

## CS3358.006 - Data structures & Algorithm

### Assignment 1

Due Date: Tuesday, 9/25/2018

Instructor: Dr. Kechang Yang ([yangk@txstate.edu](mailto:yangk@txstate.edu))

Instructional Assistant: Hanie Samimi ([h\\_s163@txstate.edu](mailto:h_s163@txstate.edu))

1. In this question, you are going to implement a stack data structure in part 1 and then use it to create postfix mini-calculator in section 2.
  - 1.1. Use the myStack.h to implement a stack template. It should be able to work for arbitrary data types (char, int, double, etc.).
  - 1.2. Postfix calculator:  
Complete the provided main script to implement a mini postfix calculator for integer numbers. We use infix notation in regular calculations like  $2+4*3$ , while its postfix notation is  $2\ 4\ 3\ *\ +$ .  
Use your stack data structure from part 1.1 and the following algorithm to to create the postfix calculator:
    - Read a line from terminal as the user input.
    - If the input is a valid integer, you push that integer onto the stack.
    - If the input is a valid operator (+, \*, /, -), pop two integers off the stack, perform the requested operation, and push the result back onto the stack. If there were less than 2 integers remained in the stack, print a meaningful error message.
    - If the user enters "c", you need to print the final result. If the user input line was not in a correct format, you should print a meaningful message.
    - If the user enters "H", you halt the program without performing any additional computations.
    - For any other input characters, print a meaningful error message.
2. Use the myQueue.cpp and myQueue.h to implement a queue data structure for integer data type:
  - To keep track of the front and rear of the queue, you can use two integer indicators.
  - Be careful about the "empty" and "full" state. You can use one of the three implementations that distinguish these two states.
  - Complete the main script of this question.

**Submission instructions:**

- \*\* Complete the main.cpp and myStack.h in the folder Q1 and complete the main.cpp, myQueue.h, myQueue.cpp in folder Q2. In submission, please pack the two folders in a zip folder called "a1-student ID", e.g. "a1-A087654321".
- \*\* Mention your compiler and platform that you used to write the script in a file called readme.txt. Add this file to the "a1-student ID", too.
- \*\* Do NOT forget to write (in comments) your name and student ID at the top of all source files. Upload the zipped folder on the TRACS.