

COMP 3270

Programming Assignment

(100 points)

Please submit on Canvas in one package the sourcecode, binary executable, test cases, and the instructions for executing the program

by 11:55PM on Wednesday July 24th, 2013

Implement a priority-queue abstract data type based on the following Max-Heap class specification. Your program needs to implement the specified Max-Heap. The priority-queue data structure should be based on the implementation of Max-Heap.

```
class heap {                                // Max-Heap class
private
    ELEM* Heap;                            // Pointer to the heap array
    int size;                              // Maximum size of the heap
    int n;                                 // Number of elements now in the heap
    void siftdown(int)                     // Put an element in its correct place ~
                                         max-heapify(int)

public:
    heap(ELEM*, int, int);                 // Constructor
    int heapsize() const;                  // Return current size of the heap
    bool isLeaf(int) const;                // TRUE if pos is a leaf position
    int leftChild(int) const;              // Return position for left child
    int rightChild(int) const;             // Return position for right child
    int parent(int) const;                 // Return position for parent
    void insert(const ELEM);                // Insert value into heap
    ELEM removemax();                      // Remove maximum value
    ELEM remove(int);                      // Remove value at specified position
    void buildheap();                      // Heapify contents of heap
};
```

The following commands should be supported for manipulating the priority queue:

```
void enqueue(int Object_ID, int priority);
int dequeue();
void changeWeight(int Object_ID, int new_priority);
```

Function **enqueue** inserts a new object into the priority queue with ID number **Object_ID** and priority **priority**. Function **dequeue** removes the object with the highest priority from the priority queue and returns its **Object_ID**. Function **changeWeight** changes the priority of the object with ID number **Object_ID** to be **new_priority**. The type for **ELEM** should be a class or structure that stores the

Object_ID and the priority for that object. You will need a mechanism for finding the position of the desired object within the heap. Use an array, storing the object with **Object_ID** i in position i . (Be sure in your testing to keep the **Object_IDs** within the array bounds.) You must also modify the heap implementation to store the object's position in the array so that updates to objects in the heap can be recorded in the auxiliary structure.

Evaluation Criteria: This assignment is a group project. Each group is comprised of two students, who will collaborate to develop the algorithms, implement the solution, and test its correctness against the above specification. Your program will be evaluated according to two criteria.

(1) Am I solving the problem right?

- Correctness of the implementation with respect to the Max-Heap algorithm, as well as the design/implementation of the priority-queue in terms of the Max-Heap class. This includes verification of the program against the mechanism of the binary heap algorithm covered in class.

(2) Am I solving the right problem?

- Validation of the program against the requirements (expected behavior) of the priority-queue as defined above. This phase involves evaluating the program's functionality using test cases.

Each group will submit a well-documented and structured source-code in C++ or Java, the test cases, the executable, and the instructions for running and testing the program. Your program should provide an interface (console or graphical) to allow a user to interactively test the **enqueue**, **dequeue**, and **changeWeight** operations.