

# ECE 491 Object-Oriented Programming in Java

Spring'04, Lab 7, Due: 03/02/04

## Purpose

IO Streams and Pipes, Java Sockets

### Part I

Given a text file whose name is given as an argument to the program, print out its lines after the following manipulation

- All letters are capitalized
- Each letter is followed by a space
- Lines are sorted by their length

For example, suppose we have a file, which consists of the following five lines.

```
University
of
Illinois
at
Chicago
```

The output will be the following lines

```
O F
A T
C H I C A G O
I L L I N O I S
U N I V E R S I T Y
```

You have to use PipedReader and PipedWriter discussed in the last lecture. I have posted the programs we discussed (<http://www.ece.uic.edu/~i491>). Though you can make the program without using PipedReader/PipedWriter, you have to use piped IO. One pipe capitalizes letters and inserts blanks after each letter except after the last one. Another pipe sorts lines by their lengths. File name should be given as an argument. Output may be directed to a file or standard out. The hardest part may be the threads, but you can finish your work without knowing threads.

### Part II

For this part you will have to retract back to the command-line calculator application that you had created in one of the previous labs. Now modify this to create a client-server application for the calculator. Modify your calculator application such that it now behaves like a server. Write a new client class that sends operands and opcodes to this server (in exactly the same way as the command-line calculator) and displays the result back to the user (again similar to the command-line calculator). Your calculator server should support at least 10 simultaneous clients. You may have to use java-threads for this application.

