**Assignment 1130**

The purpose of this assignment is mainly to give you a taste of a more conventional programming model — the edit-compile-execute cycle. For this exercise, we will use the Java programming language. Despite the name similarity, Java and JavaScript are actually significantly different from each other.

**Not for Submission**

We have selectively covered material from Chapters 6 and 8 from the Brookshear book. If you are unable to read through these chapters in their entirety, then focus on Sections 6.1, 6.2, 6.4, and 8.1.

**Not for Submission (Yet)**

A quick head’s up — your programming portfolio is due on December 14, in approximately 3 weeks. It will consist of fixed-up, improved, and overall better versions of the following assignments:

- Assignment 1019 (the Web pages)
- Assignment 1026, with all three algorithms now “packaged” within standalone Web pages
- Assignment 1121, also with the three JavaScript algorithms within standalone Web pages

Now is a good time to start reviewing your work, making improvements (especially if I marked it up previously), and generally putting your personal stamp on these pages and programs.

**For Submission**

The provided Mystery.java program contains two algorithms, named mystery1 and mystery2, respectively, that take a piece of text and produce another piece of text (or string) as a result. You are asked to figure out, through a combination of reading the source code and running the program, what exactly these algorithms do.

You will need the Java Development Kit (JDK) for this assignment. Check the Java Crib Sheet handout for detailed instructions on downloading, installing, and using the JDK.

1. Type in the Mystery.java program that was given out in class. Try to be as exact as possible — the slightest divergence may result in errors.
2. Compile the program (again refer to the Java Crib Sheet for details); you should now see a file called Mystery.class.
3. Run the program by entering java Mystery from the command line. You need to add two items to the command invocation after java Mystery:
   - Specify 1 or 2 to indicate whether you want to perform mystery1 or mystery2.
   - Follow the 1 or 2 with the double-quoted text that you wish to use as input into the algorithm. For example, java Mystery 1 “hello world” will perform mystery1 on hello world; java Mystery 2 “the quick brown fox” will perform mystery2 on the quick brown fox.
   - In case you forget, java Mystery by itself should print a help message.
   - Mystery will then respond with the result of the algorithm — do this as many times as you wish; experiment with different strings, particularly those with 2 or more words.
4. On hardcopy, state what each algorithm does, and provide “transcripts” of your interactions from the command line (a straight copy-paste into your submission will do) to show that you successfully compiled and ran the program.

Getting a program up and running in Java requires a bit more of a learning curve than JavaScript did, so don’t hesitate to ask if you get stuck.