CS 121 **Lab Assignment #4**
Bruce Bolden **20 Points**
February 6, 2015 **Due:** February 19, 2015

Objective: Become familiar with the stack abstract data type (ADT) and increase your experience with classes.

Program Description: Real calculators (such as my trusty 30+ year old HP 25) use a form of mathematical notation known as reverse polish notation (RPN). At the heart of RPN is a stack, as discussed in lecture. An RPN calculator works by reading strings from the keyboard. As long as the strings are numbers, you place them on the stack (push). The value on the top of the stack is displayed at all times as a prompt to the user input. When you encounter an operator you pop the appropriate number of numbers from the stack, perform the operation, then push the result back on the stack. For example, to calculate `3.2 + 2.4` on an RPN calculator you would enter:

```
[tony]$ ./rpn
RPN Calculator v.10 by Tony Opheim
RPN (empty) > 3.2 2.4 +
RPN 5.6 >

To divide 3.5 by 2 you would enter:

RPN 5.6 > 3.5 2 /
RPN 1.75 > ps quit
Stack contents: 1.75 5.6
RPN 1.75 >
[tony]$
```

Get the idea? The assignment is to write an RPN calculator that implements the four basic math functions (addition, multiplication, subtraction, and division) and the following special functions: `sq` (square), `sqrt` (square root), `dup` (duplicates the top of the stack), `swap` (swaps the top two elements of the stack), `ps` (print stack), and `quit` (exits your program). All numbers should be stored/manipulated as `double`. All appropriate error checking should be performed including invalid operations. The program should loop continuously for input until terminated with the quit keyword. The user interface should model the application demonstrated above *exactly*. Particularly note that input is processed a line at a time.

Requirements: In addition to the functional description above your solution also:

- MUST have the stack implemented as a linked list class. You may use (extend/modify) any code covered in the course.
- MUST have the calculator logic/implementation in a separate file from the stack implementation (an application file).
- MUST organize the source code files in accordance with the class standards. In other words, separate class declaration files (ADT header file), class implementation files (ADT implementation file), and application file.

Deliverables:

- A program design. Describe all classes and methods needed to implement your program.
- Program—fully documented and functional program. No wrap-around of lines on the printed copy!
- Programming Log:
  - Record the time required to design and implement your program.
  - Record of things you encountered/learned while implementing your program.
- Output—proof that your program worked. Turn in the output from your program illustrating *ALL* of the basic functions described above *PLUS* the following two RPN equations:
  - 2.5 1.3 - sq 7.9 2.4 - sq + sqrt
  - 2.4 sq 3.13 sq + sqrt

If you have any questions regarding this assignment, do not hesitate to contact me. Start working on this assignment as soon as possible.