# CS 211 Lab #3

In this lab, you will build a postfix (sometimes called Reverse Polish) calculator. [Wikipedia](http://en.wikipedia.org/wiki/Reverse_Polish_notation) has a pretty good article on postfix, which I recommend you read. Basically, instead of having mathematical operators between values, we place the operator at the end of the expression. As such, "3 + 2" becomes "3 2 +". As a more complex example, the equation "(1 + 2) \* 5 / (3 - 7)" would convert into "1 2 + 5 \* 3 7 - /".

Postfix calculators can be built quite easily using stacks. I used my ListStack for this lab, but you are free to use whatever you like (including the STL implementation). The basic procedure for postfix calculations using a stack are as follows:

1. Given a string-based expression
2. Split that expression into pieces (the included StringSplitter works well for this)
3. For each piece:
   1. If the piece is a number, push the number onto the stack
   2. Else if the piece is an operator (e.g. +, -, \*, /), it is time to perform a calculation:
      1. pop off the top two values on the stack and perform the requested mathematical operation.
      2. Push the result calculated in the previous step back onto the stack.

Your calculator should also detect invalid states. Such states include:

1. Having more than one value left in the stack after the final calculation is performed
2. Having too many operations in the equation
3. Having unexpected input in the expression string (e.g. "3 abcd +")

Your program will ask the user for a text file that contains one or more postfix expressions. You will output to the screen both the expression and its solution or the text "invalid expression" if the line contains an invalid postfix expression.