```
1  //Ben Scherer
2  // 6/22/2017
3  // Fuel Costs Calculator
4  //  Based on user input, calculates average gas mileage ,average gas prices and car efficiency class
5
6  //Headers to include
7  #include <iostream> //cout
8  #include <iomanip> // used to manipulate cout
9  #include <string> //needed for string variable
10 #include <math.h> //used for basic arithmatic
11 #include <limits> //user for numeric_limits
12
13 using namespace std;
14
15 //Functions
16 double getDoubleInput(string questionToAsk); //Gets and validates input for all double variables
17 char getSentinel(); //Gets and validates sentinel for breaking loop.  y/n in this case
18
19 int main() {
20     //Input Variables
21     double odometerStart, odometerEnd; //Used to calculate miles traveled between fillups
22     double pricePerGallon; //Price paid at fillup
23     double numGallons; //Number of gallons purchased
24
25
26     //Output Variables
27     double avgMPG; //Average MPG calculated from user inputs
28     double avgPricePerGallon; //Average price per gallon based off ot the three gas station prices inputed
29     double costPerMile; //Average cost per mile;
30     string carClassification; // Car Efficiency classification, based off of avgMPG
31
32     //Misc Variables
33     int numFillUps = 0;// counter.  Keeps track of the number of times gas is purchased
34     double totalGallons = 0.0; //Total gallons of gas purchased
35     double totalMiles = 0.0; //Total miles traveled
36     double totalPricePerGallon = 0.0; // sum of price per gallon.  Used to calculate average
37     double totalCost; //Total money spent on gas
38
39     //Get user input
40     cout << "------------------------------------------------------\n";
41     cout << "\t\tCIS 1111 Fuel Cost Calculator\n";
42     cout << "------------------------------------------------------\n";
43     cout << "Please enter readings:\n ";
44     do { //Get odometer readings and gallons purchased
```

```
45          odometerStart = getDoubleInput("Initial Odometer reading(miles): ");
46          odometerEnd = getDoubleInput("Ending Odometer reading(miles): ");
47          numGallons = getDoubleInput("Gallons of gas purchased: ");
48
49          numFillUps++;
50          totalMiles += (odometerEnd - odometerStart);
51          totalGallons += numGallons;
52
53      } while (getSentinel() == 'y');
54
55      //Get gas prices
56      for (int i = 1; i <= 3; i++) {
57          pricePerGallon = getDoubleInput("Please enter price per gallon at gas
              station #" + to_string(i));
58          totalPricePerGallon += pricePerGallon;
59      }
60
61      //Calculate Averages
62      avgMPG = totalMiles / totalGallons;
63      avgPricePerGallon = totalPricePerGallon / 3;
64      totalCost = (totalGallons * avgPricePerGallon);
65      costPerMile = totalMiles / totalCost;
66
67      if (avgMPG <= 15)
68          carClassification = "Very Inefficient";
69      else if (avgMPG >= 16 && avgMPG <= 30)
70          carClassification = "OK";
71      else
72          carClassification = "Very Efficient";
73
74      //output
75      cout << "\n----------------------------------------------------------------\n"
76          << "Caclulated Fuel Costs\n"
77          << "----------------------------------------------------------------\n"
78          << setw(40) << left << "Number of trips to gas station: " << numFillUps
              << endl
79          << setw(40) << left << "Total Miles Driven: " << totalMiles << endl
80          << setw(40) << left << "Average Miles Per Gallon(MPG): " << setprecision
              (2) << fixed << avgMPG << endl
81          << setw(40) << left << "Average fuel cost per gallon: " << "$" <<
              avgPricePerGallon << endl
82          << setw(40) << left << "Total spend on fuel: " << "$" <<  setprecision(2)
               << fixed << totalCost << endl
83          << setw(40) << left << "Cost per mile: " << "$" << setprecision(2) <<
              fixed << costPerMile << endl
84          << setw(40) << left << "Car Efficiency Rating: " << carClassification <<
              endl
85          ;
86
87      //pauses program
88      cout << "Press enter key to exit program\n";
89      cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

```
 90      cin.get();
 91      return 0;
 92  }
 93
 94  double getDoubleInput(string questionToAsk) {
 95      double varToReturn;
 96      bool isValidInput = false;
 97
 98      do {
 99          cout << questionToAsk << endl;
100          if (!(cin >> varToReturn)) {
101
102              cout << "#########################################################\n";
103              cout << "\tERROR: Please enter a valid value\n";
104              cout << "#########################################################\n";
105              cin.clear();
106              cin.ignore(numeric_limits<streamsize>::max(), '\n');
107          }
108          else {
109              isValidInput = true;
110          }
111      } while (!isValidInput);
112      return varToReturn;
113  }
114
115  char getSentinel() {
116      char  varToReturn;
117      bool isValidInput = false;
118
119      do {
120          cout << "Enter 'y' to enter more readings.  Enter 'n' if you are
               finished" << endl;
121          if (!(cin >> varToReturn) || (tolower(varToReturn) != 'y' && tolower
               (varToReturn) != 'n')) {
122
123              cout << "#########################################################\n";
124              cout << "\tERROR: Enter 'y' to enter more readings.  Enter 'n' if you
                    are finished\n";
125              cout << "#########################################################\n";
126              cin.clear();
127              cin.ignore(numeric_limits<streamsize>::max(), '\n');
128          }
129          else {
130              isValidInput = true;
131          }
132      } while (!isValidInput);
133      return tolower(varToReturn);
134  }
```