Definitions: Infix, Postfix, Prefix algebraic expressions

Evaluating Postfix Notation
While more characters:
Get next character
If number, push on stack
If operator, pop 2 items off stack, operate, push result back on stack

Converting Infix to Postfix
Notes:
• Operands always stay in the same order
• Operators move only to the right with respect to operands
• All parentheses are removed
• We will only use + - * /

While more characters:
Get next character
If number, put in output string
If operator:
• ( - Push on stack
• ) - Pop from stack, placing each operator in output string, until ( is popped. Discard both ( and )
• Other operators - Compare to top of stack.
  // Can only push operator on one of LOWER precedence
  o If next operator is of higher precedence than operator on stack, Push next operator
  o If operator is of lower or equal precedence, Pop stack and place operator in output string, compare again
  o If no more characters, pop stack and place operator in output string until the stack is empty

Precedence
* / highest precedence
+ - middle
( lowest precedence

Strings
#include <string>
string Postfix(" "); // declare output string and initialize to be empty
Postfix = Postfix + ch; // appends character to the output string

Homework - Due Wednesday, Feb 22 - turn in at beginning of class. Show your work.
1. Convert to postfix
   a) 4 * 6 + 9 - (5 + 2) / 6 * 3 * (4 * 9)
   b) x + y - z * ((a + b) / (x - z)) - 2
2. Evaluate the following postfix expressions
   a) 6 9 6 * + 4 2 / -
   b) 3 5 4 + 9 / 2 * 4 5 3 + + 5 - * *