CSC331: Data Structures - BMCC Fall 2023
Professor Byron
Program #3 specifications: Stack implementation
Due: 1159pm Tue 11-7-2023
No credit for late submission

Your task for this assignment is to implement a stack data structure in C++. This may be accomplished by utilizing the C++ standard template library (STL) or by utilizing a user-defined class.

1. Implement a stack data structure using C++. The program will be interactive. Data transactions will be entered as a batch of transactions in a text file at the command line using redirection. The result of each transaction will be displayed on the console.

   For example: $ prog3 < prog3.dat

2. Each input transaction will contain an arithmetic expression in post-fix format. Assume that each operand and operation (+, -, *, /, %, ^) is a single character in each arithmetic expression without spacing. You may assume that each arithmetic expression is correctly formatted. However, your program should not attempt the following:
   a. Divide by zero with / or % … Instead, your program should display an error message and then begin evaluating a new input transaction.
   b. Perform % with non-integer operands … Instead, your program should display an error message and then begin evaluating a new input transaction.

   Note that the intermediate or final results of an expression may be negative or non-integer.

   Sample input transactions and results:
   a. 23+ (result = 5)
   b. 34*5/ (result = 2.4)
   c. 832*6-/ (result = “error: division by zero”)
   d. 48*62*42^3+-% (result = 4)
   e. 92/73/5%4/+ (result = “error: non-integer operand for %”)

3. An input transaction containing “end-of-file” indicates there are no more transactions to be processed. Implement a stack to evaluate each expression and display the result. Use the C++ built-in class or a user-defined class to implement stack functions.

4. The program will be run at the command prompt by navigating to the directory containing the executable version of the program after the program is compiled. The program should display a prompt requesting input, such as “Please enter an expression in post-fix notation:”.

5. Your C++ program file should be named csc331_prog3_lastname.cpp. Your program should contain comments starting on line 1 of the program containing the following information:

    a. course ID and section
    b. your full name
    c. the program file name
    d. the program assignment number and due date
    e. the program purpose

    You are encouraged to add additional comments throughout the program that your feel might be helpful to the reader of your source code.

6. Submit your C++ program (cpp file) and related header files as separate, unzipped attachments to an email message to kbyron@bmcc.cuny.edu using a subject in this form: "csc331_prog3_lastname".

7. Do your own work. Students submitting copies of the same program will receive grades of zero for the assignment. Do not share your code until you have received your final grade for the course at the end of the semester.

8. Grading rubric

    Partial or extra credit will be awarded as follows:

    60%: significant logic with no compiling errors
    90%: 5 of 6 operations working, but only one transaction processed
    100%: 5 of 6 operations working, and terminated by "end-of-file" transaction
    110%: 6 of 6 operations working, and terminated by "end-of-file" transaction
    120%: 6 of 6 operations working, terminated by "end-of-file" transaction and user-defined stack class using a linked list