Intro to Computer Science

PA01

Writing some simple programs

Code due by Wednesday, January 29th at 11:00 AM

Paperwork due the same day at the start of class

Introduction

Your book spends a lot of time talking about problem solving. Let's look at how this would work, at least for some small problems (because you don't know enough to tackle big problems yet).

For small problems like we will be working with in this assignment, this process means that you:

1. Receive a customer request
2. Consider what this problem actually involves and how a program will solve it
3. Design the code to solve the problem
4. Write the code to solve the problem and **test** that the code actually works

In this assignment I would like you to :

1. Look at the two customer requests below (step #1 in the software development process).
2. For each request I would like you to analyze and design a solution to the problem and write a one paragraph summary of what process you need to use to solve each of the problems below). I will be collecting these paragraphs as part of a "design document" that you will hand in to me in class.
3. After you have studied the problems and prepared your problem solving solution you should write the code that solves each of the problems. Please make sure that you write the code in two separate files saved with the names given below.
4. Once you think you have the code working, **TEST it thoroughly**. Don't simply run it once with one set of values and assume it is good. Test it with

three to five different values (or sets of values actually) and make sure that the numbers the computer gives you agree with the numbers you know to be true or have calculated "by hand"

Notes and requirements:

- Make sure that you save your programs in the correctly named files
- Make sure that you are following the proper policies regarding code based homework
- Use meaningful variable name with the proper style (useCamelCase)
- Use meaningful constants and names where appropriate and use proper style (ALL_CAPS_AND_UNDERSCORES)
- Your programs should ask for input in the exact order specified in the customer requests below.

Customer Request #1 (mileage.py)

You are contacted by Dr. Otto Mobile who is analyzing the overall efficiently of a variety of American and Japanese cars. To help understand how they compare he has decided to calculate their MPG. You record the starting and ending mileage of the car, use this to calculate the total mileage driven, and then divide this by the amount of gas consumed.

For example, on a recent trip to visit my family my car's odometer was at 92567 when I left and 92783 when I returned. When I refilled the gas tank it required 10.6 gallons of gas. This comes out to 20.377 miles per gallon (my old car gets really bad mileage)

Dr. Mobile asks you to write a program that asks for the starting and ending mileage as well as the total gasoline consumed. It then calculates the MPG and prints a nice message regarding the MPG of the car.

Note: Your program should ask for the starting mileage **first**, the ending mileage **second**, and the total gasoline consumed **last**. Failure to following this ordering will result in some point deduction.

Customer Request #2 (time.py)

You are contacted by Dr. Tyme Aday who teaches time management at Whatsamata U. He wants a tool that will allow him to convert time in days, hours, minutes, and seconds into millieconds. Remember that:

- There are 24 hours in a day.
- There are 60 minutes in an hour.
- There are 60 seconds in a minute.
- There are 1000 milliseconds in a second.

He wants you to write a piece of code that prompts the user for number of days, hours, minutes, and seconds (**in that order)**.

It then calculates the **total** number of milliseconds in the given days, hours, minutes, and seconds. (For example, 0 days, 0 hours, 1 minute, and 2 seconds has 62000 milliseconds.) The program should then print a nice message regarding the total number of milliseconds.

Note: Your program should ask for input in the order specified above. Failure to follow this ordering will result in some point deduction.

Final Submission

To upload your homework for grading, log on to eLearning, select this class, and navigate to the "Assignment Submissions" area. Click on the "Programming Assignment 1" folder and upload both python files (mileage.py and time.py) in the designated location.

In addition to this, you should print paper copies of your code and design document. Please submit these stapled printouts in the following order in class the day the assignment is due:

- design document
- mileage.py
- time.py

  Note: Do not submit the design document online. Only submit the coding files (mileage.py and time.py) online. However, please print **all three** files to submit in class before th