Car rental service

You have to create a „very simplified version” of a car rental service for year 2018.

Create a **car** class:

- members:
  - number plate – string
  - year (when it was built) – integer
  - fuel type – can only be: gas, diesel, hybrid, electric
  - kms (how many kilometers the car ran since it was assembled) – float
  - rental price per day – integer
- constructors:
  - default constructor – set everything to a default value
  - 5 parameters passed – check if fuel type is a valid value
- member functions:
  - getter and setter functions
- operator overloading:
  - + operator: adds an integer parameter to the kms data member

Create a **bus** class: this should be derived from the car class

- member:
  - capacity – integer
- constructor(s)

Create a **rental** class:

- members:
  - start date (as days passed since Jan 1) – integer
  - end date (as days passed since Jan 1) – integer
  - the rented car – car object
- constructors:
  - 3 parameters – check if end date is the same or after start date
  - 8 parameters – call the car constructor for last seven parameters
  - default constructor
- member functions
  - getter and setter functions
  - lenght_of_rental – calculates the length of rental from start and end date
  - price – calculates the rental price from the length of rental and daily rental price of the car

Your main program should do the following:

1. Read the available cars for rental from a file into a vector.
2. Read the available buses for rental from a file into a vector.
3. Read the rentals that took place this year into another vector (cars and buses rentals are stored in the same file).
4. Show an example for the working of the following functions and operators:
   a.) lenght_of_rental
   b.) price
   c.) operator +
5. Answer **two** of the following questions:
   a.) How much income did the company have by renting out the cars? (We do not bother about expenses in this task.)

b.) For each car, give the count of days they were rented out.
c.) Ask the user for a particular day, and tell which cars were under rent that day.
d.) How many times were electric cars rented?
e.) Which day was the bus with highest capacity rented? (if more than one, give the first day)
f.) Which day were the most vehicles rented?
g.) Which car rental cost the most?
h.) Which car rental cost the least?
i.) Which bus rental cost the most?
j.) Which bus rental cost the least?
k.) List all the car rentals in increasing order of start dates.
l.) List all the bus rentals in increasing order of start dates.

You can assume that the input files are correct: there are no overlapping dates for a given car/bus in rentals, the same number plate does not occur twice, and the rentals input file does not contain non-existing cars/buses.

Make your code effective and understandable: write comments, use while loop (instead of for) if it is possible, use error handling, ...

Use separated main, class and header files. Upload your project directory with test input and output files in zipped format.