P3: Highest Priority

92.5/100 Points

11/14/2023

Offline Score: **92.5/100**

Add Comment

Anonymous Grading: **no**

Details

Week 7: Overloaded Operators & Hashes

(https://seattleu.instructure.com/courses/1610311/pages/week-7-synopsis)

ICE11: Overloaded operators (https://seattleu.instructure.com/courses/1610311/assignments/7157046) | P3: Highest Priority | P3x: Changing Priorities (EC) (https://seattleu.instructure.com/courses/1610311/assignments/7157077)

P3: Highest Priority

All Projects (P)

Instructions:

You are creating a priority queue system (triage) (https://en.wikipedia.org/wiki/Triage) for a hospital emergency room. The levels of priority for the patients are the following:

Level	Priority Code	Should be seen by provider within
1	`immediate`	0 minutes
2	`emergency`	15 minutes
3	`urgent`	30 minutes
4	`minimal`	120 minutes

The triage nurse will determine the patient's priority based on their injury or illness and enter the patient's full name along with the urgency level based on the table above to indicate the priority.

The interface of the program is a command-line prompt that supports the following commands:

add command

The `add` command has 2 operands that need to be supplied, separated by space.

- The `<priority-code>` operand must be one of the valid emergency keywords from the above table

(https://seattleu.instructure.com/courses/1610311/modules/items/17917034) (https://seattleu.instru

Here are some examples (user's input in green):

```
triage> add urgent Wilford Hatheway
Added patient "Wilford Hatheway" to the priority system

triage> add minimal Maxim Platt
Added patient "Maxim Platt" to the priority system

triage> add emergency Kylie Carter
Added patient "Kylie Carter" to the priority system

triage>
```

peek command

Displays the patient that is next in line, but keeps the patient in the waiting room.

```
triage> peek
Highest priority patient to be called next: Brenton Jamison
```

next command

Removes the waiting patient from the priority system and announces the name.

Example:

```
triage> next
This patient will now be seen: Brenton Jamison
```

list command

Displays the list of patients currently in the waiting room. Note that the list of patients should always be in heap order, so this command may be useful for debugging issues with your heap.

Example:

```
triage> list
# patients waiting: 1

  Arrival #   Priority Code   Patient Name
+-----------+---------------+--------------+
    2         emergency       Moxie Crimefighter

triage>
```

load command

This command is implemented for you. It reads a text file (i.e. "commands.txt") and executes each line as if you were to type it into the prompt. Download the project files (which includes "commands.txt") here: **p3-files.zip (https://seattleu.instructure.com/courses/1610311/files/69761802/download?wrap=1)** **(https://seattleu.instructure.com/courses/1610311/files/69761802/download?download_frd=1)**

help command

(https://seattleu.instructure.com/courses/1610311/modules/items/17917034) (https://seattleu.instru

```
triage> help
add <priority-code> <patient-name>
            Adds the patient to the triage system.
            <priority-code> must be one of the 4 accepted priority codes:
                1. immediate 2. emergency 3. urgent 4. minimal
            <patient-name>: patient's full legal name (may contain spaces)
next        Announces the patient to be seen next. Takes into account the
            type of emergency and the patient's arrival order.
peek        Displays the patient that is next in line, but keeps in queue
list        Displays the list of all patients that are still waiting
            in the order that they have arrived.
load <file> Reads the file and executes the command on each line
help        Displays this menu
quit        Exits the program

triage>
```

quit command

This command is implemented for you.

