CST8219 – C++ Programming
Lab 6

Introduction:
The goal of this lab is to practice inheritance, and virtual functions.

Reference:
Week 6 Powerpoint materials on Brightspace. There are many reference websites at the end of the powerpoint slides.

Steps:

1. Declare another class called ElectricVehicle, which has a variable called currentCharge, and maximumCharge. Implement another class called GasolineVehicle, which has a variable called currentGasoline, and maximumGasoline. They should both inherit from the Vehicle class. Since both electric and gasoline engines have an efficiency factor, give the Vehicle class a variable called engineEfficiency as a float.

2. Next, implement a class called HybridVehicle which inherits both from GasolineVehicle, and ElectricVehicle.

3. Make your Vehicle class as an abstract class by writing these pure virtual functions:

a) float calculateRange(); // This should return the current range of the vehicle. This is the currentGasoline or currentCharge * 100 divided by the efficiency. This number should be the number of km that the vehicle can drive.

b) float percentEnergyRemaining(); // This should return the currentEnergy / maximumEnergy * 100.0f;
For the hybrid car, output the energy remaining as the average of the two percentages.

c) void drive(float km); //This should reduce the current level of energy by:
currentEnergy -= (km/100) * efficiency;
//For the HybridVehicle, it should use up the electric energy first, and then use the gasoline engine after. If the current energy is less than 0, then output a message ***"Your car is out of energy!"***

4. The constructors for the GasolineVehicle and ElectricVehicle class should take a parameter for the maximum amount of energy that it can hold, and a float for the efficiency rating. For instance, a gasoline engine normally has an efficiency measured in L/100km, and an electric engine is measured in kWh/100km. The constructor for the HybridVehicle should take 4 parameters to set the maximumGasoline, gasolineEfficiency, maximumCharge, and electricEffeciency. Assume that when a vehicle object is first constructed, its current energy is equal to the maximum energy (it's full).

5. Write virtual destructors for all four of your classes that just print out the message:

cout << “In (ClassName) Destructor” << endl; //Change ClassName to either Vehicle, Hybrid, Gasoline, Electric,

6. Change your main function so that it looks like this:

```
Vehicle* testVehicle(Vehicle *pVehicle, const char* vehicleName)
{
        cout << vehicleName << "'s range is: " << pVehicle->calculateRange() << endl;
        pVehicle->drive(150); //drive 150 km
        cout << vehicleName << "'s energy left is: " << pVehicle->percentEnergyRemaining() << endl;
        cout << vehicleName << "'s range is now: " << pVehicle->calculateRange() << endl;

        return pVehicle;
}

int main(int argc, char **argv)
{
                                        //50L of gas, 7.1 L/100km
        delete  testVehicle(new GasolineVehicle(50, 7.1), "Corolla");

                                        //42 L of gas, 4.3 L/100km, 8.8kWh, 22 kWh/100km
        delete testVehicle(  new HybridVehicle(42, 4.3, 8.8, 22.0), "Prius" );

                                        //75 kWh, 16 kWh/100km
        delete testVehicle( new ElectricVehicle(75, 16), "Tesla 3");

        return 0;
}
```

7. You will notice that the compiler will say that the destructor is ambiguous. That's because HybridVehicle has two parent classes called Vehicle and it doesn't know which one to call. Just cast the HybridVehicle to one of the parent classes:
   delete testVehicle(dynamic_cast<GasolineVehicle*>(new HybridVehicle…
   or: delete testVehicle(dynamic_cast<ElectricVehicle*>(new HybridVehicle…

8. Once you are finished, use git bash to create a new branch called “Week6”. Then commit your work to the branch: **git commit -am “Finished Week 6”**
Create a zip file containing everything in your week2 directory and submit it on Brightspace. Make sure it includes week2.cpp, Vehicle.h, Vehicle.cpp, GasolineVehicle.h, GasolineVehicle.cpp, ElectricVehicle.h, ElectricVehicle.cpp, HybridVehicle.h, HybridVehicle.cpp CMakeLists.txt, the .git folder.

Marks: (total of 14)

The output of calculateRange works for all 3 Vehicle classes	+3
Calculating the percentEnergyRemaining works for the 3 vehicle classes	+3
The drive() function reduces the current energy in all 3 classes	+3
The destructor works as virtual for all 3 classes	+3

Vehicle, GasolineVehicle, ElectricVehicle, and HybridVehicle are all defined in their own .h and .cpp files +2