CSE1322L Assignment 6

Objective:

Airplanes sell tickets based on cabin class. First class tickets are more expensive, and come with perks. Business class is slightly cheaper than first, but has less perks. Economy is cheaper still but comes with almost no perks, and finally Basic Economy has no perks, but is cheapest.

For this assignment you'll write a very simple booking system for an airline.

Assignment Requirements:

Write a class called Passenger. Each passenger will have a Name (string), number of carryOnBags (int) and number of checkedBags (int). It must have a constructor which takes in a name, number of carry on bags, and number of checked bags and sets the object variables. Add a toString (Java) or ToString (C#) override which returns a string like "Bob has 3 carry on bags and 2 checked bags".

Next, write a class Seat which is abstract. It should have private attributes to hold the base price of a seat (int), cost of a carry on bag (int), cost of first checked bag (int), cost of second checked bag (int), cost of additional bags (int), a passenger, and the total price the passenger paid.

It must have a constructor which takes in a Passenger, and ticket type as a string (First class, Business, etc). Set the passenger in the object to the passenger that was passed in. If the cabin is "First", the base price is $1000, all bags are $0. If the cabin is "Business", the base price is $750, carry on bags, first and second checked bags are $0, but additional bags are $50. If the cabin is "Economy", the base price is $400, carry on bags and first checked bags are $0, but second and additional bags are $50 each. Finally if the cabin is "Basic Economy", the base price is $200, carry ons, and first bag are $50, second and additional bags are $100.

In the constructor you'll need to calculate the cost of this seat. Count both the base price of the ticket as well as the number of carry ons and checked bag, and set the total price the passenger paid.

Finally, the Seat class must have an override of toString (Java) or ToString (C#), which returns the details of the Passenter followed by " and paid $250" where $250 is the total cost of the ticket.

Create four concrete classes (FirstClassSeat, BusinessSeat, EconomySeat, and BasicEconomySeat) each of which inherits from Seat, and has a constructor that takes a Passenger and calls the parent's constructor as appropriate.

In the main driver method you'll need to do the following:

1) Create an ArrayList (Java) or List (C#) of Seats
2) Using a loop, Print out a menu asking the user which type of ticket they wish to sell.
3) Read in the users choice
4) Ask how many carry on bags the passenger has.
5) Ask how many checked bags the passenger has.
6) Ask the passengers name.

7) Create a new Passenger object using the data you just received.
8) Add to the ArrayList/List the new Seat of the appropriate type based on what the user chooses.
9) Continue to allow the sale of tickets until the user selects 5. Then print the manifest of the plane. See sample output below.

Sample Output:

What type of ticket do you wish to sell
1. BasicEconomy
2. Economy Class
3. Business Class
4. First Class
5. Quit

4
How many carry on bags?
1
How many checked bags?
0
Passenger Name?
Jane
What type of ticket do you wish to sell
1. BasicEconomy
2. Economy Class
3. Business Class
4. First Class
5. Quit

3
How many carry on bags?
1
How many checked bags?
3
Passenger Name?
Paul
What type of ticket do you wish to sell
1. BasicEconomy
2. Economy Class
3. Business Class
4. First Class
5. Quit

2
How many carry on bags?
1
How many checked bags?
2
Passenger Name?
Maeve

What type of ticket do you wish to sell
1. BasicEconomy
2. Economy Class
3. Business Class
4. First Class
5. Quit

1
How many carry on bags?
1
How many checked bags?
0
Passenger Name?
Tim
What type of ticket do you wish to sell
1. BasicEconomy
2. Economy Class
3. Business Class
4. First Class
5. Quit

5
Manifest:
Jane has 1 carry on bags and 0 checked bags. and paid $1000
Paul has 1 carry on bags and 3 checked bags. and paid $800
Maeve has 1 carry on bags and 2 checked bags. and paid $450
Tim has 1 carry on bags and 0 checked bags. and paid $250

Submitting your answer:

Program code - 6 Classes and a driver program

Please follow the posted submission guidelines here:
https://ccse.kennesaw.edu/fye/submissionguidelines.php

Ensure you submit before the deadline listed on the lab schedule for CSE1322L here:
https://ccse.kennesaw.edu/fye/courseschedules.php

Rubric:

Passenger class should (15 points total)
- Name, carryOnBags, and checkedBags public attributes (2 points each)
- Constructor takes in name, carry ons and checked bags, and sets object variables (5 points)
- toString/ToString override correctly returns string (4 points)

Seat class (45 points total):
- Private attributes to hold baseTicketPrice, carryon cost, first bag cost, second bag cost, additional bag costs, the passenger and the total price of the seat (2 points each)

- Constructor which takes in a passenger and the ticket type (“First”, “Business”, “Economy”, or “BasicEconomy”). (5 points for header)
- Conditional that looks at the ticket type and sets base price, carryon cost, first checked bag cost, second checked bag cost and additional bag cost (20 points)
- Logic that calculates the cost of bags successfully (10 points)
- Sets the Person attribute, and the totalTicketPrice attributes (3 points)
- Override of toString (Java) or ToString (C#) (5 points)

FirstClassSeat class (5 points total)
- Correctly calls parent constructor (3 points)

BusinessSeat class (5 points total)
- Correctly calls parent constructor (3 points)

EconomySeat class (5 points total)
- Correctly calls parent constructor (3 points)

BasicEconomySeat class (5 points total)
- Correctly calls parent constructor (3 points)

Driver Program (20 points total)
- Creates ArrayList/List of seats (3 points)
- Prompts the user with the menu correctly, reads result (3 points)
- Prompts the user for number of carry ons, checked bags and the name of the passenger (6 points)
- Adds the appropriate seat object to the arraylist (3 points)
- Correctly prints out the manifest of the plane (5 points)