Module 4: Programming Assignment 4

Start Assignment

- Due No Due Date
- Points 100
- Submitting a file upload
- File Types h, and, and cpp

Programming Assignment

A stack is a *LIFO* (last in, first out) data structure. Stacks are extremely useful in programming. A stack can be implemented using an array to hold values, and a counter to indicate where the "top" item on the stack is. See the **first example (https://onlinelearning.berkeley.edu/courses/2291660/files/313890769/download?wrap=1)** ↓ **(https://onlinelearning.berkeley.edu/courses/2291660/files/313890769/download?download_frd=1)** . (You will need to download the .zip file and decompress it first.) This is a class declaration for a stack of doubles. Store your class declaration in a file called dblstk.h and implement the specified functions exactly as given. You may define whatever private members you think you need. You may also specify additional member functions if you find them useful.

Develop a test program in a file called stack_driver.cpp for your stack implementation. This driver should define a stack to hold ten values and allow the user to specify, via a menu, the following operations. Allow the user to enter in the values to be pushed to the top of the stack 1 via menu option 2. Do not initialize stack 1 with any values, all values are to be entered by the user. The user menu should appear exactly as shown:

1. Display stack 1
2. Place a double value onto the top of stack 1
3. Remove a value from the top of stack 1
4. Check the total capacity stack 1
5. Check current number of items on stack 1
6. Copy stack 1 to stack 2
7. Check to see if the two stacks are equal
8. Quit

Enter menu selection:

Finally, obtain a copy of the instructor's program postfix.cpp; it should be in the downloadable sample source code, as described in Module 1. This program evaluates postfix arithmetic expressions using your stack implementation. Replace your driver module with the postfix.cpp module. The program should run without modification, assuming you have implemented the stack correctly.

Submit the four source code files (stack_driver.cpp, dblstk.cpp, dblstk.h, postfix.cpp) as attachments to the drop box for this module's programming assignment. You can find the drop box at the end of this module's tab page. Do not compress your attachments. Please review the requirements for submitting on the course syllabus.

Discussion Assignment

You'll find this module's assignments under the heading *Module 4: Dynamic Allocation*. Post your response as a reply to the original message. Be sure to critique at least one other student's Module 4 response by replying directly to their posting.