LAB TASKS

TASK – 01:

Create a class 'Employee' having two data members 'EmployeeName'(char*) and 'EmployeeId' (int).Keep both data members private. Create three initialized objects 'Employee1', 'Employee2' and 'Employee3' of type 'Employee' in such a way that the employee name for each employee can be changed when required but the employee Id for each employee must be initialized only once and should remain same always. Use member initializer list, accessors and mutators for appropriate data members. The result must be displayed by calling the accessors. All of the accessors must not have the ability to modify the data.

TASK – 02:

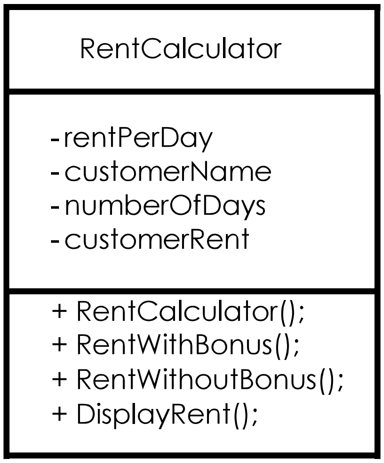
"Hotel Mercato" requires a system module that will help the hotel to calculate the rent of the customers. You are required to develop one module of the system according to the following requirements:

1) The hotel wants such a system that should have the feature to change the implementation independently of the interface. This will help when dealing with changing requirements.
2) The hotel charges each customer 1000.85/- per day. This amount is being decided by the hotel committee and cannot be changed fulfilling certain complex formalities.
3) The module should take the customer's name and number of days, the customer has stayed in the hotel as arguments in the constructor. The customer name must be initialized only once when the constructor is called. Any further attempts to change the customer's name should fail.
4) The module then analyses the number of days. If the customer has stayed for more than a week in the hotel
, he gets discount on the rent. Otherwise, he is being charged normally.
5) The discounted rent is being calculated after subtracting one day from the total number of days.
6) In the end, the module displays the following details:
    a. Customer Name
    b. Days
    c. Rent

Note that, the function used for displaying purpose must not have the ability to modify any data member.

INSTRUCTIONS

- The following class structure must be followed:



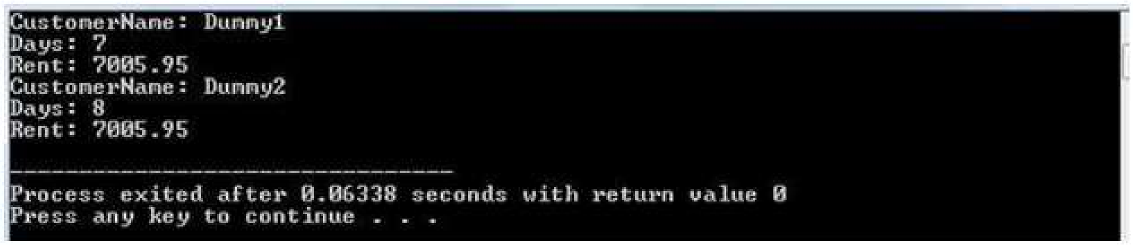
RentCalculator

-rentPerDay
-customerName
-numberOfDays
-customerRent

+ RentCalculator();
+ RentWithBonus();
+ RentWithoutBonus();
+ DisplayRent();

- Use appropriate data types, return types and function arguments.
- Display the results for two initialize distances.

REQUIRED OUTPUT



```
CustomerName: Dummy1
Days: 7
Rent: 7005.95
CustomerName: Dummy2
Days: 8
Rent: 7005.95

--------------------------------
Process exited after 0.06338 seconds with return value 0
Press any key to continue . . .
```

TASK – 03:

Define a class to represent a Bank account. Include the following members.

Data members: -

1. Name of the depositor
2. Account number.
3. Type of account.
4. Balance amount in the account.
5. Rate of interest

Provide a default constructor, a parameterized constructor to this class. Also provide Member Functions: -

1. To deposit amount.
2. To withdraw amount after checking for minimum balance.
3. To display all the details of an accountholder.
4. Display rate of interest.

Illustrate all the constructors as well as all the methods by defining objects.