**INSY 4305/5352 Assignment #4**

**File Processing**

This will be a non-GUI based assignment. You will be writing three relatively short file processing programs that replicate the functionality of the UNIX utility programs cp, cat, and grep. In all of the programs below you will be using a BufferedReader/BufferedWriterwrapped around a FileReader/FileWriter object. To open a file for reading or writing you will do the following:

BufferedReader in = new BufferedReader( new FileReader( filename ) );

BufferedWriter out = new BufferedWriter( new FileWriter( filename) );

Investigate the API docs for these classes to use the appropriate methods (most probably you'll need to use the methods readLine(), write(...), and others). You will be using command line arguments in all of the programs below. The command line arguments can be accessed inside the application from the String[] args input argument of main method. The first input argument will be stored in arg[0], the second in arg[1], and so on.

(1) (30 points) Write a program CopyFile that copies one file to another. The file names are specified on the command line. For example,

java CopyFile report.txt report.sav

should create another copy of the file report.txt in report.sav. Your program should print the following error message for inappropriate number of input arguments (for e.g., java CopyFile report.txt):

Usage: java CopyFile fromFile toFile

(2) (35 points) Write a program CatFiles that *concatenates* the contents of several files into one file. For example,

java CatFiles chapter1.txt chapter2.txt chapter3.txt book.txt

makes a long file book.txt that contains the contents of the files chapter1.txt, chapter2.txt, and chapter3.txt. The target file is always the last file specified on the command line. As in (1) above, your program should print an appropriate error message of the same format when given inappropriate number of input arguments. Note that your program should be able to concatenate any number of files (and not just 3 files as shown above).

(3) (35 points) Write a program Find that searches all files specified on the command line and prints out all lines containing a keyword. For example, if you call

java Find Buff report.txt address.txt Homework.java

then the program might print the foll. (i.e., all the lines that contain the word "Buff"):

report.txt: Buffet style lunch will be available at the  
address.txt: Buffet, Warren|11801 Trenton Court|Dallas|TX  
address.txt: Walters, Winnie|59 Timothy Circle|Buffalo|MI  
Homework.java: BufferedReader in;

The keyword is always the first command line argument. Note that your program should first print the name of the file, followed by a colon, and the entire line in which the keyword appears. As before, your program should print an appropriate error message of the same format when given inappropriate number of input arguments. Note again that your program should work for any number of files (and not just 3 files as shown above).  
(Note: You will need to use some methods from the String class to search each line returned by the readLine() method for the presence of the keyword.)

**Deliverables:**

You should submit the three .java files: CopyFile.java, CatFiles.java, and Find.java.