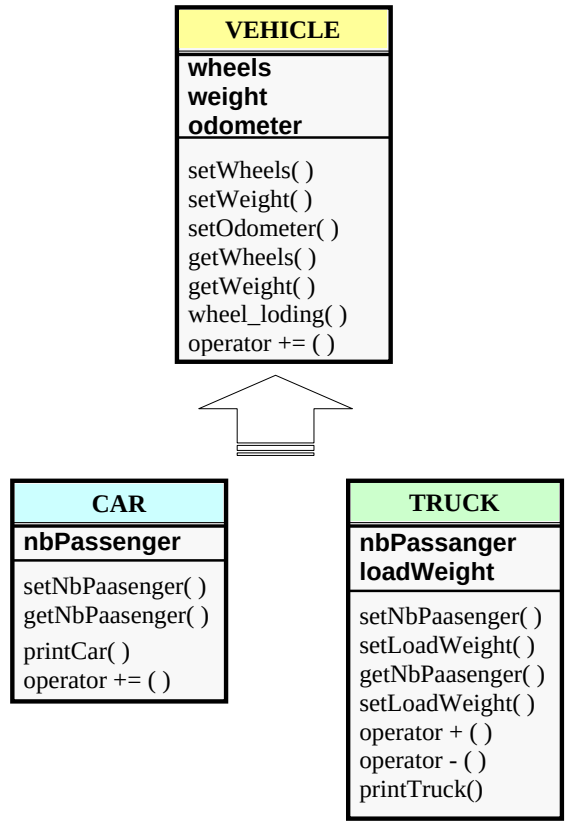
Having the need to keep some information regarding motorized vehicles, design and implement the class(es) to fulfill the following data.

- Number of wheels.
- The weight.
- The odometer reading.
- The number of passengers.
- The load weight for vehicles with cargo bay.

The motorized vehicles can be a motorcycle, a car or a truck.

Create the class(es) and all the needed constructors, getters and setters. Use your judgment for the defaults arguments.
Include in your design the following member functions:

- Operator += that increment the odometer by a certain value.
- Operator + that add a certain value from the “load weight”.
- Operator – that subtract a certain value from the “load weight”.
- Operator << that print the data value of the base object.
- Operator >> that read from the keyboard and set the value of the odometer.
- A print function that prints the data value of the derived object.
- A function that returns the weight supported (wheel loading) by a wheel of a vehicle.
  - Wheel_loading = weight / number_wheels
- A function that returns the load efficiency of a truck.
  - Load_efficiency = load_weight / (load_weight + weight)



vehicle.h

```
#ifndef VEHICLE_H
#define VEHICLE_H
#include <iostream.h>

class Vehicle {
   friend istream & operator >>(istream &,Vehicle &);
   friend ostream & operator <<(ostream &,Vehicle &);
  public:
   Vehicle(int =0, float =0.0, float =0.0);
   int getWheels(void) const ;
   float getWeight(void) const ;
   float wheel_loading(void) const ;
   void setWheels(int);
   void setWeight(float);
   void setOdometer(float);

   void operator +=(float);

  protected:
   int wheels;
   float weight;
   float odometer;
};
```

car.h

```
#ifndef CAR_H
#define CAR_H
class Car :  public Vehicle {
  public:
   Car(int, int, float, float);
   void setNbPassenger(int);
```



```
   int getNbPassenger(void) const;
   void printCar() const;
   void operator +=(float);

  private:
   int nbPassenger;
};
#endif
```

truck.h

```
#ifndef TRUCK_H
#define TRUCK_H
class Truck : public Vehicle {
   public:
      Truck(int, float, int, float, float);
      void setNbPassenger(int);
      void setLoadWeight(float);
      int getNbPassenger(void) const;
      float getLoadWeight(void) const;
      Truck operator -(float);
      Truck operator +(float);
      void printTruck() const;

   private:
      float efficiency(void)  const;

      int nbPassenger;
      float loadWeight;
};
   #endif
```

vehicle.cpp

```
#include <iostream.h>
#include "vehicle.h"

Vehicle::Vehicle(int wls, float wgt, float odo){
   setWheels(wls);
   setWeight(wgt);
   setOdometer(odo);
};

void Vehicle::setWheels(int wls){
   wheels = wls > 0 ? wls : 0;
};

void Vehicle::setWeight(float wgt){
   weight = wgt > 0 ? wgt : 0.0;
};

void Vehicle::setOdometer(float odo){
   odometer = odo > 0 ? odo : 0.0;
};

int Vehicle::getWheels(void) const {
   return wheels;
}

float Vehicle::getWeight(void) const {
   return weight;
}

float Vehicle::wheel_loading(void) const {
   return weight/wheels;
}

void Vehicle::operator +=(float by) {
   if (by > 0) odometer +=by;
}
```



```
istream & operator >>(istream &In, Vehicle &vObj){
   float vTmp;

   cout <<"Current Odometer is: " << vObj.odometer;
   cout << "\nEnter new Odometer: ";

   In >> vTmp;

   if (vTmp > vObj.odometer)
      vObj.odometer = vTmp;

   return In;
}

ostream & operator <<(ostream &Out,Vehicle &vObj){
   Out << "This vehicle has: " << vObj.wheels
       << " wheels\n";
   Out << "It weight: " << vObj.weight << " Kg\n";
   Out << "And ran for: " << vObj.odometer
       << " Km\n";
   return Out;
}
```


