

Simple Algebraic Calculations

One of the first things you will learn about C++ is how to perform numerical computations. In this project, you are given an incomplete program (see the end of this specification), which already includes the statements necessary to read input and specify output. You must add statements so that your completed program will perform the specified calculations.

The input file:

The input file for this program is named "TravelData.txt". A sample input file is given below. The input file begins with four header lines which are meaningful to a human reader but must be ignored by the program. Lines of data, each pertaining to a particular trip that has been taken, follow the header section. Each of these lines provides the total miles covered during the trip, the total number of gallons of gasoline used during the trip, and the cost per gallon of that gasoline.

```
; CS 1044 Fall 2003
; Project 2 Test File 0
;
; Miles      Gallons      $$/Gallon
372.4        17.3         1.429
181.0         5.7          1.449
 92.8         2.1          1.389
274.1        15.3         1.479
200.3         7.3          1.339
```

There may be data for an arbitrary number of trips; i.e., we cannot make any assumptions about the minimum or maximum number of trips for which we will be given data.

As your class covers input and output in C++, you should examine the given program and make sure you understand precisely how it manages its input and output.

What must be calculated:

For each trip, the program must calculate the average MPG (miles traveled per gallon of gasoline used), and the average \$PM (cost of gasoline per mile traveled) during that trip.

Also, the program must calculate summary data for the complete set of trips. Specifically, the program must calculate the total miles traveled, the total gallons of gasoline used, and the average MPG and \$PM over all of the given trips.

Note: the average MPG over all the trips is NOT just the sum of the individual MPGs divided by the number of trips. Think about it...

In order to produce the most accurate results possible in C++, all the decimal values are stored using variables of type `double`, not type `float`.

Although the given program is incomplete, it does declare all the variables that are needed to calculate the required results. You may declare additional variables if you like, but they will not be logically necessary.

The output file:

The output file is named "CostData.txt". An output file, which corresponds to the given input file, is shown below. The first two lines specify the programmer and the assignment, and the third is blank. The fourth line specifies column headers for the table the program will write, and the fifth line separates the table data from the labels.

The table body contains one line of output for each given trip, displaying the mileage, gallons of gasoline used, MPG and \$PM for that trip. Each trip is also assigned a numerical label, as shown.

When a decimal number is written, the number of digits displayed after the decimal point is called the precision of the displayed value. The mileage and gallon values must be written with precision 1, the MPG values with precision 2, and the \$PM values with precision 3 (as shown in the output sample).

```
Programmer: <your name here>
CS 1044 Project 2 Fall 2003
```

Trip	Miles	Gallons	MPG	\$PM
1	372.4	17.3	21.53	0.066
2	181.0	5.7	31.75	0.046
3	92.8	2.1	44.19	0.031
4	274.1	15.3	17.92	0.083
5	200.3	7.3	27.44	0.049

	1120.6	47.7	23.49	0.061

If you have read the description of how the Curator scores your program in the *Student Guide*, you know that is important that you use the same spelling and capitalization for all the labels shown above. The horizontal spacing does not effect scoring unless you combine things that should be separate or separate things that should be combined. In any case, the given code will exactly match the labels and spacing shown above.

Submitting your program:

You will submit this assignment to the Curator System (read the *Student Guide*), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guide*.

You will be allowed up to five submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest score you achieve will be counted.

The *Student Guide* and submission link can be found at:

<http://www.cs.vt.edu/curator/>

Evaluation:

Your submitted program will be assigned a score based upon the runtime testing performed by the Curator System. We will not be evaluating your submission of this program for documentation style. However, you should examine the given program as a guide to acceptable documentation, and include similar comments for the statements you add to it.

For a number of the later projects, we will also evaluate your submission for documentation, and for other requirements. It is best to begin preparing for that now.

Pledge:

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the header comment for your program:

```
// On my honor:
//
// - I have not discussed the C++ language code in my program with
//   anyone other than my instructor or the teaching assistants
//   assigned to this course.
//
// - I have not used C++ language code obtained from another student,
//   or any other unauthorized source, either modified or unmodified.
//
// - If any C++ language code or documentation used in my program
//   was obtained from another source, such as a text book or course
//   notes, that has been clearly noted with a proper citation in
//   the comments of my program.
//
// - I have not designed this program in such a way as to defeat or
//   interfere with the normal operation of the Curator System.
//
// <Student Name>
```

Failure to include this pledge in a submission is a violation of the Honor Code.

