CIS 41A Lab 7: OOP

Lab 7 involves some problem solving so you can choose to work with one partner if you like. If you work with one partner, make sure both partners work together so both people learn from the lab.

Re-write your lab5.py so that it's an object oriented program and has additional features. The program still manages a theater seating chart and lets the user buys tickets for seats of their choice.

1. Create a file called seat.py.

seat.py contains a superclass named Seat and 3 subclasses named: Premium, Choice, and Regular.

These subclasses represent the 3 different types of tickets that the user can buy. Premium is the most expensive and comes with extra perks (such as a swag bag and a drink ticket, as shown in the sample output). Choice is the middle price seat and comes with a smaller extra perk (such as a drink ticket only, as shown in the sample output). Regular is the lowest price seat and doesn't come with any extra perk.

[Feel free to be creative with what the extra perks are, you are not limited to the sample shown in the output.]

* The Seat class has:
  + instance variables price and taken (a boolean).
  + a constructor that initializes the taken attribute to False and has a *default argument* price which you should provide a default value.
  + an abstract method getExtra that requires all subclasses to implement it.
* Each of the 3 subclasses Premium, Choice, and Regular has:
  + a constructor that requires a price input argument and stores it in the instance variable price by using the constructor of the superclass.
  + an implementation of the superclass getExtra method that returns a string describing the extra perk that comes with the ticket. Each type of ticket has different extra perks, as described above.
* In addition to the requirements above, there are additional methods that you will need to implement:
  + getPrice: return the value in the instance variable price
  + isTaken: return True if the seat is taken, False if the seat is still available.

It's also up to you to decide in which class the extra methods belong. The general rule of thumb is: if all 3 subclasses need to implement the same method in the same way, then the method belongs in the superclass.

2. Create a file called chart.py

chart.py contains the class Chart. This class contains the list of lists (the table) of Seat objects.

The Chart class has:

* An instance variable chart which is the list of lists of Seat objects.
* A constructor that has similar functionality to the readChart function of lab5:
  + prompt the user for a filename or enter key to use the default file lab7input2.txt
  + read in the first line of the input file, which has the 3 prices in order of: premium, choice, and regular.
  + read in the rest of the input file, and for each price, create the appropriate Seat object (Premium, Choice, or Regular) to store in the list of lists.
  + call the print method to print the seating chart with prices.
* A print method that has similar functionality to the printChart function of lab5:
  + print the row and column headers
  + print the seating chart with prices or with 'X', columns right justified (see sample output)
* A buySeat method that has the same functionality as the buySeat method of lab5:
  + in a loop:
    - prompt the user for the row, col or 0 to end the loop
    - check that the row, col values are valid, and if the seat is available, mark the seat with 'X', sum up the price, and store the row,col location in a tuple..
  + when the loop ends, print the total cost, and for each seat's row,col location, print the extra perks, and then print the resulting seating chart with 'X' for the seats that are bought. See sample output.
* Be careful in the Chart class: don't directly use private instance variables of the Seat class, use the Seat methods instead.

3. Download the file lab7.py that has the test driver code to create a Chart object and run the buySeat method in a loop. The lab7.py file should run as is, with no modification needed.

4. Don't forget to:

* document each method with a docstring
* make sure all instance variables are private
* for each class, make sure to only have the instance variables that are described above. All other variables are local variables. (Why?)

Upload to Canvas both chart.py and seat.py. No need to upload lab7.py

Sample output:

Enter file name or hit Enter for default lab7input2.txt: # enter key

Price chart

Column

1 2 3 4 5 6 7 # same chart format as lab 5

Row ===================================

1 | $10 $10 $20 $20 $20 $10 $10

2 | $5 $10 $20 $20 $20 $10 $5

3 | $5 $5 $10 $10 $10 $5 $5

4 | $5 $5 $5 $5 $5 $5 $5

Available seats are shown with price

Enter row,col for seat 1 or enter 0 to end: a,b # non-numeric input, same as lab 5

Invalid row or column

Enter row,col for seat 1 or enter 0 to end: 8, 10 # invalid input, same as lab 5

Invalid row or column

Enter row,col for seat 1 or enter 0 to end: 2, 3

Enter row,col for seat 2 or enter 0 to end: 2, 3 # seat taken, same as lab 5

Sorry, that seat is not available.

