Problem 1. Car Instrument Simulator
Design a set of classes that work together to simulate a car's fuel gauge and odometer. The FuelGauge class will store the car's current amount of fuel in gallons, report the car's current amount of fuel, indicate when the fuel tank is empty, and refuel the car's fuel tank to a maximum of 15 gallons. The Odometer class will be a friend of the FuelGauge class, so it will have access to private members of the FuelGauge class.

The Odometer class will store the car's current mileage and report the car's current mileage. As well, it will increment the mileage by 1 mile. When the maximum mileage of 999,999 miles is exceeded, it will reset the current mileage to 0. For every 24 miles, the Odometer class will decrease the FuelGauge object by 1 gallon to simulate the car's fuel economy as 24 miles per gallon.

Demonstrate the classes by creating instances of each. Simulate filling the car up with fuel, and then run a loop that increments the odometer until the car runs out of fuel. During each loop iteration, print the car's current mileage and amount of fuel.

The Source Code is in Three Separate Files

Source Code:

File 1 (Named FuelGauge.h):

```
#pragma once
//FuelGauge.h
#ifndef FUELGAUGE_H
#define FUELGAUGE_H
#include <iostream>
class FuelGauge {
private:
	int gallons; //gallons in the tank
public:
	friend class Odometer;

	FuelGauge(int gals) //constructor initializes gallons member
	{
		gallons = gals;
	}

	int getGallons() //getter method for gallons member
	{ return gallons; }

	void report() //reports the number of gallons and if the tank is empty
	{
		if (gallons == 0)
			std::cout << "Fuel: " << gallons << "\nEMPTY!\n";
		else
			std::cout << "Fuel: " << gallons << std::endl;
	}

	void addToTank(int gal) //adds gallons to tank up to 15 gallons
	{
```

```
            if (gallons <= 14 && gallons + gal <= 15)
            {
                  std::cout << "Adding fuel, going from " << gallons << " to " <<
gallons + 1 << std::endl;
                  gallons += 1;
            }
            else if (gallons + gal > 15)
            {
                  std::cout << "Adding fuel, going from " << gallons << " to
15\n";
                  gallons = 15;
            }
            else
            {
                  std::cout << "Tank is full, cannot add any more!\n";
            }
      }
};
#endif // FUELGAUGE_H
```

File 2 (Named Odometer.h):

