Programming Assignment 6: Contacts Class Objects List

COP 3035 - Fall Term 2019
Point Value: 100 points
Project Due : Thursday 12/5/2019 at 11:59 PM

Reminder: this assignment is due at this specified time deadline and NO late work will be accepted!!!

Learning Objectives

- To write a program which requires the creation of a class data structure
- To write an initializer and other methods for the class, including accessor and mutator (get and set) methods
- To write an application program for the class to demonstrate its use with a *list* of class objects

Problem Statement

Create a *Contact* class that can hold information and perform operations with contact objects. The basic idea here is that on a mobile phone, you typically have a contacts list or address book. The *Contact* class must have the following *private* data members. Note that this table provides just the basic "word" names, you will have to use correct Python syntax to make these private members in your program.

Member	Description
name	string
address	string
age	int
phone	string
type	string (will be one of "NONE", "FRIEND", "RELATIVE" or "WORK")

The class must have the following public methods. Note again, only the basic names are provided here; you will have to determine the appropriate naming syntax and the return values, if any, for these methods.

Method	Description
__init__	sets name, address and phone to "unavailable", age to zero and type to "NONE"

setName	accepts a string argument that is copied to the name member variable
setAddress	accepts a string argument that is copied to the address member variable
setAge	accepts an integer argument that is copied to the age member variable
setPhone	accepts a string argument that is copied to the phone member variable
setType	accepts a string argument that is used to set the contact type member variable
getName	returns the value in name
getAddress	returns the value in address
getAge	returns the value in age
getPhone	returns the value in phone
getType	returns the value in type
__str__	as specified by the Python language's standard usage

Write this program in a *single* Python program file.

Springfield

Montgomery Burns
+11 092 8476 234

Ms Krabappel
23425234

Ned Flanders
839 923 923

Patty Bouvier

Principal Skinner
30498753

Selma Bouvier
239872394

Waylon Smithers

Demonstrate the class by writing an application for it using main and other functions. Note that your application will include functions other than main, which are not part of the class, but are needed to modularize your application.

Your application program must do the following:

1. Create a list of class objects which will store 5 contacts. These 5 objects are initialized by the initializer method.
2. Print out values for all the data members of all 5 contact objects in the list, at this initial point, which is right after the list creation.
3. Open the data file provided to you on the class web site, named *contacts.txt*. Then read in the first line containing the number of contacts for which data is provided, and next the actual contact data.

Each item of contact data is given on one line in the file. Read in the file data for the provided number of contacts and use it to replace the data in that number of contacts, starting at the beginning of the list.

4. You will hard code the file name into the program. If the file does not open when your program tries to open it, end the run by handling the standard exception thrown, as shown in lectures, along with printing an error message for the user.
5. After you have read in the provided file data, print out the number of actual contacts provided in the file and the data for each contact. At this time you will only print contact data for the part of the list that has meaningful contact information (which may be fewer than 5).

Input and Other Requirements

The only input is from the data file. You may assume the file data is completely correct and you do not have to do any error checking on file contents. You may assume that the number of contacts specified in the file is an integer between 1 and 5.

Output

- As usual: introduction, echoprinted input, closing termination message, error messages as needed, and any informative messages the user may need or want to see
- Elements described in this write-up
- Follow the course style guidelines

Be creative!

Miscellaneous

You must set up the class exactly as specified in this write-up, with class members exactly as described. You may not add or delete data members or function members from the class. Members must behave exactly as described in this write-up.

You will most likely have non-member functions in addition to main in your program, as your program needs to be modular and use functions and parameter passing appropriately. For example, a function in your program called "print_heading" which printed out introductory program output, would be a non-class-member function.

What File To Turn In, File Naming Requirements, and How to Turn In Your Work using Canvas

You must turn in your Python program source file which must be named as follows (note that you will have to rename the provided file):

Use this format: *yourLastNameLowerCase_FSUID_p6.py*
Your FSUID will be unique to you, will typically be your FSU email ID, and will be something like "ab23c." Hence file names will look something like "smith_ab23c_p6.py"

Submit your Python file (.py) to Canvas using the Submit button for this assignment. Be sure to download the file after you submit it in order to check that you submitted the correct program file to Canvas and that it was successfully received by Canvas.

Last Update: 7/29/2019 A. Ford Tyson