**Homework-6**

**Out Date:** 10/30/2017 (Tuesday)

**Due Date:** 11/04/2017 (Sunday) 11:59PM

**Problem Statement:** Design and implement a set of classes that work together to simulate a car’s fuel gauge and odometer. The classes you will design are:

The **FuelGauge** Class: This class will simulate a fuel gauge. Its responsibilities are

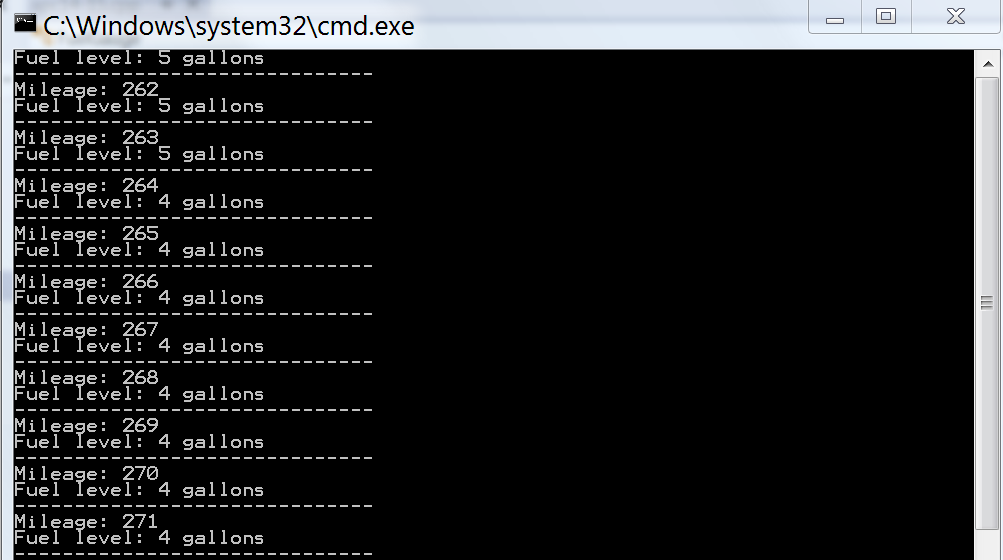
* To know the car’s current amount of fuel, in gallons **[5 points]**.
* To report the car’s current amount of fuel, in gallons **[5 points]**.
* To be able to increment the amount of fuel by 1 gallon. This simulates putting fuel in the car **[10 points]**. (The car can hold a maximum of 15 gallons.)
* To be able to decrement the amount of fuel by 1 gallon, if the amount of fuel is greater than 0 gallons. This simulates burning fuel as the car runs **[10 points]**.

The **Odometer** Class: This class will simulate the car’s odometer. Its responsibilities are:

* To know the car’s current mileage **[5 points]**.
* To report the car’s current mileage **[5 points]**.
* To be able to increment the current mileage by 1 mile. The maximum mileage the odometer can store is 999,999 miles. When this amount is exceeded, the odometer resets the current mileage to 0 **[10 points]**.
* To be able to work with a **FuelGauge** object. It should decrease the **FuelGauge** object’s current amount of fuel by 1 gallon for every 24 miles traveled **[10 points]**. (The car’s fuel economy is 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 **[10 points]**. During each loop iteration, print the car’s current mileage and amount of fuel **[10 points]**.

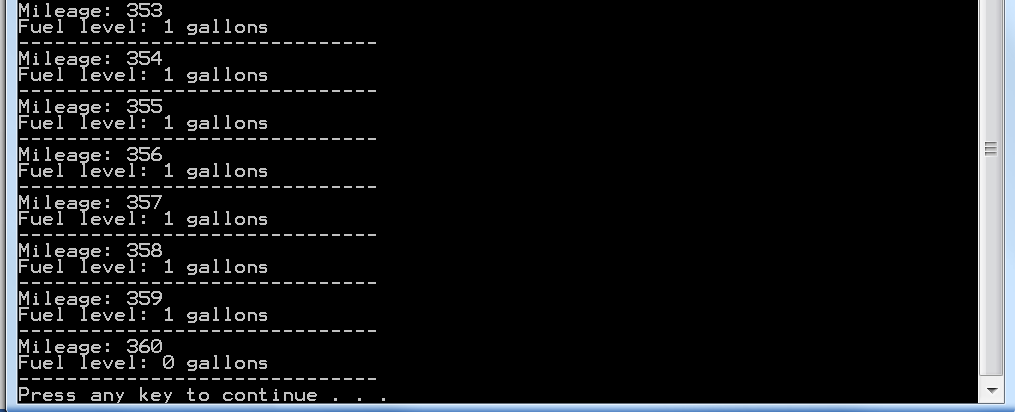
**Sample output:**



.

.

.



**Scoring Distribution [100 points]**

* 80 points for implementing the above mentioned requirements.
* 10 points for appropriate comments
* 10 points for programing style

**Blackboard Submission**

1. Submit the file
   1. FuelGauge.h, FuelGauge.cpp
   2. Odometer.h, Odometer.cpp
   3. Prog.cpp
2. Zip the files
3. Upload the zip file to Blackboard