

Computer Project #8

Assignment Overview

This assignment focuses on the design, implementation and testing of Python programs to process data files using functions and dictionaries, as described below.

It is worth 60 points (6% of course grade) and must be completed no later than 11:59 PM on Monday, November 16.

Assignment Deliverable

The deliverable for this assignment is the following file:

`proj08.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **handin system** before the project deadline.

Assignment Background

Most cell phones provide a mechanism to save and restore the owner's collection of contacts using a text file. If the format of that text file is known, a program which is not resident on the cell phone can be used to manage the collection of contacts.

Assignment Specifications

1. The program will display a menu which lists the following options:

- A) Read collection of contacts from file
- B) Write collection of contacts to file
- C) Add new contact
- D) Remove existing contact
- E) Update existing contact's phone number
- F) Update existing contact's email address
- G) Display contacts by prefix
- X) Exit from the program

The user will select an option by entering the appropriate character (upper or lower case). For example, the user will enter "A" (or "a") to select the first option (read contact list from file).

The program will support the exact set of options given above. You may not add options, remove options, or change the order of the options.

2. Option A will permit the user to read a collection of contacts from a file. It will prompt the user for the complete name of the text file containing the collection of contacts. The program will assume that the text file has the following format:

name (character string)
phone number (character string in the format DDD-DDD-DDDD)
email address (character string in the format local@domain)

A semicolon will separate one field from the next field. For example:

```
Susan;517-345-1234;susan_smith@anr.msu.edu  
Dennis Jones;989-123-4567;Dennis@gmail.com  
Miller, Matthew;616-765-4321;matt@yahoo.com
```

This option will clear all entries from the dictionary of contacts, then read the text file and insert each contact into the dictionary of contacts. The contact's name will serve as the key for that contact's information (phone number and email address).

You may assume the text file has been created by Option B. Thus, each contact name is unique, and each phone number and email address is formatted correctly.

3. Option B will permit the user to write the collection of contacts to a file. The program will prompt the user for the complete name of the text file to which the collection of contacts is to be written. The program will write the contents of the dictionary of contacts into that text file, using the format given for Option A.

4. Option C will permit the user to insert a new contact into the collection of contacts. The program will prompt the user to enter the name of the new contact, prompt the user to enter the new contact's phone number, and prompt the user to enter the new contact's email address (three separate prompts, in the specified order); it will then insert the new contact into the dictionary of contacts.

This option will verify that the new contact's name is unique, that the phone number is formatted correctly (only digits, the character "-" in correct locations), and that the email address is formatted correctly (the character "@" between the local part and the domain part).

5. Option D will permit the user to remove an existing contact from the collection of contacts. The program will prompt the user to enter the name of the contact; it will then remove that contact from the dictionary of contacts.

6. Option E will permit the user to replace the phone number for an existing contact. The program will prompt the user to enter the name of an existing contact, and then prompt the user to enter the replacement phone number.

This option will verify that the replacement phone number is formatted correctly.

7. Option F will permit the user to replace the email address for an existing contact. The program will prompt the user to enter the name of an existing contact, and then prompt the user to enter the replacement email address.

This option will verify that the replacement email address is formatted correctly.

8. Option G will permit the user to display a subset of the collection of contacts. The program will prompt the user to enter a name prefix; the program will then display all information about each contact whose name begins with that name prefix. That list of contacts will be displayed in sorted order, based on the contact names.

For example, if the name prefix is "Wi", then all contact names which begin with the characters "Wi" will be displayed.

If the user enters an empty name prefix, all contacts in the collection will be displayed.

9. The program will detect, report and recover from invalid user inputs. Whenever the user enters an invalid input, the program will display an appropriate message on the screen and will then immediately return to the menu of options. For example, assume the user selects option E (update existing contact's phone number), and then enters a contact name which is not present in the dictionary. The program will report the error to the user and will then prompt the user to select a menu option (it will not re-prompt the user to enter a different contact name, it will not prompt the user to enter a phone number, and it will not modify the dictionary of contacts).

10. The program will be composed of an appropriate set of functions. The program must define and use a separate function to process each of the menu options in the set {A-G}, and it must define and use at least two other meaningful functions.

Assignment Notes

1. Items 1-9 of the Coding Standard will be enforced for this project.

2. It is critical that your program uses the specified menu options, and that your program prompts the user for inputs in the exact order specified.

3. A significant emphasis in this assignment is on the definition and use of an appropriate set of functions to handle tasks within the overall program. As noted above, your program must contain a separate function definition for each of the menu options (with the exception of option X) and it must contain at least two other meaningful functions (you may define and use additional functions beyond the required functions).

4. Your program will remove leading and trailing whitespace characters from all user inputs. For example, if the user enters the string " William Smith ", the program will remove one space character from the start of the string and two space characters from the end of the string, resulting in the string "William Smith".

Suggested Procedure

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step may be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python statements, you must work on your own.
- Complete the program for one option (such as Option A) before working on the remainder of the project.
- Develop a simple version of the program, then run the program and track down any errors.
- Use the **handin system** to turn in the first version of your solution.
- Cycle through the steps to incrementally develop your program:
 - Edit your program to add new capabilities.
 - Run the program and fix any errors.
 - Use the **handin system** to submit the current version of your solution.
- You would be wise to back up your files on your H: drive, also.
- Be sure to log out when you leave the room, if you're working in a public lab.