Classes

1. Create a class called Motor that represents the motor, fuel tank and transmission of a car. It should contain the private attributes gearRatio (float), maximumFuelLevel (float, in liters), currentFuelLevel (float , in liters), currentSpeed (float , in km /h), fuelEfficiency (float) where fuelEfficiency (float) represents the fuel consumption of liters per 100km.
2. Create constructors: One without any arguments, one with all, more as you see fit. Which default values do you deem reasonable? Why?
3. Create the public functions getCarSpeed, getFuelLevel, getFuelQuota, refill , setEngineSpeed , getGearRatio, drive(int) where drive(int) gets a distance driven (in km) and updates the fuel level accordingly.
4. Use classifiers (such as const , volatile , mutable) as you see fit. Justify your decisions.

Inheritance

1. Rename the class to ElectroMotor and create a class DieselMotor and an interface Motor. Move as many attributes and functions from ElectroMotor to Motor as you see fit. Replicate all other attributes and functions in the newly created class DieselMotor.
2. Use classifiers (such as const , volatile , mutable, virtual) as you see fit. Justify your decisions.
3. How is the interface Motor distinguished from the classes ElectroMotor and DieselMotor?

Polymorphism

1. Modify the interface Motor so that every class extending it is forced to implement an own version of the function getGearRatio without having to implement the functions getCarSpeed, drive. Which classifier is necessary for this?
2. Create a class Gear that represents one of the gears a transmission of a diesel-powered car can be in. It should only contain the private attribute gearRatio (float) and public function getGearRatio(). Modify the class DieselMotor to accept a positive number of gears with different gear ratios and store them (sorted) in a suitable attribute in a container provided by the standard library.
3. Create the functions increaseGear (), decreaseGear(), setGear(unsigned int) for the class DieselMotor to switch between its gears. Modify the function getGearRatio accordingly.