Computer Science II: Data Structures: Programming Assignment 1

Alexander Starr – 00567613

July 8, 2013

Problem Description

The problem we are solving for this programming assignment is the conversion of an expression in infix notation to postfix notation, and then the evaluation of the postfix expression. Infix notation features the operators between the operands. Postfix notation has the operator after two operands (or two sub-expressions which can be evaluated).
Both of these goals are accomplished through the use of stacks, which have the property of first-in, last-out. In the conversion of infix to postfix notation, the stack is used for the operators. However, the stack is used for the operands in the evaluation of the postfix notation.

Source Code and Test Results

Please see `main.cpp` for the source code of my solution.
The following are examples of input and output of the program:

```
Enter an infix expression:  1+2/3
Postfix:  1 2 3 / +
Evaluation:  1.66667

Enter an infix expression:  (1+2) / 3
Postfix:  1 2 + 3 /
Evaluation:  1

Enter an infix expression:  1+2*3-4/5^6
Postfix:  1 2 3 * + 4 5 6 ^ / -
Evaluation:  6.99974

Enter an infix expression:  (1+2)*(3-4)/5^(6-5)
Postfix:  1 2 + 3 4 - * 5 6 5 - ^ /
Evaluation:  -0.6

Enter an infix expression:  (11-1)/(3^2+1) * 100 - 1
Postfix:  11 1 - 3 2 ^ 1 + / 100 * 1 -
Evaluation:  99

Enter an infix expression:  (1 + 2)/ 3
Postfix:  1 2 + 3 /
Evaluation:  1
```

Note that the amount of spacing between characters doesn't matter. Also, the program respects the mathematical order of operations, as well as parentheses.