Metropolitan State University, Saint Paul, Minnesota
ICS 372 Object-Oriented Design and Implementation Exam 1

Date and Time 6:00 PM (or a few minutes later) on February 11, 2021
Duration: 3 hours (maximum)
Points 100

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. 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 breakout room and we can talk briefly.
3. If you have to leave the exam for more than a couple of minutes (like a restroom break), please get permission from your instructor. Use the approach in (2) above.
4. The questions in this document ask you to write Java code. These must be submitted as one or two **Eclipse Java** project(s) to the dropbox for Exam 1. Ensure that any Java code you write is syntactically correct and works, as specified.
5. The projects must be named <Your-last-name>Exam1Q1 or <Your-last-name>Exam1Q2 or <Your-last-name>Exam1Q (code for both questions in the same project), as applicable. If you do not submit an Eclipse Java project or the name does not closely follow the above naming conventions or use modules in your project, you will lose 10% of the grade for the question(s) involved.
6. I will not accept the exam after 9:20 PM, except from those who may have special needs. The penalty for every minute (or part thereof) taken to submit the answers, above the three-hour limit, will mean a loss of 1 point.
7. Please follow the directions carefully, so you don't lose credit.
8. The exam must be turned in on time. I will announce the deadlines at the start of exam. If you are late starting the exam, this may mean a loss of time for you to take the exam.

You don't need to document the code. But the coding standards should otherwise follow the standards specified in the respective document in the assignment folder on D2L. I will look for proper access specifiers for all class/interface members.

I will look for clean design and implementation in both solutions. Ensure cohesion, proper reuse, proper access specifiers, irredundant code, etc.

Question 1 (25 points)

A set of classes is used to handle the different ticket types for a theater. The class Ticket is as an abstract class.

a) (9 points) Write the class Ticket with exactly one instance field (you can have one static field as well) that is private int: The field should be named serialNumber. Create a no-argument constructor in this class. In the constructor, generate and store a unique value into the serialNumber field. The serial number for the first Ticket object must be 1 and the serial number must increase by 1 for every successive instances of Ticket. Create a getter for serialNumber. The class must have an abstract method named getPrice(), which returns a double. (Every ticket is required to return its price.) Note that there is no field for price. Override toString() to return a message such as the following:

```
Ticket  85  price  38.5
```

That is, the string has the word Ticket, followed by the serial number, then the word price, and then the price itself.

The additional classes are as described below.

Class	Description	Sample toString output
Advance	A type of Ticket. If purchased 10 or more days in advance, the cost is $35; if purchased less than 10 days in advance, but before the day of the event, the cost of $45. Otherwise, the cost is $50 (not really an advance ticket). These costs could change.	Ticket 34 price 35
StudentAdvance	A type of Advance and costs half of what an Advance ticket costs	Ticket 11 price 17.5

b) (8points) Write the <u>complete</u> declarations for the class Advance. Include all necessary instance variables and methods. The constructor should take as its only parameter, the number of days in advance that this ticket is being purchased. You cannot change the Ticket class.

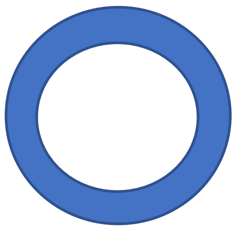
c) (8 points) Write the <u>complete</u> declarations for the class StudentAdvance. Include all necessary instance variables and methods. The constructor should take as its only parameter, the number of days in advance that this ticket is being purchased. The toString method should return not only what the Ticket class returns, but an additional reminder that student id should be produced at the gate. The methods should continue to work - with no modification necessary - even if prices are changed in the Advance class.

Remember that you must submit an Eclipse, Java project without modules. The project must be named <your-last-name>Exam1Q1. There is a 10% penalty for not following this requirement in any way. You may also put code for both questions in the same project and submit the

Question 2

Create a class named Band that is like an object adapter of Circle. I qualified the sentence with the word like, because we don't have any interface for Band or for Circle. In any case, you should use this as a hint, and not get confused.

A band looks like the following, visually. It is essentially two concentric circles (circles with the same center). Somewhat like a flattened donut or bagel. You can view it as a circle with an inner circle carved out of it. The width of the band is the difference between the radii of the outer circle and the inner circle.



In this problem, you need to create two classes:

1) Circle, with a constructor that takes in the x and y coordinates of its center and radius as parameters.
2) Band, which represents a band. It must be coded to meet the following requirements.
   a) support the following test program and produce the output given below it. The constructor parameters are x-coordinate of the center, y-coordinate of the center, band width, and band radius (radius of the outer circle), in that order. If the radius is not more than the band width, the constructor throws an exception. You can see this manifested in the last line of the display.
   b) Use the Circle class appropriately. Your grade will partly depend on how well you assign/distribute the functionality to/among Circle and Band.

```
public class BandDriver {
 public static void main(String[] args) {
    try {
        Band band1 = new Band(50, 100, 10, 30); // x = 50, y = 100, width = 10, outer radius = 30
        System.out.println(band1.getArea()); // Returns the area occupied by the band
        band1.display(); // displays the Band object as shown in the output
        Band band2 = new Band(20, 30, 40, 10); // outer radius is too small for the width
    } catch (Exception e) {
        System.out.println(e);
    }

 }
}
```

The output is

```
1570.0
```

```
Centered at (50, 100) width 10
java.lang.Exception: Illegal parameters
```

Remember that you must submit an Eclipse, Java project without modules. The project must be named <your-last-name>Exam1Q2. There is a 10% penalty for not following this requirement in any way. You may also put code for both questions in the same project and submit the project as <your-last-name>Exam1Q.