LAB #9 – Tickets: OOP & Inheritance

You will continue to use object-oriented programming (OOP) with inheritance. In this program, I will give you the verbal representation of the problem and you will need to think about the states and behavior for implementing a university ticketing system.

1. Consider the task of representing types of tickets to campus events. Each ticket has a unique number and a price. There are three types of tickets: walk-up tickets, advance tickets, student advance tickets. You need to take out a few pieces of notebook paper and rip them in half. A half of a piece of notebook paper will represent a class for this program.

Based on the requirements below, determine how many classes you need, and write the names of the classes, the states and behaviors for each class, and which classes will be superclasses versus subclasses on the half pieces of paper.

- **Walk-up tickets** are purchased the day of the event and cost $50.
- **Advance tickets** purchased 10 or more days before the event cost $30, and advance tickets purchased fewer than 10 days before the event cost $40.
- **Student advance tickets** are sold at half the price of normal advance tickets: When they are purchased 10 or more days early the cost $15, and when they are purchased fewer than 10 days early they cost $20.

All tickets must have the following properties:
- The ability to construct a ticket by number.
- The ability to ask for a ticket’s price.
- The ability to print a ticket object as a String. An example String would be “Number: 17, Price: 50.0”.

Walk-up tickets are also constructed by number, and they have a price of $50.

An advance ticket is constructed with a ticket number and with the number of days in advance that the ticket was purchased. Advance tickets purchased 10 or more days before the event cost $30, and advance tickets purchased fewer than 10 days before the event cost $40.

A student advance ticket is constructed with a ticket number and with the number of days in advance that the ticket was purchased. Student advance tickets purchased 10 or more days before the event cost $15, and student advance tickets purchased fewer than 10 days before the event cost $20 (half of a normal advance ticket). When a student advance ticket is printed, the String should mention that the student must show his/her student ID (for example, “Number: 17, Price: 15.0 (ID required)”).
2. After a TA checks your class and inheritance information written on the pieces of paper, you may begin the rest of the lab. Implement your superclass and subclasses. Define all common operations in the superclass, and specify all differing operations in such a way that every subclass must implement them. No actual objects of the superclass will be created because each actual ticket will be an object of a subclass type. Make sure your classes provide encapsulation!!!

3. Now, write a PurchaseTicket client that creates an array of the three types of tickets. When you are creating your advance and student advance tickets, you need to have a way to determine the price based on the event day and the purchase day. You can test this by using arbitrary integer numbers such as an event day of 31 and purchase day of 10. Of course you can make this much more rigorous, but we are only testing here, refer to the Extras for more.

Test the functionality of your design by printing the individual price and ticket number attributes, as well as the object containing the String information. ***I'm being vague here because part of this lab is to make sure you understand how to go about testing effectiveness of your implementation without knowing exactly how the end product will behave.***

Extras:

- Add the functionality for a more rigorous date system for the advance tickets, i.e. month and day for the event rather than an integer. Also, use today's date as a means to determine how far in advance the ticket is being purchased. Hint: Use Java API to research the Date class.

- Start on your Assignment #6, and have a good Thanksgiving break!!!