Assignment #2

```
Original Due:  2:00 PM, Thursday, October 7
    Extended:  2:00 PM, Thursday, October 14
```

You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. You may not copy solutions from the world wide web. The code you write must be your own.

Provided Files:

- A2.c - A lexical analyzer program (written in C) for simple arithmetic expressions
- A2.java - A shell file
- input1.txt - A sample input file
- input2.txt - A sample input file
- answer1.txt - The required output for input1.txt
- answer2.txt - The required output for input2.txt

Description: Convert the given C implementation of lexical analyzer (A2.c) to an equivalent Java program (A2.java). Given an input, your Java program should produce the same output that A2.c produces. To validate your code, you should create at least three non-trivial test inputs on your own. Use a diff tool to ensure your program produces the correct output. Even minor differences in output will cause you to fail grading tests and lose points. You must use the shell file (A2.java) for this assignment.

Submission: A2.java, input**3**.txt, input**4**.txt, input**5**.txt, …

General Programming Assignment Requirements:

- Classes must be in the default (no package statement) unless otherwise specified. You will lose all points if you put a package statement in your program.
- If your program that does not compile, you will lose all points.
- If you submit the wrong file, you will lose all points.
- You must add the header and fill it in the shell file. Otherwise, you will lose all points.

Checklist: Did you remember to:

- worked on the programming assignment by yourself?
- fill in the header in your file A2.java?
- ensure your program does not suffer a compile error or runtime error?
- ensure your program creates the correct output and that it matches the expected output exactly?
- put a comment at the start of each method describing in broad terms, what it does?
- properly indent your source code so that your indenting is readable and consistent?
- use good names for variables to make your program easy to understand?
- turn in your Java source code in a file named A2.java and your own input files through D2L?