

# Homework # 1

## due Tuesday, September 15, 2:00 PM

In this assignment, you write a small Java applet that introduces you to the world, do a design of a academic chat system, and write a small text-based Java program that simulates temperatures in Milwaukee.

### 1 Introducing Yourself

In Chapter 3, Prof. Jia has the planet Venus introduce itself. For this assignment, write an applet `HelloFromMe.java` to introduce yourself. You must create an HTML file `HelloFromMe.html` so that the introduction can be done using `appletviewer`.

Give your name and something about where you originate. You should have a photo (probably of yourself) in `Me.gif` and at least two lines of text, in different sizes. Choose appropriate colors (which should be different than the ones in the book) and choose an appropriate initial window size. The picture should be **centered** within the window with an appropriate margin width. Make sure the picture works well when viewed as an applet.

None of us (to my knowledge!) come from Venus, and so you will need to find or create an appropriate GIF file. If you did this assignment in a previous semester, please choose a new picture and text.

### 2 UML Design

An online course doesn't use a normal classroom and black-board. If the instructor uses PowerPoint, these can be viewed by students any time, but if one wishes to have an interactive experience similar to our classroom discussions, we will need some sort of 'chat' system. Over the course of this semester, you will be implementing this project. I will be using a version of this software next semester when I teach CompSci 654 (Introduction to Compilers) in "blended" (partially online) format.

In the working system, there are two shared spaces: a discussion space similar to a standard chat tool—a participant can post comments at any time. The other shared space is the virtual "blackboard." The instructor can post a line at any time; the instructor can also permit one other participant (a student) to post a single line. For example, if the instructor asks someone to write a line of code on the project being developed in class, they will be given permission to post to the blackboard.

All participants have two windows (one for the discussion, one for the blackboard). The instructor additionally has a window listing the currently logged on and active participants. One line may be highlighted: the student currently permitted/expected to post to the blackboard. That student will have some indication that they are "on."

Develop an object-oriented model for this system:

1. Determine the actors and use cases of the system and describe the relationships among them using use case diagrams.
2. Elaborate each of the uses cases. Describe the scenarios in each of the use cases.
3. Identify the classes of the system, and describe the responsibilities and the features of each class.

4. Determine the relationships among the classes, and use class diagrams to depict the relationships.

Your assignment (which should be done on paper and turned in at the beginning of lecture) should aim to be as complete as possible for this simple system, including all use cases and exceptional behaviors. Don't attempt to model timing (time outs, etc.).

### 3 Cooler Near the Lake?

The directory `$CLASSHOME/src/homework1/` contains a file `heat.cc` that simulates a heat system involving a large lake and inland areas. When it is run, it prints simulated temperatures for the lake and for the areas. The area closest to the lake (Milwaukee's eastside?) shows influence from the lake (Lake Michigan?) whereas the area farthest from the lake (Waukesha?) shows little influence. The following questions are *optional*!

- According to the simulation, what is the coldest month for the Milwaukee's eastside?, Waukesha?
- What is the warmest month for each?
- When is the lake warmest?
- For what months is it indeed "cooler near the lake" in this simulation?
- Does this match with your experience? (The temperature given is the *mean* temperature over the day, not the high or low.)

What your actual task to do is to convert this program into two Java files: `Area.java` and `LakeClimateSystem.java`. You should arrange the code so that the same behavior is seen by

```
millier.cs % java LakeClimateSystem
```

### 4 Submitting Your Work

The design for the PAWS system should be turned in on paper at the *start* of lecture on Tuesday, September 15th. You may also turn it in ahead of time to me (John Boyland) in person.

You submit your program work by putting it in the `homework1` directory in your AFS class volume. You should have two separate directories (Eclipse "projects"), one called `intro` and the other called `heat`. In any case, you will lose permission to write things in this directory after the deadline, which is 2:00pm on Tuesday, September 15th. In other words, you must be done before lecture starts.

The `homework1` directory needs only to include the following:

- `intro/HelloFromMe.java`: the source code for the applet;
- `intro/HelloFromMe.html`: the HTML code for the applet;
- `intro/Me.gif`: the GIF file;
- `heat/Area.java`: The Area class;
- `heat/LakeClimateSystem`: The simulation program.

The sub-directories may include other files as well, such as `.class` files.