Create a base class: **“Vehicle.hpp”** and **“Vehicle.cpp”**

The Vehicle class will have these member variables which should be **private**:

```
std::string name_;
std::string manufacturer_;
int wheels_;
int gas_;
int passengers_;
int cargo_;
```

The Vehicle class will have these member functions, all **public**:

Vehicle( ); *//This default constructor should set all member variables to either “ “ or 0 depending on their type*

Vehicle(std::string name, std::string manufacturer, int wheels, int gas, int passengers, int cargo); *//Sets all member variables to what was passed in the parameters*

std::string getName( ) const; *//Should return the name_ variable*
void setName(std::string name); *//Should set the name_ variable*

std::string getManufacturer( ) const; *//Should return the manufacturer_ variable*
void setManufacturer(std::string manufacturer); *//Should set the manufacturer_ variable*

int getWheels( ) const; *//Should return the wheels_ variable*
void setWheels(int wheels); *//Should set the wheels_ variable*

int getGas( ) const; *//Should return the gas_ variable*
void setGas(int gas); *//Should set the gas_ variable*

int getPassengers( ) const; *//Should return the passengers_ variable*
void setPassengers(int passengers); *//Should set the passengers_ variable*

int getCargo( ) const; *//Should return the cargo_ variable*
void setCargo(int cargo); *//Should set the cargo_ variable*

Create sub class: **“Car.hpp”** and **“Car.cpp”**

The Car class represents an Vehicle object with the following specifications:

- It has 4 wheels
- Can hold 5 passengers
- Can hold 12 litres of gas
- Can carry 850 kg

The Car class contains the following methods, knowing the information above and that each constructor must call the Vehicle parameterized constructor:

Car( ); *// name and manufacturer of this vehicle should be “ “ since no value was passed*

Car(std::string name, std::string manufacturer);

void addBumperStickers(std::string sticker); *// adds a string to list_of_bumper_stickers_*

void getBumperStickers( ); *// prints out each sticker on its own line from list_of_bumper_stickers_*

For the **private** member, it should only contain a vector of strings, that will be named as:

```
vector<std::string> list_of_bumper_stickers_;
```

Create another sub class: **“Motorcycle.hpp”** and **“Motorcycle.cpp”**

The Motorcycle class represents an Vehicle object with the following specifications:

- It has 2 wheels
- Can hold 2 passengers

- Can hold 1 litre of gas
- Can carry 0 kg

The Motorcycle class must contain the following methods, knowing the information above and that each constructor must call the Vehicle parameterized constructor:
Motorcycle( ); *// name and manufacturer of this vehicle should be " " since no value was passed*

Motorcycle(std::string name, std::string manufacturer);

void toggleSideMotorcycle( );
```
/*
    if sideMotorcycle_ is false, change wheels amount from 2 to 4
        and passenger amount from 2 to 3 and set sideMotorcycle_ to true
    if sideMotorcycle_ is true, change wheels amount from 4 to 2
        and passenger amount from 3 to 2 and set sideMotorcycle_ to false
*/
```

bool getSideMotorcycle( ); *// returns the current state of sideMotorcycle_*

For the **private** member, you must have a variable that represents if the motorcycle has a side car:
bool sideMotorcycle_;

Create another sub class: **"Truck.hpp"** & **"Truck.cpp"**

The Truck class represents an Vehicle object with the following specifications:
- It has 16 wheels
- Can hold 3 passengers
- Can hold 125 litres of gas
- Can carry 80000 kg

The Truck class must contain the following methods, knowing the information above and that each constructor must call the Vehicle parameterized constructor:
Truck( ); *// name and manufacturer of this vehicle should be " " since no value was passed*

Truck(std::string name, std::string manufacturer);

void toggleTrailer( );
```
/*
    if hasTrailer_ is false, change wheels amount from 16 to 26
        and cargo amount from 80000 to 16000 and set hasTrailer_ to true
    if hasTrailer_ is true, change wheels amount from 26 to 16
        and cargo amount from 16000 to 80000 and set hasTrailer_ to false
*/
```

bool getTrailer( ); *// returns the current state of hasTrailer_*

For the **private** member:
bool hasTrailer_;

Submission:
You will submit the following files:

Vehicle.cpp	Vehicle.hpp
Car.cpp	Car.hpp
Motorcycle.cpp	Motorcycle.hpp
Truck.cpp	Truck.hpp

Testing: Create your own main function to test those files!

How to compile:
g++ Vehicle.cpp Car.cpp Motorcycle.cpp Truck.cpp <test main file> -std=c++17
