CSE 2383: Data Structures and Algorithm Analysis

Challenge 2 – Queue Who?

Submission Window Opens:
Due Date Posted to Canvas

Points Available:
90 points for a working demonstration

10 points for correct submission & understandable code

Objectives:

- Demonstrate a working stack
- Demonstrate sufficient knowledge to convert a stack into a queue

Assignment:

In class, we have learned about stacks. For this challenge, you must write a program that implements a stack derived from the code provided in class. In addition, you should add a display function to your stack class that shows the contents of your stack (stack's top to stack's bottom shown left to right). A series of function calls and their subsequent output are included below. You must demonstrate a working stack to the TA. As part of the demonstration process, the TA will also ask you to modify your stack code such that it becomes a queue. DO NOT BRING QUEUE CODE FOR DEMONSTATION. Do, however, practice and study the implementation and differences between these data structures. Your queue's display function will need to show the queue from its front to back displayed left to right.

As with the previous challenge, you must fully pass the demonstration before a grade will be assigned to your submission. Grades are assigned based on day of submission (see syllabus). Once you have demonstrated your submission, you must upload it to Canvas within 48 hours.

You will need to bring your laptop to the TA for demonstration. If you cannot bring your laptop, make arrangements ahead of time with the TA to demonstrate it some other way. Arrangements must be made at least a day ahead of your planned demonstration.

Deliverables

1. Demo your working code to the class TA ***before uploading*** it to Canvas. You cannot proceed to step 2 before doing this.
2. Once your code is working and you've demoed it to the TA, upload all your code to Canvas as a single ZIP file. Name your ZIP file <netID>_2383_Ch0.zip, where <netID> is your MSU Net ID.

Example Testing Code:

```
Stack S;
int tmp;
S.push(5);
S.peek(tmp);
cout << tmp << endl;
S.push(33);
S.push(1);
S.push(7);
S.peek(tmp);
cout << tmp << endl;
S.push(33);
S.push(12);
S.display( cout );
S.pop( );
S.display( cout );
S.push(14);
S.display( cout );
S.pop( );
S.pop( );
S.display( cout );
S.pop( );
S.pop( );
S.display( cout );
S.peek(tmp);
cout << tmp << endl;
```

Example Testing Output:

```
5
7
12 33 7 1 33 5
33 7 1 33 5
14 33 7 1 33 5
7 1 33 5
33 5
33
```