

CS 3305: Data Structures Spring 2021

Assignment 10 – Heaps 100 points

Note: If you re-upload the files, you must re-upload ALL files as the system keeps the most recent uploaded submission only.

Note 2: Never hard-code test data in the test program. Always allow the user to enter the test data using menu option.

The goal of this assignment is to reinforce the concept of heap and implement priority queue as a heap in C++.

For this assignment, use the zip file [Priority_Queue_Heap.zip](#) provided with this assignment. It includes five files. Compile, and run the files and understand how class heap and the main function work.

Complete the implementation of the template class [PQ_Heap.template](#). The only **member data** we need for this class is a **heap object**.

To implement all priority queue functions defined in file [PQ_Heap.h](#), you may need to include additional functions to files *heap.h* and *heap.template* (provided in the zip file). Do NOT rename these 2 files.

Next, create a new test file [testPQH.cpp](#) to test class *PQ_Heap*. Use a menu system as follows. The menu allows the user to start with option 0 (to select and set the data type of queue), then exercise other options.

0. Enter Queue Type (int or string)
1. Enqueue Element
2. Dequeue Element
3. Check is_Full
4. Check is_Empty
5. Print Size
6. Display Front Element
7. Print Queue Elements
8. Quit program

For option 7, utilize function *check_heap()*. Make sure the code gives proper messages for special cases, such as dequeue from an empty queue or enqueue to a full queue, etc. Re-display the menu after each option is fully exercised.

Hint: How do we force the user to start with option 0? How do we force the user to start with option 0 before opening other options? A simple approach is to use a boolean variable (say `bool flag = false;`); and while the flag value is false, we keep asking the user to select option 0 (i.e., cannot select any other option). [If you remember, we have done this in Java using the concept of **sentinel loop**]. Once option 0 is selected with the correct type, set variable flag to true and that opens other options for user to select from. Notice that we need to start with the option 0 to be able to create/select/work with the correct Heap object for the user inputs.

Do not forget to include author header in each submitted file as shown, **no header, no points!**

```
// Name:          <your name>
// Class:         CS 3305
// Term:          Spring 2021
// Instructor:    Dr. Haddad
// Assignment:    10
```

Submission:

Please submit all five files ([heap.h](#), [heap.template](#), [PO_Heap.h](#), [PQ_Heap.template](#), and [testPQH.cpp](#)) to the assignment submission folder in D2L by the due date posted in D2L. **Do NOT submit file `main.cpp` provided with the assignment.** **No late submissions are accepted.** Once again, please include **author header** block in each file - **no headers, no points.**

Important Note: The code must be correctly running right and gives correct result in the required environment (CodeLite and GNU C++ compiler) before being uploaded.