Erick Narvaez

Nasir Ratnani

Hameed Hakimi

Chris Rodriguez

Program 4

**Problem Statement**

The Car Instrument Simulator project involves designing a set of classes that can represent the fuel gauge and odometer of a car that holds a maximum of 15 gallons and has a fuel economy of 24 mpg. The FuelGauge class should be able to know the car’s current fuel amount, report it, increment it, and decrement it. The Odometer class should be able to know the car’s current mileage, report it, increment it, decrement it, and decrease a FuelGauge object’s fuel for every 24 miles driven.

We shall demonstrate the classes by creating a main program that simulates filling up a car and driving until the car runs out of fuel.

**Approach to Solution**

This program was written in C++ using CLion IDE and vim on Ubuntu 15.10. It contains two classes, FuelGauge and Odometer. The FuelGauge class has an int field - fuel, and a const static int field - maxCapacity. The incrementFuel and decrementFuel methods will add/subtract 1 to fuel, respectively, and increment will add until the tank is at max capacity, while decrement will subtract until the tank is empty.

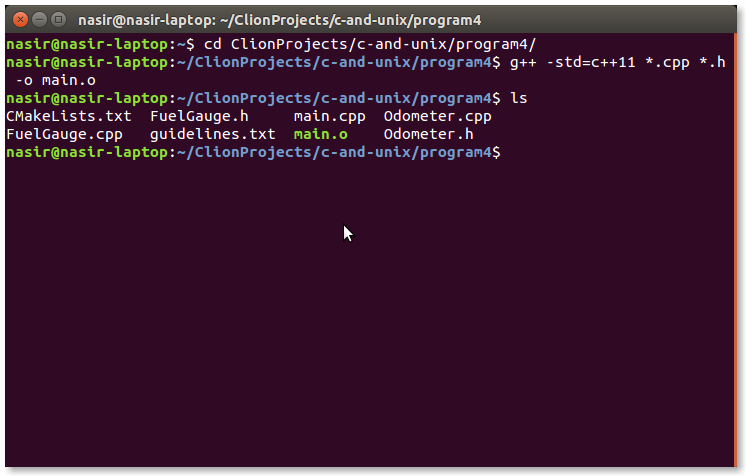
The Odometer class has an int mileage field, and two const static int fields, maxMileage and mpg. It has two methods, incrementMileage which accepts a FuelGauge object as a parameter and passes by reference to allow the FuelGauge to change. When the mileage of the Odometer exceeds 999,999 it resets to 0, and when the mileage modulo mpg is equal to 0, the fuel is decrement by one. Odometer also has an accessor for mileage called getMileage.

The simulation is main.cpp creates both a FuelGauge and Odometer object, fills up the FuelGauge to maximum capacity, and passes it to the Odometer method incrementMileage. It will continue to increment mileage and decrease fuel accordingly until the FuelGauge is empty.

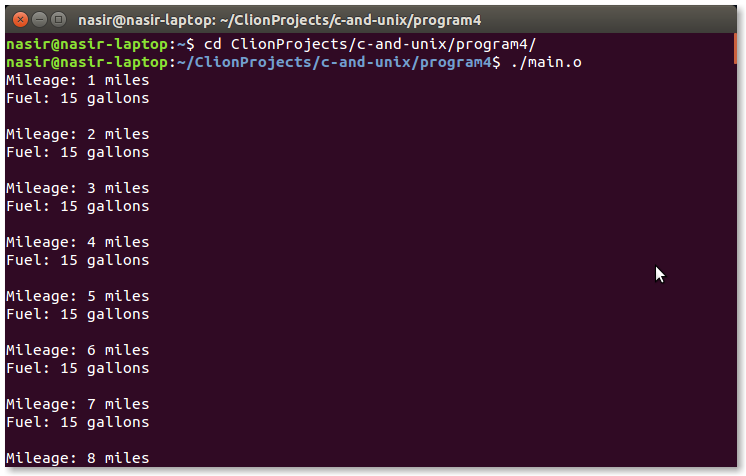
**Solution Description**

To run the solution, compile inside the program4 folder using the following command:

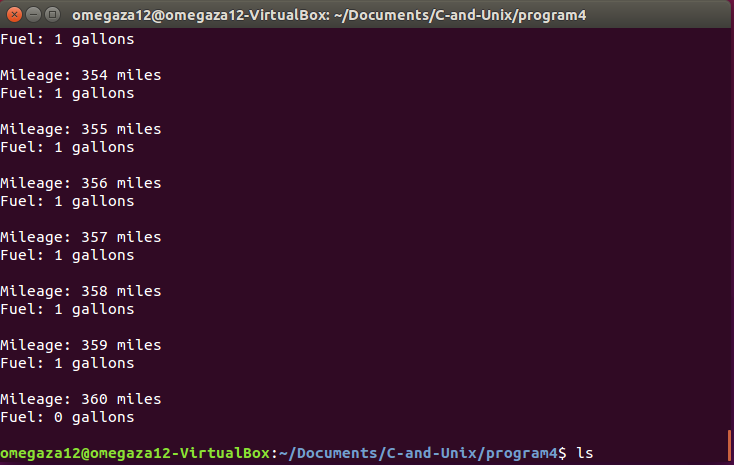
g++ -std=c++11 \*.cpp \*.h –o main.o



The following is a screenshot of the command and running the program. The first screenshot if of the programs beginning, and the second is of the end.



First Screenshot



Second Screenshot

**Course Learning Objectives (1-4):**

1. Ability to create classes of abstract data consisting of variables and functions

2. Ability to utilize C++ constructors, copy constructors, and destructors

3. Ability to utilize C++ OOP features using static member data and member functions

4. Ability to utilize C++ File and Stream Input/Output Processes

This project utilized classes, constructors, static member data, and output processes, which demonstrates the first 4 Learning Outcomes. We created two classes (1st CLO), made new constructors for the classes (2nd CLO), made the fuel efficiency of the car a static member variable of the Odometer class (3rd CLO), and printed the information of the car (4th CLO).