CMPE 223
Spring 2023
Programming Homework 4

This assignment is due by 23:59 on Sunday, 11 June 2023.

You are welcome to ask your HW related questions. You should use only one of these options:

1. Moodle Homework **Question Forum**: HW Question-and-Answer (Q&A) Forum on Moodle is always available. Use the "Forum" link at the course Moodle page.
2. Homework **RECITATION HOURS**: There will be two Q&A RECITATION HOURs on the following days:

   - CMPE223-HW4-OfficeHour1:
     01.06.2023, 18:00-19:00, https://tedu.zoom.us/j/94452083117
   - CMPE223-HW4-OfficeHour2:

     08.06.2023, 18:00-19:00, https://tedu.zoom.us/j/91735917014

   - Note: Please make sure that you have read the HW document well before participating. However, no HW related questions will be accepted except from the above options.
   - Any reports without submitting codes to VPL will not be evaluated.
   - You should submit your report to the corresponding report submission area and codes to corresponding VPL. DO NOT SUBMIT ANY ZIP FILE TO VPL!!!

PROGRAMMING TASK(s)

Task 1

In this task, you are supposed to implement a Java program for a library by using heap-based priority queue. The library manager is trying to find a book and waiting customer information for a specific day. In our library system, you can assume that one month has always 30 days we deal with only monthly data. You have a log file consisting of three parts, as shown below:

```
***BOOK INFO**
J. K. Rowling,Harry Potter and the Goblet of Fire,3
Albert Camus,The Stranger,2
Franz Kafka,The Metamorphosis,1
**DAY INFO**
14
16
18
30
***CUSTOMER INFO***
1959,C1,12,2,Harry Potter and the Goblet of Fire
1961,C2,12,2,Harry Potter and the Goblet of Fire
1953,C3,12,2,Harry Potter and the Goblet of Fire
1953,C4,12,5,Harry Potter and the Goblet of Fire
1951,C5,13,4,Harry Potter and the Goblet of Fire
1980,C6,15,1,The Metamorphosis
1982,C7,15,3,The Metamorphosis
1982,C8,15,3,The Metamorphosis
1984,C9,18,4,The Stranger
2000,C10,19,5,The Stranger
```

Figure 1:logs1.txt

Book Info: This part has three data: Writer of Book, Name of Book and Number of Books in the library.

Day Info: This part has only one data which is a day number. You should print a list of waiting customer for these days in the desired format.

Customer Info: This part contains five data: Registration year to library, Customer ID, the start of the reservation date, total reservation day, desired book name.

Specifications of customer info part are as follows:

- The customer who registers first has a high priority.
- If two customers have the same registration year, the first come has higher priority than the other customer.
- Records are always in ascending order for the start of the reservation date.

First of all, you should read the name of this log file from the user, then you should parse it. Then, you should print customers who want to take a book but can't, based on their priority and current books in the library. for *logs1.txt*, sample run should be as follows: (Green text shows user input)

```
Enter log file name:
log1.txt
Day 14:
Customer info:
C5 waits Harry Potter and the Goblet of Fire since day 13.
C2 waits Harry Potter and the Goblet of Fire since day 12.
Book info:
J. K. Rowling,Harry Potter and the Goblet of Fire,0
Albert Camus,The Stranger,2
Franz Kafka,The Metamorphosis,1
Day 16:
Customer info:
C7 waits The Metamorphosis since day 15.
C8 waits The Metamorphosis since day 15.
Book info:
J. K. Rowling,Harry Potter and the Goblet of Fire,1
Albert Camus,The Stranger,2
Franz Kafka,The Metamorphosis,0
Day 18:
Customer info:
C8 waits The Metamorphosis since day 15.
Book info:
J. K. Rowling,Harry Potter and the Goblet of Fire,2
Albert Camus,The Stranger,2
Franz Kafka,The Metamorphosis,0
Day 30:
Customer info:
No waiting customer
Book info:
J. K. Rowling,Harry Potter and the Goblet of Fire,3
Albert Camus,The Stranger,2
Franz Kafka,The Metamorphosis,1
```

Your implementation must obey the following requirements:

- There may be at most 200 customers in the data file.
- The customer with the highest priority should be examined first.
- In case of having two customers with the same highest priority, the customer who has waited longer should be selected first.
- While printing the waiting customers and books information, if day number in DAY INFO part and the start of the reservation date + total reservation date are equal, you should accept that those books are not delivered to library yet.
- While printing the waiting customers and books information, if day number in DAY INFO part and the start of the reservation date are equal, you should accept that those books are not received by the customer yet.