The Program:

```
// CS 1044 Project 2   Fall 2003
//
// Programmer:      <your name here>
// Last modified:   <your information here>
//
// Purpose:
// This program performs simple computations relating to the cost
// of auto/truck trips. Given the total mileage, total gallons of
// used, and the cost per gallon of gas, the program calculates the
// average miles per gallon and the average gas cost per mile for
// that trip.
//
// In addition, the program calculates the total mileage, total gas
// consumption, average miles per gallon, and average gas cost per
// mile over all trips listed.
//
#include <iostream>    // for cout
#include <fstream>     // for file streams for input/output
#include <iomanip>     // for formatting manipulators
#include <cstdlib>    // for EXIT_SUCCESS
#include <climits>    // for INT_MAX
using namespace std;

int main() {

    ifstream In("TravelData.txt"); // Open the input file.

    ofstream Out("CostData.txt"); // Open the output file.

    Out << fixed << showpoint; // Enable floating-point output formatting.

    double Miles,           // mileage of trip
```

```

        Gallons,          // gallons of gas used on trip
        GasCostPerGallon; // cost of a gallon of gas on trip

double MilesPerGallon,   // miles per gallon on trip
        GasCostPerMile;  // cost (of gas) per mile on trip

double GasCost;          // cost of gas used on trip

double TotalMiles,      // total mileage of all trips
        TotalGallons,   // total gas used on all trips
        TotalCost;     // total cost of all trips

double OverallMPG,      // average MPG over all trips
        OverallCostPerMile; // average cost per mile over all trips

int    numTrips;        // number of trips taken

// Initialize the running totals and counters appropriately:
TotalMiles  = 0.0;
TotalCost   = 0.0;
TotalGallons = 0.0;
numTrips    = 0;

// Other initialiations:
MilesPerGallon = 0.0;
GasCostPerMile = 0.0;

// If the input file does not exist, this will detect that.
// We handle that by printing an error message and stopping the program.
if (In.fail()) {
    cout << "Input file not found: " << "TravelData.txt" << endl;
    cout << "Exiting..." << endl;
    return EXIT_SUCCESS;
}

// Write the header information to the output file:
Out << "Programmer: <your name here>" << endl;
Out << "CS 1044 Project 2 Fall 2003" << endl;
Out << endl;
Out << "Trip Miles Gallons MPG $PM" << endl;
Out << "-----" << endl;

In.ignore(INT_MAX, '\n'); // ignore the four header lines
In.ignore(INT_MAX, '\n');
In.ignore(INT_MAX, '\n');
In.ignore(INT_MAX, '\n');

// Try to read a line of data for the first trip:
In >> Miles >> Gallons >> GasCostPerGallon;

while (In) {

    // Some calculations may go here...

    // Write the specified trip data to the output file:
    Out << setw( 4) << numTrips
        << setw( 7) << setprecision(1) << Miles
        << setw(10) << setprecision(1) << Gallons
        << setw( 8) << setprecision(2) << MilesPerGallon
        << setw( 9) << setprecision(3) << GasCostPerMile << endl;

    // Try to read a line of data for another trip:

```

```
In >> Miles >> Gallons >> GasCostPerGallon;
}

// Calculate the overall averages:
if (TotalGallons > 0) {
    // Some calculations may go here...
}
else {
    OverallMPG = 0.0;
}

if (TotalMiles > 0) {
    // Some calculations may go here...
}
else {
    OverallCostPerMile = 0.0;
}

// Mark the end of the output table:
Out << "-----" << endl;
// Write the summary information to the output file:
Out << setw(11) << setprecision(1) << TotalMiles
    << setw(10) << setprecision(1) << TotalGallons
    << setw( 8) << setprecision(2) << OverallMPG
    << setw( 9) << setprecision(3) << OverallCostPerMile
    << endl;

// Close the input and output files:
In.close();
Out.close();
return(EXIT_SUCCESS);
}
```