

Programming Project #1

Assignment Overview

This project focuses on some mathematical manipulation. It is worth 10 points (1% of your overall grade). It is due Monday, January 23rd before midnight.

The Problem

There is a lot of discussion these days of our "carbon footprint". The term footprint is another way to indicate how much carbon dioxide and other greenhouse gasses an individual (or some small group) emits. Controlling this emission is one of the keys to controlling global warming, so it is a good idea to provide a simple calculator for your carbon footprint.

Some Background

The EPA provides a nice web site where you can make such calculations (<https://www3.epa.gov/carbon-footprint-calculator/>) as well as an Excel spreadsheet to explain the values. We are just going to look at the calculations for automobile gasses. Here are the facts:

- The average car emits 19.6 lbs_CO2/gallon_gas
- That emission, when accounting for other greenhouse gasses (carbon monoxide, nitrous oxide) increases the amount by a factor 100/98.65 . That is, the amount of CO2 times this ratio is the total amount of greenhouse gas emitted

Program Specifications

Your program will do the following:

1. Take as input two floating point values (in this order, as indicated):
 - a. the mpg, the number of miles per gallon for your car
 - b. the miles driven per year
2. Print the following three results on a single line, each value separated from the other by a single space. The precision is to two decimal point of accuracy (see note 1 below)
 - a. the lbs of greenhouse gas emitted in one year
 - b. the change in lbs of greenhouse gas emitted if the mileage went up by 1 mpg
 - c. the change in lbs of greenhouse gas emitted if the mileage went up by 5 mpg

Deliverables

proj01.cpp -- your source code solution (remember to include your section, the date, project number and comments).

1. Please be sure to use the specified file name, i.e. "proj01.cpp"
2. Save a copy of your file in your CSE account disk space (H drive on CSE computers).
3. You will electronically submit a copy of the file using the "handin" program:
<http://www.cse.msu.edu/handin/webclient>

Assignment Notes:

1. We might as well try to make the output somewhat readable. You can look this up in the text or on the internet, but you can use the following modifiers that affect how numbers print.
 - a. `cout << fixed` Elements will be printed as floating point numbers (ex 123.456)

- b. `cout << scientific` Elements will be printed in scientific notation (ex 1.23456×10^2)
 - c. `cout << setprecision(2)` Floating point numbers will have 2 values after the decimal point and will be rounded, (123.46)
 - d. Thus `cout << fixed << setprecision(2) << 123.4567 << endl;` will print 123.46
2. The following statement will read two variables off of the same, space separated line. It is an example of chaining input:

```
...
double d1, d2;
cin >> d1 >> d2;
```

- ...
3. There are three pairs of input and output files provided for you to check yourself. The pairs are inX.txt and outX.txt, where X is some integer number, and the number indicates the pair. Thus if you compile your code, using the default executable name a.out, then `./a.out < int1.txt` should produce the output found in out1.txt. Look at lab00 for examples of redirected input.
4. You do not have to check for bad input values. In general, we will explicitly indicate the errors we are looking for, but for now we are not checking for input errors.
5. If you read off of a single line for the input, you can do a handy trick off the command line. You can redirect a file (with that single line of input) to your main program. Thus in the terminal, in the directory where your program proj01 resides, you can do:

```
./proj01 < fileOfInput.txt
```

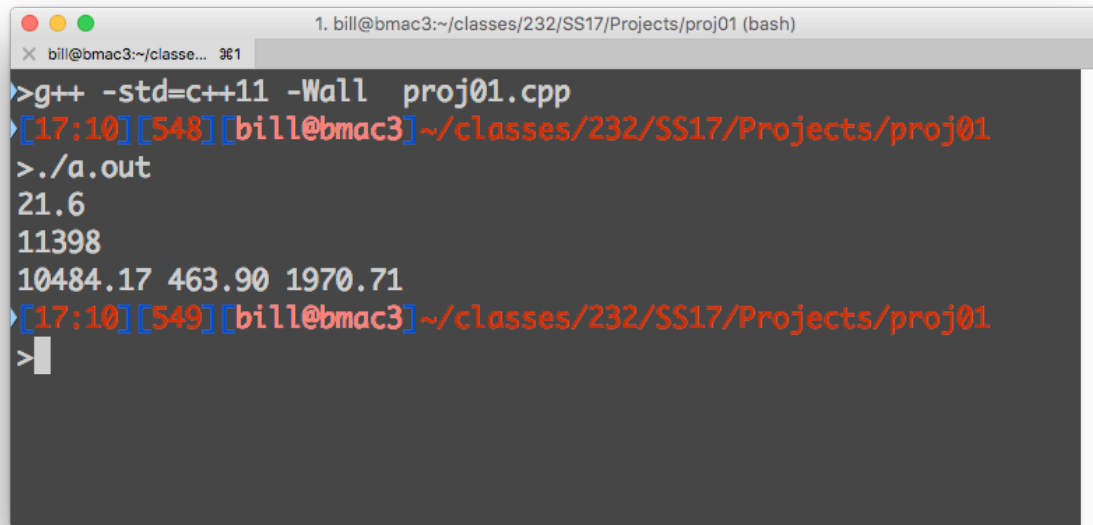
This will automatically feed the input to the `cin` statement and produce the output. Makes it easier to test things (and to grade them too!!).

6. You can check your calculations on <http://www.wolframalpha.com/> or using python or some other approach that you are comfortable with.

Getting Started

1. Create a new directory for your project
2. If you are in a CSE lab or on x2go, select the H: drive as the location to store your file.
3. Save the name of the project: proj01.cpp
4. Prompt for some of the values and print them back out, just to check yourself.
5. When you do a calculation, use the provided table as a way to check yourself.
6. Use the handin web site to hand in the program (incomplete as this point, but you should continually hand things in).
7. Now you can calculate the individual elements of the project. After writing code for each of the below aspects, compile and print out values to make sure you are getting the correct values. Do it **INCREMENTALLY**, that is write and check various calculations
8. Now you enter a cycle of edit-run to incrementally develop your program.
9. Use handin to hand in your final version.
10. **Remember**: the project must compile on the lab machines or the x2go environment for project credit. Check it before you hand it in!

Sample Interactions



```
1. bill@bmac3:~/classes/232/SS17/Projects/proj01 (bash)
X bill@bmac3:~/classes/232/SS17/Projects/proj01 961
>g++ -std=c++11 -Wall proj01.cpp
[17:10][548][bill@bmac3]~/classes/232/SS17/Projects/proj01
>./a.out
21.6
11398
10484.17 463.90 1970.71
[17:10][549][bill@bmac3]~/classes/232/SS17/Projects/proj01
>
```