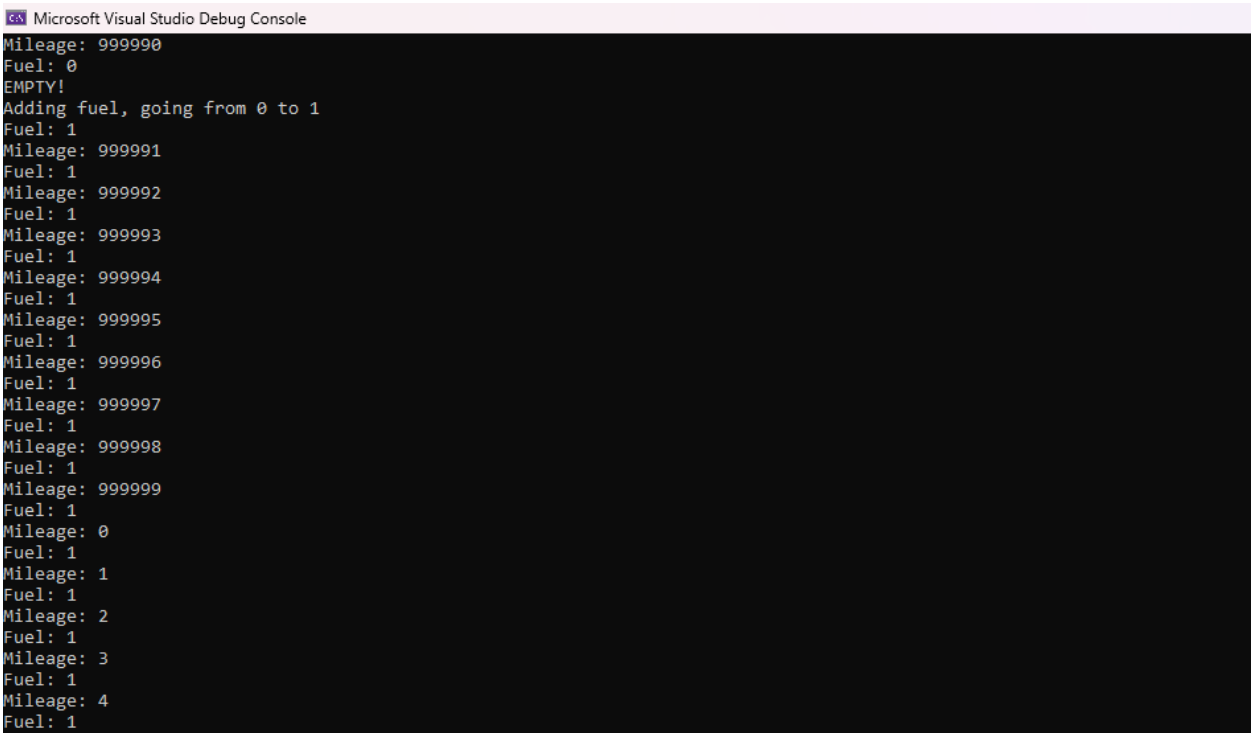
```
#pragma once
//Odometer.h
#ifndef ODOMETER_H
#define ODOMETER_H
#include <iostream>
#include "FuelGauge.h"
class Odometer {
private:
      int mileage; //total mileage on vehicle
      int addedMiles; //miles drives since the object was declared
public:
      Odometer(int miles) //constructor initializes mileage and addedMiles
      {
            mileage = miles;
            addedMiles = 0;
      }

      void report() //reports number of miles
      {
            std::cout << "Mileage: " << mileage << std::endl;
      }

      void advance(int miles, FuelGauge& obj) //adds miles to the vehicles, and
regulates fuel economy
      {
            if (mileage != 999999)
                  mileage++;
            else
                  mileage = 0;
            addedMiles++;
            if (addedMiles == 24) //if the vehicle has driven 24 miles, it loses
one gallon
            {
                  obj.gallons -= 1;
                  addedMiles = 0;
```

```
            }
        }
};
#endif // ODOMETER_H
```

File 3 (Named main.cpp):

```
//DO NOT MODIFY THIS SECTION
#include <iostream>
#include "FuelGauge.h"
#include "Odometer.h"
using namespace std;
int main()
{
        //ifstream ifile;
        Odometer odom(999990); //Create and initialize
        FuelGauge gauge(0); //Create and initialize to zero
        odom.report(); //Display info from odom
        gauge.report(); //Display info from gauge
        gauge.addToTank(1); //Add one gallon
        gauge.report(); //Display new info from gauge
        while (gauge.getGallons() > 0) //Repeat while we still have fuel
        {
                odom.advance(1, gauge); //Advance 1 mile. Include reference to gauge
                odom.report(); //Report info from odom
                gauge.report(); //Report info from gauge
        }
        return 0;
}
//ADD YOUR CODE FROM HERE
```

Output:



```
Mileage: 5
Fuel: 1
Mileage: 6
Fuel: 1
Mileage: 7
Fuel: 1
Mileage: 8
Fuel: 1
Mileage: 9
Fuel: 1
Mileage: 10
Fuel: 1
Mileage: 11
Fuel: 1
Mileage: 12
Fuel: 1
Mileage: 13
Fuel: 1
Mileage: 14
Fuel: 0
EMPTY!

C:\Users\Kolbe Williams\OneDrive\Documents\College Classes\C++\Lab10\CPP_Lab10\x64\Debug\CPP_Lab10.exe (process 22128) exited with code 0.
Press any key to close this window . . .
```