**Case Study: IRCTC Ticket Booking System**

Implement IRCTC Ticket Booking System which helps user to book tickets quickly.

User can do different functionalities like search, calculateFare and display.

Create **enum** called ***Tavelclass*** with different travelclass types as

* AC2TIER
* FIRST
* SLEEPER
* AC3TIER
* AC3ECONOMY

Define a class ***Train*** to implement with following description:

* ***Data members***

1. train\_name
2. train\_number
3. available number of seats
4. travelclass of enum type
5. fromplace [can be only New Delhi ]
6. toplace [ can be Kolhapur,Surat,Nagpur,Ranchi,Dehradun]
7. distance [ based on from place and toplace]

Available train routes are

* New Delhi To Kolhapur [ Distance: 1720 km]
* New Delhi To Surat [ Distance: 1205 km]
* New Delhi To Nagpur [ Distance: 980 km]
* New Delhi To Ranchi [ Distance: 2100 km]
* New Delhi To Dehradun [ Distance: 300 km]
* ***Functions***

1. Default constructor
2. display function
3. booktrain function
4. searchTrain() as a global function which takes two arguments called train number and array of Train objects. It will display details if the train is found, or a message indicating that no train found.
5. calculateFare() function based on distance between from place and to place.

Fare would be calculated as mentioned below:

* distance >100 AND distance<500 then fare should be 800Rs.
* distance >=500 AND distance<1000 then fare should be 900Rs.
* distance >=1000 AND distance<1500 then fare should be 1100Rs.
* distance >=1500 AND distance<3000 then fare should be 1300Rs.
* Distance>=3000 then fare should be 2000

* ***Client Code***
* Create an array of 5 train objects.
* Allow user to book train and store that train information in above array.
* Call searchTrain() function.
* Call calculateFare() function to display fare value.
* Display information of all available booked trains.

**Notes**:

1. Follow best coding practices to solve this coding problem like variablename,classname,function name,take care of memory related issues.
2. Code should be implemented in 3 files.
3. Add comments where necessary.
4. You will be evaluated based on correctness of code, completion of requirements, readability among other parameters.

**Case Study: Vehicle Inventory Management System**

A major vehicle dealer wants to develop a system to track the inventory of vehicles. After discussion among major stakeholders, following major business requirements have been outlined.

* Create a class called **Car**
* **Car** category may be of types **SUV, HatchBack, Sedan**
* There should be a provision to store Chassis Number, Vehicle Brand, price, numberofseats, tax and insurance amount and category.
* Implement functions like constructor, accept() and display function.
* Overload << operator to display car details.
* **Create a class called BookVehicle**
* It has data member called pointer to Car class so that allocation of Car object can be managed by this class.
* Implement default constructor.
* Destructor
* Overload -> and \* operator in this class.
* **Client code**
* System should allow creation of new car booking through BookVehicle class and display car details.
* Create single object of Car and display it using << operator.
* Also create an array of 3 Car objects called “inventory”.Implement calculateAvg () as a global function to get average price of vehicles of a particular category stored in the inventory. Pass above array and category to this function.

**Notes**:

1. Follow best coding practices to solve this coding problem like variablename,classname,function name,take care of memory related issues.
2. Code should be implemented in 3 files.
3. Add comments where necessary.
4. You will be evaluated based on correctness of code, completion of requirements, readability among other parameters.

**Case Study: Bill and Budget System**

The Bill and Budget system is a software application that offers a range of functionalities to help individuals and businesses manage their financial transactions effectively. With features such as electricity bills, water bills. The bill and budget system can help users manage their budgets by providing them with a comprehensive overview of their expenses and income.

* Create enum called ElectricitySlabs with three slabs such as
* E1=125
* E2= 150
* E3=200
* Create enum called WaterSlabs with three slabs such as
* W1=10
* W2= 15
* Create class called Electricity with following description.
* Data Members
* SanctionLoad of type integer[ can be 1,3,5]
* PresentReading of type long
* PreviousReading of type long
* Functions

i) implement get and set function

ii) calculateElectricityBill() function which will calculate and return electricity bill for a month with below information:

If present reading is less than past reading, then bill amount would be zero.

Consider Consumption as difference between present reading and past reading.

Based on value of sactionload 1,3,5 calculate bill as

(SanctionLoad \* E1) + (Consumption \* E1 / 100)

(SanctionLoad \* E2) + (Consumption \* E2 / 100)

(SanctionLoad \* E3) + (Consumption \* E3 / 100)

* Create class called Water with following description.
* Data Members
* PresentReading of type long
* PreviousReading of type long
* Functions

i) implement get and set function

ii) calculateWaterBill() function which will calculate and return Water bill for a month with below information:

If present reading is less than past reading, then bill amount would be zero.

Consider Consumption as difference between present reading and past reading.

Based on value of Consumption calculate amount as

If Consumption is less than 1000 then amount is

amount=100 + Consumption \* 0.010

Otherwise, amount would be

amount=150 + Consumption \* 0.015

Final Waterbill would be

WaterBill = (W1 \* 10) + (amount \* W2 ) / 1000

* Client Code

Create object of Electricity on heap and calculate and display Electricity bill.

Create object of Water on stack and calculate and display Water bill.

**Notes**:

1. Follow best coding practices to solve this coding problem like variablename,classname,function name,take care of memory related issues.
2. Code should be implemented in 3 files.
3. Add comments where necessary.
4. You will be evaluated based on correctness of code, completion of requirements, readability among other parameters.