**CSCI 135 Programming Exam #2**
Fundamentals of Computer Science I
Fall 2013

This part of the exam is like a mini-programming assignment. You will create a program, compile it, and debug it as necessary. This part of the exam is open book and open web. You may use code from the course web site or from your past assignments. When you are done, submit all your Java source files to the Moodle exam #2 dropbox. Please **double check you have submitted all the required files**.

You will have 100 minutes. No communication with any non-staff members is allowed. This includes all forms of real-world and electronic communication.

Grading. Your program will be graded on correctness and to a lesser degree on clarity (including comments) and efficiency. You will lose a substantial number of points if your code does not compile or crashes on typical inputs.
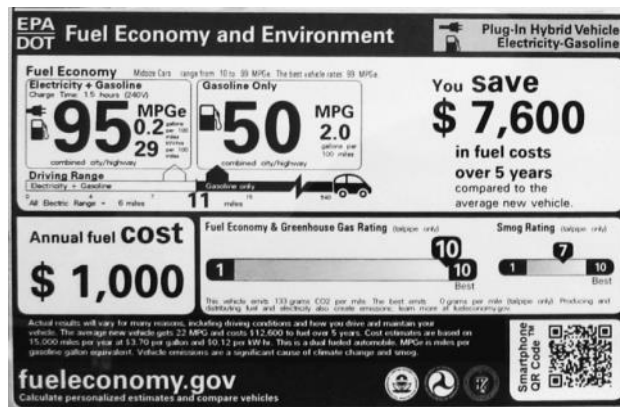
Overview. You are trying to decide what car to buy. Should you buy an oh-so-manly Hummer H3? Or perhaps a plug-in Toyota Prius hybrid is a more sensible choice. To help you decide, you want to develop a program that calculates the *total cost of ownership* of different makes and models of cars. The total cost of ownership includes not only the original price for the vehicle, but also how much you'll likely spend on gas for the lifespan of the vehicle. To do this, you will be developing two support classes:

> **Vehicle** – Represents a type of car. For example: Hummer H3, seats 5, 14.0 MPG, costs $32,300.

> **Trip** – Represents a journey you commonly make in your car. For example: a trip to the grocery store, with a roundtrip distance of 2.5 miles and that occurs 52 times per year.

Using these support classes, you will develop a program `CarCalc`. This program takes as input how long you plan to own the car, the average price of gas, and an optional make of car you are interested in. It reads a catalog of different vehicles and your yearly driving trips from a text file. From this input it prints out a total cost of ownership for each type of car.

To get started, create an empty Eclipse project and extract the contents of this zip file into your project directory: car.zip

Part 1: Vehicle. This data type represents a particular type of car. It knows things like its make (e.g. "Toyota"), model (e.g. "Prius"), the number of passengers it can hold (e.g. 5), the average miles-per-gallon (e.g. 50.0), and the cost (e.g. $30,395). You can assume the number of passengers is always an integer, miles-per-gallon is a floating-point value, and the cost is an integer.

Here is the API you are to implement for `Vehicle`:

public class Vehicle

```
        Vehicle(String make, String model, int passengers, double mpg, int price)
     int getPrice()
  double getPassengerMilesPerGallon()
  String toString()
  String getMakeModel()
 boolean isSameMake(String make)
  double costToDriveDistance(double miles, double pricePerGallon)
```

We have provided a stub version of `Vehicle.java` that includes comments describing exactly what each method should do. You will need to decide on an appropriate set of instance variables. We have also provided a test `main()` method. Here is our output:

```
% java Vehicle
Hummer H3, 5 passengers, MPG 14.0, PMPG 70.0, $32300
Toyota Prius, 5 passengers, MPG 50.0, PMPG 250.0, $30395
Toyota Avalon, 5 passengers, MPG 25.0, PMPG 125.0, $39650
Ford E350_Super_Duty_Extended, 15 passengers, MPG 11.0, PMPG 165.0, $36545

Assuming gas costs $3.79 per gallon, cost to travel 123 miles:
Hummer H3 = $33.30
Toyota Prius = $9.32
Toyota Avalon = $18.65
Ford E350_Super_Duty_Extended = $42.38
```

```
Toyota Prius, 5 passengers, MPG 50.0, PMPG 250.0, $30395
Toyota Avalon, 5 passengers, MPG 25.0, PMPG 125.0, $39650

Hummer H3, 5 passengers, MPG 14.0, PMPG 70.0, $32300
```

Part 2: Trip. This data type represents a particular car-based journey. It knows things like a friendly name for the trip (e.g. "groceries"), a round-trip distance (e.g. 27.6 miles), and the frequency you make the trip (e.g. 52 times per year). You can assume the round-trip distance is a floating-point value and the frequency is an integer value.

