M4.B4: Module 4 Priority Queues and Heaps Application Programming Assignment

Due Feb 27, 2022 by 11:59pm **Points** 100 **Submitting** a file upload **Attempts** 1
Allowed Attempts 1

This Programming Assignment is based on Application Exercise 5.7.29:

- Describe an algorithm for the New Unknown Airline (NUA) Upgrade System.
- Develop a program that processes the request and cancellations for upgrade and provides the list of k-highest priority flyers among the *n* frequent flyers on the waiting list.
- Your implementation must process the request and cancellations in O(logn) time and find the k-highest-priority flyers in O(k logn) times using the data structures in Chapter 5.

In your submission, you must upload **two** files:

- Submit a typed Word or PDF document with description of your solution on Canvas.
  - Your answers should be very clear, in proper order, and use complete sentences.
  - Review your work several times before submission to be sure the steps of the algorithm are clearly and properly stated and in the correct order.
  - Provide pseudocode for the main algorithms, except for user interface, input/output, etc.
- Submit a single zip file named NUA_Upgrade_System_Lastname.zip containing the code file and test files.

Here are some further guidelines for programming code:

- Use an OOP language, such as Java, Python or C++.
- Comment your code.
- Your code file must compile and accept any number of inputs in the format you specified.

Grading Rubric for Programming Assignment 1 (100 Points)

Criteria	Ratings			Pts
PDF or Word file describing the algorithms for implementation	**20 to >17.0 pts** **Exemplary** The Rubric of Application Exercise Applies	**17 to >10.0 pts** **Moderate** The Rubric of Application Exercise applies	**10 to >0 pts** **Insufficient** The Rubric of Application Exercise Applies	20 pts
Algorithm to process Upgrade Request Implementation	**16 to >13.0 pts** **Exemplary** Correct implementation with error checks	**13 to >8.0 pts** **Moderate** Correct implementation without error checks	**8 to >0 pts** **Insufficient** Code does not compile	16 pts
Algorithm to provide k-highest-priority flyers at anytime	**16 to >13.0 pts** **Exemplary** Correct implementation with error checks	**13 to >8.0 pts** **Moderate** Correct implementation without error checks	**8 to >0 pts** **Insufficient** Code does not compile	16 pts
Input Methods	**16 to >13.0 pts** **Exemplary** Correct implementation with error checks	**13 to >8.0 pts** **Moderate** Correct implementation without error checks	**8 to >0 pts** **Insufficient** Code does not compile	16 pts
Output Methods	**16 to >13.0 pts** **Exemplary** Correct implementation with error checks	**13 to >8.0 pts** **Moderate** Correct implementation without error checks	**8 to >0 pts** **Insufficient** Code does not compile	16 pts
Works with any input and provides correct output	**16 to >13.0 pts** **Exemplary** Correct implementation with error checks	**13 to >8.0 pts** **Moderate** Correct implementation without error checks	**8 to >0 pts** **Insufficient** Code does not compile	16 pts
				Total Points: 100