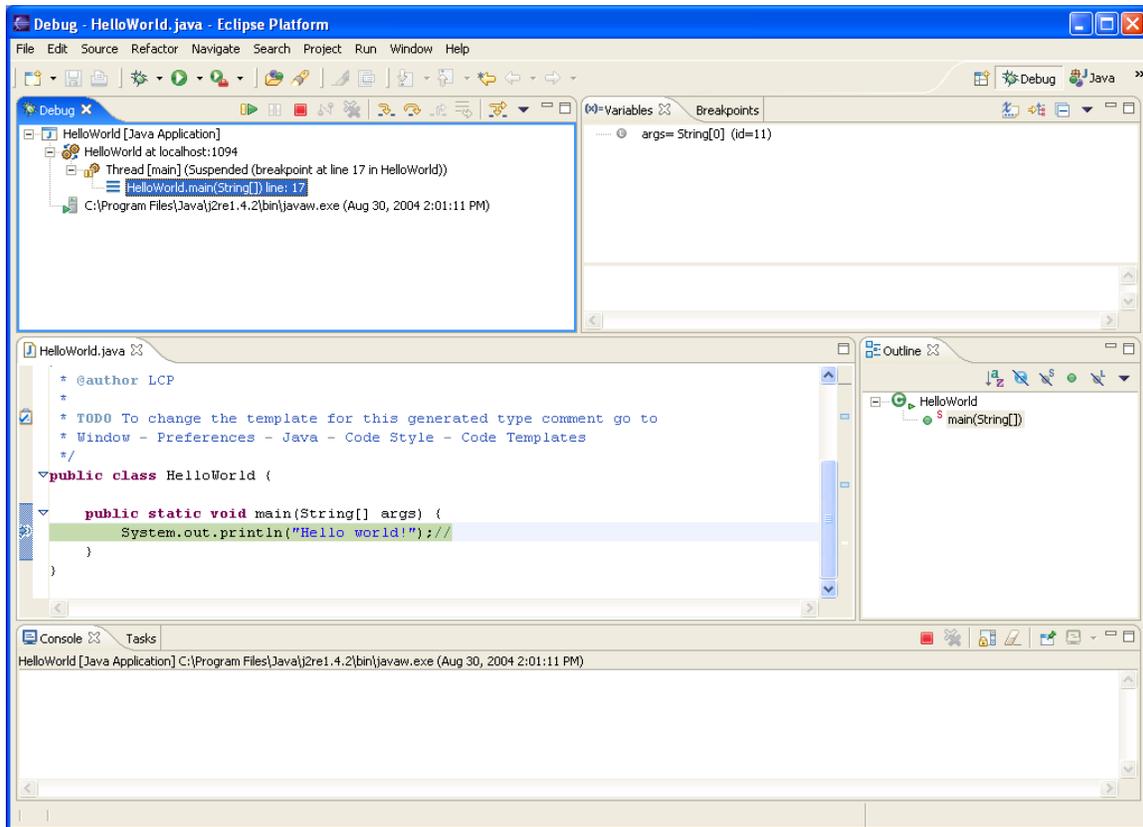
Courtesy of The Eclipse Foundation. Used with permission.

Step 6 - Checking your program

You are not done writing your program until you have checked it by examining its status as it proceeds from start to end, making sure that all variables and expressions have their intended values and behavior at each step.

To check (or, in the old-fashioned terminology used by Eclipse, debug) our program, we need to set a **breakpoint** in our code so the program suspends execution and allows us to see its status. Without a breakpoint, the program will run to completion without letting us look at its status. To set a breakpoint, double-click in the gray margin on the left side of the editor next to the line at which you'd like program execution to stop. A blue dot will appear, indicating an active breakpoint. In this example, set the breakpoint at the line with `System.out.println(...)`

Starting the program to check is similar to running it. Eclipse provides two options: Use the full-service **Run** → **Debug** menu selection to use a launch configuration, or use the express **Run** → **Debug As** → **Java Application** selection if the default options are OK. Use the **Run** → **Step Over** or **Run** → **Step Into** commands to step line by line through your program. (We'll describe this in more detail in lecture.)



Courtesy of The Eclipse Foundation. Used with permission.

Eclipse will show you the values of all variables, expressions, and other program status. There's not much of it here, but there will be plenty even in Problem Set 1.

Congratulations! You have just written your first Java program.

More Information on Eclipse

Eclipse documentation, including a „Getting Started“ section can be found at: <http://help.eclipse.org/ganymede/index.jsp>. This offers a more detailed recitation than what we've laid out here.

Problem 4: Submitting Problem Set 0

For each problem set, you will electronically submit your Java code through our website. You do not have to submit a printed copy of your homework.

To test that you understand homework submissions, try submitting `HelloWorld.java` as your solution to Problem Set 0. To do this, follow the instructions in the document „How to Submit Assignments.“ This document can be found on the course website. Bring any questions or problems to your first recitation.

Please make sure you know how to submit your homework before recitation. If you have any questions, bring them up during your first recitation or office hours.

Turn In

- Turn in **electronic** copies of **all source code** (`.java` files). No printed copies are required.
- Place a **comment** with your full name, recitation section, TA's name, and assignment number at the beginning of all `.java` files in your solution. Comments begin with a double slash (`//`).
- Place all of the files in your solution into a **single zip file**. Submit this single zip file on the 1.00 Web site under the appropriate section and problem set number. For directions, see '**How to Submit Assignments**' on the 1.00 Web site.
- Your solution is **due at 12 noon**. Your uploaded files should have a time stamp of no later than 12 noon on the due date.
- **Do not** turn in compiled byte code (`.class` files) or backup source code (`.java~` files).

MIT OpenCourseWare
<http://ocw.mit.edu>

1.00 / 1.001 / 1.002 Introduction to Computers and Engineering Problem Solving
Spring 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.