Here is the API you are to implement for `Trip`:

`public class` Trip

```
        Trip(String name, double milesRoundTrip, int timesPerYear)
 String toString()
 double expensePerYear(Vehicle v, double pricePerGallon)
```

We have provided a stub version of `Trip.java` that includes comments describing exactly what each method should do. You will need to decide on an appropriate set of instance variables. We have also provided a test `main()` method. Here is our output:

```
% java Trip
Hummer H3, work, round trip 10.3 miles, 250 times/year, expense per year $715.48
Toyota Prius, work, round trip 10.3 miles, 250 times/year, expense per year $200.34
```

Part 3: CarCalc. This program calculates the total cost of ownership of a set of vehicles. The program takes input both from command-line arguments as well as input from a file via standard input. The program requires two command-line arguments and possibly an optional third argument:

> ***Number of years*** – Total number of years you plan to own the vehicle. For purposes of total cost of ownership, we assume at the end of this number of years the car has no remaining value.

> ***Price per gallon*** – The average cost for a gallon of gas. We assume the price remains the same for the entire length of time you own the vehicle.

> ***Make of vehicle*** – Optional third argument. If present, only results for vehicles matching this make (e.g. "Toyota") are returned. This matching is case insensitive, i.e. specifying "Toyota" or "toyota" amounts to the same thing.

If fewer than two command-line arguments are provided, it should print a friendly help message and exit:

```
% java CarCalc
CarCalc <num years> <price per gallon> [make of vehicles]
```

If two or more arguments are given, your program will print out information regarding these arguments:

```
% java CarCalc 10 3.89
Calculating total cost of ownership for 10 years
Using price per gallon of $3.89
```

```
% java CarCalc 10 3.89 Toyota
Calculating total cost of ownership for 10 years
Using price per gallon of $3.89
Results for only Toyota vehicles
```

You can assume the first argument is always a positive integer and the second is always a positive floating-point value. All dollar amounts in your output should be rounded to **_two decimal places_**.

Your program will read in vehicles and trips from a file via standard input. StdIn.java is in the zip file containing the stub code. Here is the provided cars_trips.txt input file:

```
4
Hummer H3                       5  14.0  32300
Toyota Prius                    5  50.0  30395
Toyota Avalon                   5  25.0  39650
Ford    E350_Super_Duty_Extended 15  11.0  36545

8
work              10.3  250
grocery_shopping   2.5  52
home_depot        136.8  2
ski_discovery      90.0  8
ski_big_sky       253.0  2
hike_glacier      490.0  1
bike_homestake     21.4  4
summer_road_trip  2000.0  1
```

The file starts with an integer saying how many total vehicles are in the file. This is followed by information for each vehicle. The vehicle information is ordered as follows: make, model, number of passengers, miles-per-gallon, and cost of the vehicle. You can assume the make and model are always a single string token (e.g. "E350 Super Duty Extended" uses underscores instead of spaces in the file).

After the vehicle information, another integer specifies how many trips are in the file. The information for each trip is ordered as follows: name of trip, roundtrip distance in miles, and the number of times per year the trip is made. As with the make and model, you can assume the trip name is always a single string token.

Here are several example runs of our final program:

```
% java CarCalc 10 3.89 < cars_trips.txt
Calculating total cost of ownership for 10 years
Using price per gallon of $3.89
Hummer H3, total cost $51139.27
Toyota Prius, total cost $35670.00
Toyota Avalon, total cost $50199.99
Ford E350_Super_Duty_Extended, total cost $60522.25
```

```
% java CarCalc 10 3.89 toyota < cars_trips.txt
Calculating total cost of ownership for 10 years
Using price per gallon of $3.89
Results for only toyota vehicles
Toyota Prius, total cost $35670.00
Toyota Avalon, total cost $50199.99
```

```
% java CarCalc 10 3.89 bogus < cars_trips.txt
Calculating total cost of ownership for 10 years
Using price per gallon of $3.89
Results for only bogus vehicles

% java CarCalc 40 4.99 < cars_trips.txt
Calculating total cost of ownership for 40 years
Using price per gallon of $4.99
Hummer H3, total cost $128966.28
Toyota Prius, total cost $57461.56
Toyota Avalon, total cost $93783.12
Ford E350_Super_Duty_Extended, total cost $159574.81

% java CarCalc 1 4.99 < cars_trips.txt
Calculating total cost of ownership for 1 years
Using price per gallon of $4.99
Hummer H3, total cost $34716.66
Toyota Prius, total cost $31071.66
Toyota Avalon, total cost $41003.33
Ford E350_Super_Duty_Extended, total cost $39620.75
```