Enter row,col for seat 2 or enter 0 to end: 2, -3 # invalid input, same as lab 5

Invalid row or column

Enter row,col for seat 2 or enter 0 to end: 2,4 # buy 2 more seats

Enter row,col for seat 3 or enter 0 to end: 2,2

Enter row,col for seat 4 or enter 0 to end: 0

Your total: $50 # print total, same as lab 5

For your 3 seat(s) at:

Row 2 column 3: your swag bag and drink ticket are at will call # new: print extra

Row 2 column 4: your swag bag and drink ticket are at will call # perks

Row 2 column 2: your drink ticket is at will call

Your seats are marked with 'X' below # print resulting chart, same as lab 5

Price chart

Column

1 2 3 4 5 6 7

Row ===================================

1 | $10 $10 $20 $20 $20 $10 $10

2 | $5 X X X $20 $10 $5

3 | $5 $5 $10 $10 $10 $5 $5

4 | $5 $5 $5 $5 $5 $5 $5

Continue to buy seats? y/n: y # loop from lab7.py

Available seats are shown with price

Enter row,col for seat 1 or enter 0 to end: 2,3

Sorry, that seat is not available.

Enter row,col for seat 1 or enter 0 to end: -2,2

Invalid row or column

Enter row,col for seat 1 or enter 0 to end: 4,4 # buy 1 seat

Enter row,col for seat 2 or enter 0 to end: 0

Your total: $5

For your 1 seat(s) at: # no extra perk for Regular price seat

Row 4 column 4: drinks are available for purchase at intermission

Your seats are marked with 'X' below

Price chart

Column

1 2 3 4 5 6 7

Row ===================================

1 | $10 $10 $20 $20 $20 $10 $10

2 | $5 X X X $20 $10 $5

3 | $5 $5 $10 $10 $10 $5 $5

4 | $5 $5 $5 X $5 $5 $5

Continue to buy seats? y/n: n

Second test case:

Enter file name or hit Enter for default lab7input2.txt: lab7.txt # invalid file

Can't open lab7.txt

Enter file name or hit Enter for default lab7input2.txt: lab7input1.txt # chosen file

Price chart

Column

1 2 3 4 5 6 7 8 9 10

Row ==================================================

1 | $40 $40 $50 $50 $50 $50 $50 $50 $40 $40

2 | $40 $40 $50 $50 $50 $50 $50 $50 $40 $40

3 | $30 $40 $40 $50 $50 $50 $50 $40 $40 $30

4 | $30 $40 $40 $50 $50 $50 $50 $40 $40 $30

5 | $30 $30 $30 $40 $40 $40 $40 $30 $30 $30

6 | $30 $30 $30 $40 $40 $40 $40 $30 $30 $30

7 | $30 $30 $30 $30 $40 $40 $30 $30 $30 $30

8 | $30 $30 $30 $30 $40 $40 $30 $30 $30 $30

9 | $30 $30 $30 $30 $30 $30 $30 $30 $30 $30

Available seats are shown with price

Enter row,col for seat 1 or enter 0 to end: 8,8

Enter row,col for seat 2 or enter 0 to end: 0

Your total: $30

For your 1 seat(s) at:

Row 8 column 8: drinks are available for purchase at intermission

Your seats are marked with 'X' below

Price chart

Column

1 2 3 4 5 6 7 8 9 10

Row ==================================================

1 | $40 $40 $50 $50 $50 $50 $50 $50 $40 $40

2 | $40 $40 $50 $50 $50 $50 $50 $50 $40 $40

3 | $30 $40 $40 $50 $50 $50 $50 $40 $40 $30

4 | $30 $40 $40 $50 $50 $50 $50 $40 $40 $30

5 | $30 $30 $30 $40 $40 $40 $40 $30 $30 $30

6 | $30 $30 $30 $40 $40 $40 $40 $30 $30 $30

7 | $30 $30 $30 $30 $40 $40 $30 $30 $30 $30

8 | $30 $30 $30 $30 $40 $40 $30 X $30 $30

9 | $30 $30 $30 $30 $30 $30 $30 $30 $30 $30

Continue to buy seats? y/n: n