Metropolitan State University, Saint Paul, Minnesota
ICS 372 Object-Oriented Design and Implementation
Exam 2 Programming Questions

Date and Time 6:00 PM (or a few minutes later) on March 25, 2021
Duration: 3 hours (maximum for this part and the multiple choice part, combined)
Points 50 (for programming questions)

Ground Rules

1. This is an open book/open notes test. You may use any resources on the internet. The only restriction is that you are not allowed to communicate in any way with anyone: like phone call, online chat, email, posing questions on online forums, etc. So, this is an individual effort.
2. **I reserve the right to ask you questions on any code you write. Your grade may change (a decrease is more likely), depending on your answer.**
3. You must have your video on. Mute the audio. If you need to ask me a question, please send a message privately on Zoom chat. I may move you to a waiting room and we can talk briefly.
4. If you have to leave the exam for more than a couple of minutes, please get permission from your instructor. Use the approach in (3) above.
5. Two of the questions ask you to write Java code. You could code them as separate Eclipse Java projects and submit to the dropbox for Exam 2. Ensure that any Java code you write is syntactically correct and works as specified. Name the projects <Your-last-name>Exam2Q1 and <Your-last-name>Exam2Q2. You could put everything into one project and call the project <Your-last0name>Exam2Q. Be sure to organize the code into clearly-named packages. The penalty for not submitting a properly-named Eclipse project is 10%. The zip file may be called almost anything; it is the project name I am really picky about.
6. Please follow the directions carefully, so you don't lose credit.
7. The exam must be turned in on time. I will announce the deadlines at the start of exam. If you are late to start the exam, this may mean a loss of time for you to take the exam. The maximum time you have to take the exam is 3 hours. After that, for every minute (or fraction of it) the submission (multiple choice or programming part) is late, you will lose 1 point. I will NOT accept exams after 9:20 PM. **So do not wait until the last minute to turn in your exam.**

The multiple-choice/true-false/multi-select/short-answer questions is under Assessments...Quizzes.

Question 1 (25 points)

The following new use case allows the actor to retrieve a member's name for a given member id.

Actor	System
1. The actor wants to know the name of a member.	

2. The actor initiates the functionality to display a member's name.	
	3. The system prompts for the member id.
4. The actor enters the member id.	
	5. The system verifies that the member exists. If there is no such member, it displays an error message. Otherwise, it displays the member name. 6. The use case ends.

If the member does not exist, the error message should be

The member you are interested in does not exist

Otherwise, the display should be

The name of member with id <id> is <name>

Design and implement the functionality. The functionality should not be achieved by depending on the results of other commands such as getting the list of books. The changes must be made to the project DathanExam2Q1 (download from the folder Exam 2 under Content). (You will have to rename the project.) The code is essentially the same as Class Project 1 Version 1.1, which we discussed in the videos/class.

The design and implementation must conform to the approach in the current version. For example, the way values are returned from Library to UserInterface should be consistent with the way it is done in similar methods already existing in the system.

The new functionality should be invokable by keying in 13. Save will now be 14 and Help 15. The help screen must incorporate the new command, and displays related to the new command must be consistent with the current system.

The code must obviously compile error free and should be properly tested. The code need NOT be documented.

A significant part of the grade would be based on conformity to the current implementation, assuming, of course, the output is correct.

Follow these additional requirements to ensure full credit.

1) Rename the project as <Your-last-name>Exam2Q1.
2) Put any new methods at the end of the classes to which they belong, so they are easy to locate. For example, if you put new methods named m1() and m2() in the Hold class, they must appear after all existing methods in Hold.

Question 2 (25 points)

The code given in the Java project DathanExam2Q2 in the folder Exam 2 under Content is an attempt to solve the following problem.

A hotel has many types of rooms. Every room is uniquely identified by a room number. Some are single rooms (normal occupancy is one person), some are double rooms (normal occupancy is 2 persons), and the rest are suites (normally 4 people are allowed in a suite). Even rooms with the same occupancy (like single rooms) are often different; for example, one may have a little more

luxury or may have a better view. The charge per night is dependent on the room type (single, double, suite) and how good the room itself is.

Every room allows extra persons (above the normal occupancy) for a fee.

a) A single room allows one extra occupant. So there could be two people in the room.
b) A double room allows up to two extra occupants. So there could be up to four people in a double room.
c) A suite allows up to 4 extra occupants. So there could be up to eight people in a suite

The Room class captures the above idea. It has the fields

1. roomNumber: stores the room number
2. type: the type of the room. (Value is one of the final fields, *SINGLE_ROOM*, *DOUBLE_ROOM*, *SUITE)*
3. basicChargePerNight: the rent for the room per night
4. extraChargePerPersonPerNight the extra charge for each additional guest in the room per night
5. totalRentPerNight: the total charge per night; depends on the basic charge, the number of extra persons, and the extra charge for any additional guest per night.
6. *SINGLE_ROOM* = 1 constant to denote a single room
7. *DOUBLE_ROOM* = 2 constant to denote a double room
8. *SUITE* = 3 constant to denote a suite

When a room is occupied, the hotel computes the charge per night for that occupancy: this depends on the room type, the normal nightly charge, any extra person(s) in the room, and the cost for each additional person per night.

What you need to do:

The approach taken by the project does not follow some object-oriented design principles. Refactor the code appropriately.

There is no need to document your code.

Do not submit any of the original code. I will assume that any file you have in the project is part of your refactoring, so it would be to your disadvantage to submit anything other than the refactored code. This will be the case, even if your code does not use the extra classes/methods.

Rename the project as <Your-last-name>Exam2Q2.