

CS 1510: Intro to Computing - Fall 2017

Assignment 1: Writing simple programs

Due: Friday, September 7, 2017, by 11:59 p.m.

Introduction

This semester we will talk a lot about problem solving. Let's look at how this would work, at least for some small problems. 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 could 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 you will:

1. Look carefully at the two customer requests below (step #1 in the software development process).
2. For each request analyze and design a solution to the problem.
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
- Use meaningful variable names with the proper style (use `_snake_casing` - i.e., compound words are separated by underscores)
- Use meaningful constants where appropriate, and proper naming style (`ALL_CAPS_AND_UNDERSCORES`)
- Your programs should ask for input **in the exact order** specified in the customer requests below.
- Every file containing python code that you submit should contain a header comment block containing three pieces of information as shown below:

```
"""  
File: filename  
Author: your-name  
Description: one-line description of the file  
"""
```

Customer Request #1 (mileage.py)

You are contacted by an colleague who is analyzing the overall efficiency 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.

You must 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 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 a school teacher who teaches time management. He wants a tool that will allow her 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.

She 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 message regarding the total number of milliseconds.

Submit your work

Submit both files **time.py** and **mileage.py** to eLearning.

Remember, **no late work will be accepted**. Start early!