In your implementation, you must use a heap-based priority queue to store customers who are waiting for a book. If you do not use such a heap-based priority queue to store these customers, then you will get no points from this task of homework.

As a hint, you use the heap data structure to hold customers that are waiting for a book and to find the customer with the highest priority. Update the heap whenever a new customer receives/delivers a desired book.

WHAT TO HAND IN

- The Java sources for your program to upload corresponding VPL.
- The Java sources should be WELL DOCUMENTED as comments, as part of your grade will be based on the level of your comments.
- You should test your Java source files on (if) available Moodle VPL environment to ensure your code solution's correctness before submitting. VPL simply tests your program's output by checking against given sample input. You should pass that task's VPL test case successfully.
- A **maximum-3 pages** PDF report document that explains your own answers for programming task 1 in a clearly readable PA report format (refer to **PA REPORT FORMAT** section). You should upload this report to corresponding submission area.
- For given task 1, only code or report submission will not be graded. In other words, you should submit both correct code solution and its related report for the task 1 in order to be graded.

PA REPORT FORMAT

A programming assignment report is a self-description of a programming assignment and your solution. Please note that if you do not have correct code solution for the task, you should not report about that since it will not be graded. The report must not be hand-written. You may use a word processor or the on-line editor of your choice and prepare as a PDF document. The report must be grammatically correct and use complete English sentences. Each report should include the following sections, in the order given:

Information (%5): This section includes your ID, name, section, assignment number information properly.

Problem Statement and Code Design (%30): Include a brief summary of the problem and/or your sub-tasks to be completed in this assignment. You should show your modular design rationale by creating a *structure chart* that indicates your top-down, stepwise refinement of the problem solution. You may create the structure chart using available graphical tools like MS PowerPoint, SmartDraw etc.

Implementation and Functionality (%40): Since you have modular source code, you should describe each sub-module (program) in this section. Each sub-module should include names and types of any input/output parameters as well as the *pseudocode* algorithm that used for completing its task. By this way, you give meaning to each chart boxes from the previous section.

Testing (%15): You should provide a tester class that is able to identify key test points of your program. This class should be able to generate additional (apart from the given sample input/output) test data for the purpose of being clear on what aspects of the solution are being tested with each set. This section should also include a description of any program *bugs* that is, tests which has incorrect results. You should write these to describe your tests, summarize your results, and argue that they cover all types of program behavior.

Final Assessments (%10): In this final section, you should briefly answer the following questions:

- What were the trouble points in completing this assignment?
- Which parts were the most challenging for you?
- What did you like about the assignment? What did you learn from it?

IMPORTANT

IMPORTANT NOTES: Do not start your homework before reading these notes!!!

1. This assignment is due by 23:59 on Sunday, June 11th.

2. You should upload your homework to Moodle before the deadline. No hardcopy submission is needed. You should upload your reports to submission part and upload your code to the related VPL.

3. The standard rules about late homework submissions apply (20 points will be deducted for each late day). Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.

4. You ARE NOT ALLOWED to modify the given method names, if it exists. However, if necessary, you may define additional data members and member functions.

5. Your classes' name MUST BE as shown in the homework description, if it exists.

6. The submissions that do not obey these rules will not be graded.

7. To increase the efficiency of the grading process as well as the readability of your code, you have to follow the following instructions about the format and general layout of your program.

8. Do not forget to write down your id, name, section, assignment number or any other information relevant to your program in the beginning of your Java files. Example:

```
//-------------------------------------------------------
// Title: Scheduler tester class
// Author: Name/Surname
// ID: 2100000000
// Section: 1
// Assignment: 4
// Description: This class tests the …
//-------------------------------------------------------
```

9. Since your codes will be checked without your observation, you should report everything about your implementation. Add detailed comments to your classes, functions, declarations etc. Make sure that you explain each function in the beginning of your function structure. Example:

```
void setVariable(char varName, int varValue)
//-----------------------------------------------------------
// Summary: Assigns a value to the variable whose
// name is given.
```

```
// Precondition: varName is a char and varValue is an
// integer
// Postcondition: The value of the variable is set.
//-------------------------------------------------------
{
      // Body of the function
}
```

10. Indentation, indentation, indentation...

11. This homework will be graded by your TA, Bedrettin Çetinkaya. Thus, you may ask him your homework related questions through HW forum on Moodle course page. You are also welcome to ask your course instructors.