car.cpp

```
#include "vehicle.h"
#include "car.h"

Car::Car(int nbPass, int wls, float wgt, float odo)
                     : Vehicle(wls, wgt, odo) {
   setNbPassenger(nbPass) ;
}

void Car::setNbPassenger(int nbPass) {
   nbPassenger = nbPass > 0 ? nbPass : 0;
}

int Car::getNbPassenger(void) const {
   return nbPassenger;
}

void Car::printCar() const {
   cout << nbPassenger << " Passengers car\n";
   cout << static_cast<Vehicle>(*this);
}

void Car::operator +=(float by) {
   Vehicle &tmpRef =(*this);

   cout << "Your last odometer reading is: "
        << odometer << endl;
    tmpRef +=by;
}
```


truck.cpp

```
#include "vehicle.h"
#include "truck.h"

Truck::Truck(int nbPass, float ldWgt, int wls,
                     float wgt, float odo)
                     : Vehicle(wls, wgt, odo) {

   setNbPassenger(nbPass);
   setLoadWeight(ldWgt);
}

void Truck::setNbPassenger(int nbPass) {
   nbPassenger = nbPass > 0 ? nbPass : 0;
}

void Truck::setLoadWeight(float ldWgt) {
   loadWeight = ldWgt > 0 ? ldWgt : 0.0;
}

int Truck::getNbPassenger(void) const {
   return nbPassenger;
}

float Truck::getLoadWeight(void) const {
   return loadWeight;
}

float Truck::efficiency() const {
   return loadWeight / (loadWeight + weight);
}
```

```
void Truck::printTruck() const {
   cout <<"This truck load is: " << loadWeight << endl
      << "It weigts: "<< weight << endl
      << "Efficiency value is: " << efficiency() << endl;
}

Truck Truck::operator -(float lWgt) {
   if (lWgt < loadWeight)
      loadWeight = loadWeight - lWgt;
   return (*this);
}
Truck Truck::operator +(float lWgt) {
   loadWeight = loadWeight + lWgt;
   return (*this);
}
```


Prog.cpp

```
#include <iostream.h>
#include "vehicle.h"
#include "car.h"
#include "truck.h"

void main() {
   Vehicle *vehPtr;
   Truck *trkPtr;
   Vehicle motorcycle(2, 98.7, 3421.3);
   Car fiat(4, 4, 1800, 76213.2);
   Truck volvo(2, 5000, 10, 2000, 32123.5);

   cout << motorcycle.getWheels() << endl;
   motorcycle +=1123.5;
   cout << endl << motorcycle << endl;

   fiat.printCar();
   fiat +=50.0;
   cout << endl;
   vehPtr = &fiat;
   cout << *vehPtr;

   volvo.printTruck();
   volvo = volvo + 534.5;
   volvo = volvo - 100.5;
   cout << endl;
   volvo.printTruck();
```



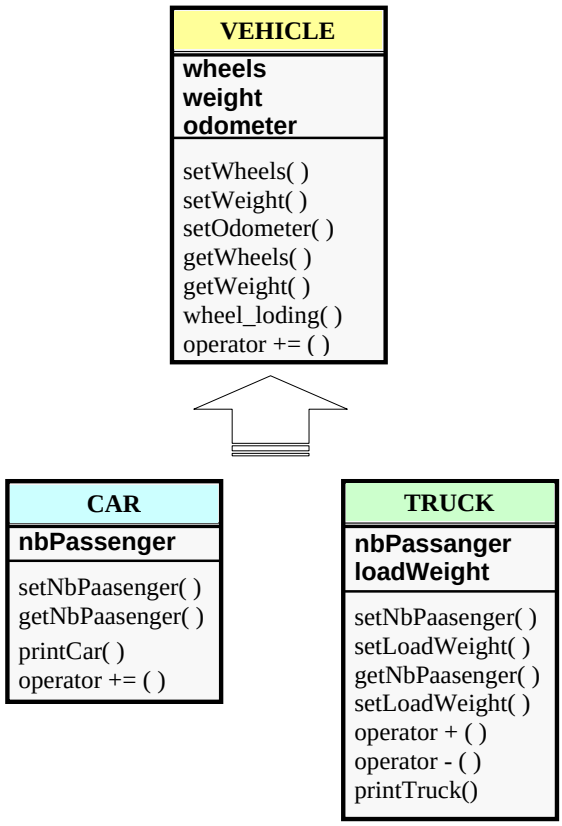
```
   motorcycle = fiat;
   cout << endl << motorcycle << endl;
   fiat.printCar();

   volvo = motorcycle;
   cout << endl << motorcycle << endl;

   trkPtr =  static_cast<Truck*>(&motorcycle);
   cout << trkPtr->getWheels() <<endl;
   cout << trkPtr->getNbPassenger() <<endl;
}
```

Syntax Error…

Unknown results…



```
2

This vehicle has: 2 wheels
It weight: 98.7 Kg
And ran for: 4544.8 Km

4 Passengers car
This vehicle has: 4 wheels
It weight: 1800 Kg
And ran for: 76213.2 Km
Your last odometer reading is: 76213.2

This vehicle has: 4 wheels
It weight: 1800 Kg
And ran for: 76263.2 Km
This truck load is: 5000
It weigts: 2000
Efficiency value is: 0.714286

This truck load is: 5434
It weigts: 2000
Efficiency value is: 0.730966

This vehicle has: 4 wheels
It weight: 1800 Kg
And ran for: 76263.2 Km

4 Passengers car
This vehicle has: 4 wheels
It weight: 1800 Kg
And ran for: 76263.2 Km

This vehicle has: 4 wheels
It weight: 1800 Kg
And ran for: 76263.2 Km

1245032
4
```

Unknown results…