Ege University Electrical&Electronics Engineering

Object-Oriented Programming Midterm Group A

In the automation program to be written for a bus company, write the interface and code implementation of the Bus class, which are required to model buses, in separate files in accordance with the driver program below. The UML diagram of the class to be designed is shown below. Just fill in the blanks without rewriting the main function.

Bus
- plaque : String[1] - brandModel: String[2] - ticketFee : Integer[3] - seats[30] : Integer[4]
+ <<constructor>>Bus(p:String, m:String, b:Integer)[5] + getVacantSeats( ) : Integer[6] + moreVacant(b:Bus) : Bus[7] + operator ==(b:Bus) : Boolean[8] + setPassenger(number:Integer)[9] + firstVacantSeat( ): Integer[10] <<friend>> ostream& operator << (ostream&, const Bus&)[11]

[1]: Holds the bus license plate.

[2]: Holds bus brand and model.

[3]: The ticket price of the bus.

[4]: It represents the seats in the bus. There is a maximum of 30 seats. 0 for those whose tickets are not sold, and ticket fees to those sold.

[5]: It transfers the sent plate, brand-model and ticket fee to the relevant members. It transfers zero (0) to all seats. It passes null values to all members by default. It transfers zero to numeric ones and spaces ("") to literal ones.

[6]: Returns the number of unsold empty seats on the bus.

[7]: Returns the one with more seats available from the two buses (the one who called the method and the one that is the parameter / sent). Returns any if the values are the same.

[8]: It compares the total revenue (total ticket sales) of the two buses (who called the method and the parameter / sent). Returns true if same, false if different.

[9]: According to the number of people sent, it makes the assignment from the first empty seat to the back seat. It does not make any changes if the desired number of vacant seats is not available.

[10]: Returns the number of the first vacant seat on the bus. Returns zero if there are no seats available.

[11]: Writes bus plate, brand-model, and seat sales status to the screen in the output format shown.

SAMPLE DRIVER PROGRAM

```
#include <iostream>
using namespace std;
…………………// include the class's header file in the code
int main()
{
Bus a(“06J346”, “Mercedes Travego”, 25);
Bus b(“67J879”, “MAN Neoplan”, 50), c;
........; //Assign 6 people to 06J346
........; //Assign 3 people to 67J879
........; //Print 06J346's information
cout << "Number of empty seats in 06J346: ";
.........;
//Which is the first empty seat number in 06J346?
cout << "First empty seat in 06J346: " << endl;
.........;
//Is 06J346 equal to 67J879?
if ( ……….   )
 cout << " 06J346 and 67J879 equal " << endl;
else
 cout << " 06J346 and 67J879 different " << endl;
c = ………………………………;
// With 06J346, 67J879 transfer the more vacant one to c
cout << "The more vacant:" << endl;
………………………………// Print c's information
system("pause");
}
```

SAMPLE PROGRAM OUTPUT

```
06J346, Mercedes Travego

Seats:

25|25|25|25|25|0|0|0|0|0|0|0|0|

0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|

Number of empty seats in 06J346: 25

First empty seat in 06J346: 6

06J346 and 67J879 equal

The more vacant:

67J879, MAN Neoplan

Seats:

50|50|50|0|0|0|0|0|0|0|0|0|0|

0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
```