Triage System details (`p3.cpp`):

The person that you inherited this project from has done most of the user interface for you. The items left to be done are clearly marked by either printing an error to the screen, or having a comment with the text **TODO** in them (or both). Note that all user input and printing to the screen must happen in this class. Download the project files (which includes "p3.cpp") here: **p3-files.zip (https://seattleu.instructure.com/courses/1610311/files/69761802/download?wrap=1) (https://seattleu.instructure.com/courses/1610311/files/69761802/download?download_frd=1)**

Patient class details (`Patient.h`):

The `Patient` class should contain:

- private variables `name` (patient's full name), `priorityCode` (patient's assigned priority), and `arrivalOrder` (assigned arrival number).
- overloaded operators in order to properly add, peek, and remove patients from the waiting room.
  - How do you compare two patients? In other words, define what it means to be sorted. Document any assumptions you make.
- `to_string` function that returns the string representation of the object.
  - For example: Jane Smith { pri=urgent, arrive=3 }
- constructor(s) and any other required functions needed to make the class work.

Note:

- You are <u>not</u> allowed to use any printing/reading from console in the `Patient` class (i.e. `cout`, `cin`, `printf`, etc).
- Do <u>not</u> create any additional files. `Patient.h` should include the entire class definition (specification and implementation).

Patient Priority Queue class details (`PatientPriorityQueue.h`):

< (https://seattleu.instructure.com/courses/1610311/modules/items/17917034) (https://seattleu.instru

assigned proper `arrivalOrder` (start numbering at 1).

Note that when determining which patient needs to be seen next, the higher level of emergency (table at the top of the page) patients will always be seen first. If there is a tie, and multiple patients have the same level of emergency, the order that they have arrived in the waiting room will be used to break the tie.

You will need to implement the following public functions:

- Constructor - Creates an empty triage system with no patients.
- `add` - Adds the patient to the priority queue. Heap order is defined as the order that patients must be seen, so this function needs to maintain heap order.
- `peek` - Returns the highest priority patient without removing it.
- `remove` - Removes the highest priority patient from the queue and returns it. This function needs to maintain heap order.
- `size` - Returns the number of patients still waiting.
- `to_string` - Returns the string representation of the object in heap (or level) order.
  - You may want to `#include <sstream>` and use a `stringstream` object to capture the string traversal.

Notes / Hints:

- You may use the code from **ICE 9: Construct a heap (https://seattleu.instructure.com/courses/1610311/assignments/7157058)** for this assignment.
- The implementations of member functions must match the public declarations provided above. Do not change the arguments, return values, or function names defined above.
- Do not add any additional `public` functions or fields to the `PatientPriorityQueue` class. You may create additional `private` helper functions or fields. Remember that helper functions may have the same name as the public function, but will need to have different arguments.
- Do not create any additional files. `PatientPriorityQueue.h` should include the entire class definition (specification and implementation).
- You are not allowed to use any printing/reading from console in the `PatientPriorityQueue` class (i.e. `cout`, `cin`, `printf`, etc).

Sample output:

The following is a section of the console window:

```
Welcome to the hospital triage system.


triage> help
add <priority-code> <patient-name>
            Adds the patient to the triage system.
            <priority-code> must be one of the 4 accepted priority codes:
                1. immediate 2. emergency 3. urgent 4. minimal
            <patient-name>: patient's full legal name (may contain spaces)
next        Announces the patient to be seen next. Takes into account the
            type of emergency and the patient's arrival order.
```

