

CENG 202: DATA STRUCTURES

Programming Assignment 5

Deadline: WEDNESDAY, 2 June 2023 by **20:00**

Rules

1. You will use Java to implement the tasks
2. You will lose 20 points for each day your assignment is submitted late.
3. No deadline extensions will be granted.
4. Your code must be developed individually. You cannot collaborate with your peers. You cannot post your code on any public repository, nor can you share it with anybody. Any violations to this code of conduct will be treated as academic misconduct.
5. Submitted code will be automatically checked using tools that detect plagiarism.
6. In case cheating is detected, the grade of this assignment will be zero and a further penalty of 50 points will be applied to the next homework assignment.
7. You may be asked to explain your code and rewrite some parts of it in front of the instructor.
8. Submit your codes to AYBUZEM.
9. Write your name as a comment line in each file you submit.
10. In order to be accepted, your submission
 - must be of text format with a '.java' extension (no Word or other formats and no archives will be accepted)
 - You will receive a zero grade if you do not submit the assignment or you do not adhere to this rule!

Note: You can only use code/pseudo code from the slides and the textbook. You need to develop the rest of the code yourself.

What to turn in electronically:

- Your source code (*.java file).

Task:

A software company hires you to keep their products in a data structure. The company wants to find their cheapest product quickly. In addition, the company wants to reduce the price of a specific product, remove the cheapest product from sale, and add new products. They want you to implement the following operations:

ListMin

RemoveMin

Add

DecreasePrice

- There cannot be more than one product with the same name, but there may be more than one product with the same price.
- In case, the company wants to remove the item with the minimum price and if there is more than one product with the equal minimum price, then each of them must be removed.
- All these operations will be specified in a file.
- The program takes the inputs from “process.txt”

The sample file content (`process.txt`) is as follows:

ListMin

RemoveMin

DecreasePrice product1 5

Add product1 156.56

Add product2 45.00

Add product3 245.99

Add product4 99.70

RemoveMin

Add product2 20.60

ListMin

Add product5 70.17

Add product6 10.10

ListMin

RemoveMin

ListMin

DecreasePrice product1 150

ListMin

Accordingly, the operations you need to implement are explained as follows:

ListMin	-> Error, no item added yet.
RemoveMin	-> Error, no item added yet.
DecreasePrice product1 5	-> Error, no item added yet.
Add product1 156.56	-> product1 with price 156.56 added.
Add product2 45.00	-> product2 with price 45.00 added.
Add product3 245.99	-> product3 with price 245.99 added.
Add product4 99.70	-> product4 with price 99.70 added.
RemoveMin	-> product2 is removed since it has the min price
Add product2 20.60	-> product2 with price 20.60 added.
ListMin	-> product2 with price 20.60 listed (without removing).
Add product5 70.17	-> product5 with price 70.17 added.
Add product6 10.10	-> product6 with price 10.10 added.
ListMin	-> product6 with price 10.10 listed (without removing).
RemoveMin	-> product6 is removed since it has the min price
ListMin	-> product2 with price 20.60 listed (without removing).
DecreasePrice product1 150	-> product1's price is decreased by 150 (making it 6.56)
ListMin	-> product1 with price 6.56 listed (without removing).

You'll implement a binary heap for this question. Make sure `RemoveMin` and `Add` works in $O(\log n)$, where n is the number of elements in the heap. After finding the product from the data structure, `DecreasePrice` should also work in $O(\log n)$. `ListMin` must work in $O(1)$. Use arrays to implement heaps. After `RemoveMin` and `DecreasePrice` operations, make sure to update the heap and preserve the heap structure.

Write a driver program to test all methods.