Driving in Formation Vehicles Redux

CSE 1325 – Fall 2018 – Homework #5
Due Tuesday, September 25 at 8:00 am

Assignment Overview

In our final homework before the first exam, we'll modify our (or the suggested) solution from Homework #4 to print the data we calculated into a nice, neat table!

Full Credit: **Using git** (with *at least* 3 commits) and a Makefile:

- **Baseline** either the full credit, bonus, or extreme bonus code, your choice, from either **your code from Homework #4** or, if you prefer, the suggested solution provided by the professor **as your full_credit starting point for Homework #5.**
- **Using the C++ <iomanip> library, modify main.cpp to print a *table* of cost data** for the gas and electric (and, if you baselined the extreme bonus, plugin-hybrid) vehicles. **The decimal points must align. Include column headers** as the cost per kWh and cost per gallon for electricity and gas, respectively.

Deliver file CSE1325_05.zip to Blackboard. In **subdirectory "full_credit"**, include your full git repository, including your modifications to **main.cpp** , the (optionally) modified .h and .cpp files for classes **Vehicle**, **Gas_vehicle**, and **Electric_vehicle, Makefile,** and any optional screenshots you believe will assist the graders.

Bonus: **Using git** (3+ additional commits) and a copy of the above code and the <sstream> library, **modify main to include a $ preceding each cost (header and body)** in the table**. The decimal points must still align, and the $ must *immediately* precede the first digit of each cost.**

In CSE1325_04.zip **subdirectory "bonus"**, include your git repository with the updated version of **main.**

Extreme Bonus: No Extreme Bonus is available this week.

Notes

Use Git

You are REQUIRED to use git locally (not GitHub, unless you so choose[1]) while developing this code. You may elect to include a screenshot of your git commit log in case of problems, though you will no longer lose points if you don't. **Check the Notes in Homework #2 or Git in 5 Pages if you can't remember how.**

Use a Makefile

You are REQUIRED to use a simple Makefile that can build and execute your main program given the simple command "`make`" (or Tools → External Tools → Build in Gedit), and also support "make test" to build and run all of your regression tests. We covered how to create this step-by-step in Lecture 05, so these will no longer be routinely supplied to you.

Streaming Out a Table

This homework should be fairly self-explanatory. Use the I/O manipulators discussed in Lecture 08 to align the table columns using 2 digits of fixed precision, with the year, make, model, and body type printed in the right-most column – one row per vehicle. Notice the two table headers, for cost/kWh and cost/gallon. depending on whether a vehicle is electric or gas. (If you baseline from Homework #4's extreme bonus, you may display only 5 price values for the plug-in hybrid – or get creative!)

Your output should look something like this:

```
ricegf@pluto:~/dev/cpp/201808/P5/full_credit$ ./main
      0.05      0.08      0.11      0.13      0.15  Cost per kWh
      2.00      2.25      2.50      3.00      4.00  Cost per gallon
  ========  ========  ========  ========  ========  ==============================
      1.60      2.56      3.53      4.17      4.81  2014 Telsa Model S 85 Sedan
      1.21      1.94      2.66      3.15      3.63  2014 Telsa Model 3 LR Sedan
      1.40      2.23      3.07      3.63      4.19  2018 GM Bolt Hatchback
      1.29      2.06      2.84      3.35      3.87  2018 Nissan LEAF SL Hatchback
      7.69      8.65      9.62     11.54     15.38  2017 Toyota RAV4 Crossover
      9.52     10.71     11.90     14.29     19.05  2018 Ford F-150 Truck
      6.90      7.76      8.62     10.34     13.79  2018 Nissan Rogue Hatchback
      9.09     10.23     11.36     13.64     18.18  2018 Chrysler Pacifica Minivan
```

Deliverables

In the full_credit directory, you will provide:

- At least 7 C++ source files named **vehicle.h and .cpp, gas_vehicle.h and .cpp, electric_vehicle.h and .cpp,** and **main.cpp,** and their associated git repository. How clever you would be to also have regression tests for your classes!
- A *required* make file named **Makefile** that builds your application by default ("make").
- *Optional* screenshots that illustrate your application or git repository history.

1 If you decide to use Github, please include a text file named GitHub.txt with a link to your public GitHub repository in CSE1325_02.zip.

Bonus

Continue to use git and a copy of the above code and the <sstream> library, **modify main to include a $ preceding each cost (header and body)** in the table**. The decimal points must still align, and the $ must *immediately* precede the first digit of each cost.**

You may NOT use std::put_money or another library to format the currency amounts – the point of this bonus is for you to **practice using string streams**. Simply create a new string stream for each cell in the table, and stream a $ and a formatted dollar amount to it. Then stream the string value of that string stream into the width allocated for the cell.

Your output should look something like this:

```
ricegf@pluto:~/dev/cpp/201808/P5/bonus$ ./main
     $0.05     $0.08     $0.11     $0.13     $0.15  Cost per kWh
     $2.00     $2.25     $2.50     $3.00     $4.00  Cost per gallon
 ========  ========  ========  ========  ========  ==============================
     $1.60     $2.56     $3.53     $4.17     $4.81  2014 Telsa Model S 85 Sedan
     $1.21     $1.94     $2.66     $3.15     $3.63  2014 Telsa Model 3 LR Sedan
     $1.40     $2.23     $3.07     $3.63     $4.19  2018 GM Bolt Hatchback
     $1.29     $2.06     $2.84     $3.35     $3.87  2018 Nissan LEAF SL Hatchback
     $7.69     $8.65     $9.62    $11.54    $15.38  2017 Toyota RAV4 Crossover
     $9.52    $10.71    $11.90    $14.29    $19.05  2018 Ford F-150 Truck
     $6.90     $7.76     $8.62    $10.34    $13.79  2018 Nissan Rogue Hatchback
     $9.09    $10.23    $11.36    $13.64    $18.18  2018 Chrysler Pacifica Minivan
```

Deliverables

In the bonus directory, you will provide the same files as the full_credit directory.