(https://seattleu.instructure.com/courses/1610311/modules/items/17917034) (https://seattleu.instru

```
quit        Exits the program

triage> add minimal Sirjames
Added patient "Sirjames" to the priority system

triage> add emergency Moxie Crimefighter
Added patient "Moxie Crimefighter" to the priority system

triage> add immediate Brenton Jamison
Added patient "Brenton Jamison" to the priority system

triage> list
# patients waiting: 3
  Arrival #   Priority Code   Patient Name
+-----------+---------------+--------------+
      3       immediate       Brenton Jamison
      1       minimal         Sirjames
      2       emergency       Moxie Crimefighter

triage> peek
Highest priority patient to be called next: Brenton Jamison

triage> next
This patient will now be seen: Brenton Jamison

triage> list
# patients waiting: 2
  Arrival #   Priority Code   Patient Name
+-----------+---------------+--------------+
      2       emergency       Moxie Crimefighter
      1       minimal         Sirjames

triage> next
This patient will now be seen: Moxie Crimefighter

triage> next
This patient will now be seen: Sirjames

triage> next
There are no patients in the waiting area.

triage> load load /home/fac/mthayer/submit/23fq5005/files/p3-files/commands.txt

triage> add urgent Wilford HathewayAdded patient "Wilford Hatheway" to the priority system

triage> add immediate Brenton JamisonAdded patient "Brenton Jamison" to the priority system

triage> add immediate Jayson AshworthAdded patient "Jayson Ashworth" to the priority system

triage> add minimal Maxim PlattAdded patient "Maxim Platt" to the priority system

triage> add emergency Kylie CarterAdded patient "Kylie Carter" to the priority system

triage> add minimal Peta SmedleyAdded patient "Peta Smedley" to the priority system

triage> add immediate Orson ParishAdded patient "Orson Parish" to the priority system

triage> add minimal Rodger NessAdded patient "Rodger Ness" to the priority system

triage> add emergency Burton AstonAdded patient "Burton Aston" to the priority system

triage> add immediate Emery PearsonAdded patient "Emery Pearson" to the priority system

triage> add minimal Jeremiah EveredAdded patient "Jeremiah Evered" to the priority system

triage> add emergency Ethelbert StringerAdded patient "Ethelbert Stringer" to the priority system

triage> add emergency Terrence HuddlestonAdded patient "Terrence Huddleston" to the priority system
```

(https://seattleu.instructure.com/courses/1610311/modules/items/17917034) (https://seattleu.instru

```
      5       immediate       Brenton Jamison
     13       immediate       Emery Pearson
      6       immediate       Jayson Ashworth
     12       emergency       Burton Aston
      8       emergency       Kylie Carter
     15       emergency       Ethelbert Stringer
     10       immediate       Orson Parish
     11       minimal         Rodger Ness
      7       minimal         Maxim Platt
      4       urgent          Wilford Hatheway
     14       minimal         Jeremiah Evered
      9       minimal         Peta Smedley
     16       emergency       Terrence Huddleston

triage>
```

Submission:

You must name your files **Patient.h**, **PatientPriorityQueue.h**, and **p3.cpp**

To submit, type the following command at the prompt in the directory where the P3 files reside:

```
/home/fac/mthayer/submit/23fq5005/script/p3_runme
```

You have read/write permissions on your submission directory at:

```
/home/fac/mthayer/submit/23fq5005/p3/yourusername
```

View Rubric

Select Grader

Elaine Huynh (TA)

(https://seattleu.instructure.com/courses/1610311/modules/items/17917034) (https://seattleu.instru

P3: Highest Priority

Criteria	Ratings	Pts
PatientPriorityQueue add function is properly implemented view longer description	**Comments** Should not use same heapify method used for remove -4: add not implemented properly - should implement a siftUp method that compares the new element with its parents and swap with parent until it's priority level is less than its parent's	21 / 25 pts
PatientPriorityQueue remove function is properly implemented view longer description	**Comments** -0.5: should not create new patient object -2: remove() should return a Patient object,	23.5 / 25 pts
PatientPriorityQueue peek and size functions are properly implemented view longer description	**Comments** good	5 / 5 pts
Patient class is properly implemented view longer description	**Comments** -1: should use the const keyword for getter functions	24 / 25 pts
Patient and PatientPriorityQueue to_string functions are properly implemented view longer description		5 / 5 pts
Program functions as expected and all functionality is tested view longer description		10 / 10 pts
Documentation/Style view longer description	**Comments** - missing class pre/postconditions for some methods (-1)	4 / 5 pts
Other comments view longer description	**Comments** nice work!	-- / 0 pts

(https://seattleu.instructure.com/courses/1610311/modules/items/17917034) (https://seattleu.instru

(https://seattleu.instructure.com/courses/1610311/modules/items/17917034)

(https://seattleu.instru