

CS 350 – Computer/Human Interaction

Fall 2006 – Java Prototype: A Chat Program

Out: October 4

Due: October 23 (corrected due date)

Your company has decided to go ahead with a more complete implementation of the chat program. They have decided to switch to Java as the platform for this product.

The Task

- Your task is to rewrite the previous (VB.NET) prototype using the Java language. This prototype may be either a stand-alone application or a networked client application and server. As this prototype is mainly to demonstrate the technology with respect to the chat interface, the prototype does not need to actually communicate via the Internet. Files can still be used for this prototype to store messages. The instructor will be happy to assist in discovering the capabilities of the Java environment.
- You must add at least one feature to your chat program not present in your VB.NET prototype. For instance, you might decide to add a login screen, to show how that would look. However, you do not actually have to implement a login mechanism in the program. In other words, you may implement additional *non-functioning interface elements* in this prototype to demonstrate what the finished product might look like.
- Although these features will not actually work, they should *appear to work*, so as to demonstrate what the interface would look like. Thus, logging in does not actually have to work, but you should implement a fake interface, which appears to work. (For example, every time you log in, the same name might appear in the list of participants.) As usual, you should provide some feedback when sending messages, and appropriate error messages if something goes wrong.
- The solution must be written in Java.

What to submit

All Java source files of your project are to be zipped (Windows) or tarred (Unix) and submitted as an attachment via email to the instructor.

Hand in the following items:

- A signed statement that the work you are submitting is your own (see [Project Overview](#))
- A description of the program structure and instructions on how to install your program, including any machine specific code that may need to be changed by the instructor to run your program (e.g., directory names).
- A description of any added features of this prototype.
- Hardcopy printouts of your *well-commented* Java code files.
- A discussion about why you chose to organize the interfaces and storage the way you did, focusing on activity and information design.

Grading

Prototypes will be graded in the following manner:

- 25 points - prototype demonstrates an interface that meets the above specification
- 5 points - aesthetics and usability
- 10 points -discussion