Programming Assignment #7

Match Brackets using a Stack

CS 2308.003, Fall 2011
Instructor: Jill Seaman

Due: Wednesday, 11/30/2011 (upload electronic copy by 11pm)

Problem:

Given a text file, your program will determine if all the parentheses, curly braces, and square brackets match, and are nested appropriately. Your program should work for mathematical formulas and many computer programs.

Your program should read in the characters from the file, but ignore all characters except for the following:
`{ } ( ) [ ]`

Your program should use the IntStack implementation (IntStack.h and IntStack.cpp, which will be available from the class webpage).

The general algorithm is to use the stack to store the opening unmatched brackets. When a closing bracket is encountered, check it against the one on top of the stack (pop it off)--make sure it matches. When you are finished there should be no unmatched brackets left on the stack.

Input/Output:

Your program should prompt the user to enter a filename. It should then try to open the file and then check it make sure the brackets all match appropriately. If they all match, the program should output a message saying so. If not, the program should output an appropriate error message.

There are three types of errors that can happen (and they can happen with any kind of bracket):

missing } : if you reach the end of the file, and there is an opening { that was never matched.
expected } but found) : this is a wrong closing bracket, like: { x[size]=10; **)** ...
unmatched } : this occurs if there is a closing bracket but not an opening bracket (not even one of the wrong kind).

NOTES:

The stack is an int stack but you will probably want to put characters on the stack. You can push a character on the stack, but you must pop into an int variable. This is only a problem if you want to display the popped character to the screen. If so you will need to convert it to a char first.

Your program will not be nearly as sophisticated as a compiler. A compiler would ignore brackets inside of comments and inside of quotes (string and character literals). So there will be many valid programs that would cause errors if tested in your program.

Follow the style guidelines described here:
http://www.cs.txstate.edu/~js236/styleguidelines.txt
(there is a link to these on the course webpage as well).

Logistics:

For this assignment you will have only one file to submit, the one containing your main function. Name your file **assign7_xxxxxxxxx.cpp** where xxxxxxxxx is your 9 character TX state ID number, the one that is on your ID card.

There are **two** steps to the turn-in process (also see the Assignment Turn-in Policy):

1. Submit an **electronic copy** using the following upload link:

   http://www.cs.txstate.edu/~js236/homework
   (There is a link directly to this page on the course website).

   Click on your course number, and log in with your Net ID and follow the directions to upload your file.

2. Submit a **printout** of the file at the beginning of class on the day after it is due. Please print your name on the front page, and staple if there is more than one page.