

Intro to Computing Homework #4 March 27, 2009 (11:59 PM)

This homework uses the same file of customer records as lab 6 which can be found at:

<http://www.cs.uni.edu/~fienu/cs051s09/labs/lab6.zip>

Recall that customer records are in the file `customerData.txt`. **Each customer record is on a single line with 12 fields separated by commas(‘,’)**. The order of the fields on a line is: First Name, Middle Initial, Last Name, Street Address, City, State, Zip Code, Country, Email Address, Telephone Number, Gender, and Birthday.

Your goal is to design and write a menu-driven program that provides the user the ability to:

- search for customer records based on user specified criteria and display the search results to the user
- save search results to a new file in one of two possible formats (as mailing labels or formatted just like the `customerData.txt` file). Allow the user to enter the file name for saving.
- update a customer record in the file of records (e.g., a customer might have moved or gotten a new phone #)

On startup before your program enters the menu-drive phase, it should:

- read each customer-record line in `customerData.txt` as a string,
- split the line into a list of strings for the fields,
- store this list for a customer into a list of customers (so you have a list of lists)

During the menu-driven phase, searches and updates should be on this list of customers. When the user selects from the menu that they want to exit the program, then the customers list should be written back to the `customerData.txt` file. (When debugging your program, you should probably write to a different file.)

I'll let you decide on how flexible of searches you'll allow in your program (e.g., females from Iowa), and give extra credit for really general capabilities. If you save the search results as mailing labels, each mailing label should be formatted as below with 3 blank line separating each label:

Jane Smith
123 Main Street
Cedar Falls, IA 50613

As always, when you write your program, be sure you:

- mock-up how you want your menu-driven system to behave before you try to write code
- think about the top-down design before you start to write code
- use meaningful variable names with good style (i.e., useCamelCase or use_underscores)
- use comments at the start of the program **and** before each function describing what they do (see Program 3-3 on pp. 88-90)
- use a main function (see Program 3-3 on pp. 88-90) located at the top of program with a call to it at the bottom to start execution
- use global constants where appropriate with good style (ALL_CAPS_AND_UNDERSCORES). (Put your global constants after your initial comments describing the program and before your main function definition so they can be found and changed easily in future versions of your program.)

Submit your homework electronically at http://math-cs.ens.uni.edu/~schafer/submit/which_course.cgi

Submit the following files:

- **customerDB.py** (your Python program)
- **design.doc** (or `design.txt`, or `design.rtf`) a document describing the design of your program including a hierarchy chart and a little text