

CSE 1384 - Classes

Lab 5

Objectives:

- Continue practicing past concepts
- Practice using basic C++ classes

Assignment:

Create a program to utilize the given C++ class (**do not change the class given to you**). The program will use the Person class to aid in creating an address book to manage the user's contacts. The initial contact list will be loaded from a file. The file **contacts.txt** has been provided for you as a base point. Each row in the file should be organized like so:

```
firstName lastName phoneNumber emailAddress
```

Each item will be separated by a single space.

Your main program should have four functions:

- main
 - Parameters: none
 - Return values: 0
 - Purpose: refer to main program guidelines below
- readContacts
 - Parameters: contacts vector, opened file OR string filename
 - Return values: nothing OR string filename
 - Purpose: read the contents of the file in a loop. Populate a Person object with each appropriate item and add that Person to your Contacts vector.
- addContacts
 - Parameters: contacts vector, string filename
 - Return values: none
 - Purpose: get input from the user for first name, last name, phone number, and email – add where appropriate to the vector and file (hint: this will REQUIRE altering the file)
- displayContacts
 - Parameters: contacts vector
 - Return values: none
 - Purpose: display your vector contents

Main program:

Create a Person type vector and receive a string file name.

Either: loop to verify a valid file for opening and THEN call the read contacts with an opened file. OR, call the read contacts with your vector and file name and verify the file exists in the function. You should return the valid file name back to the main so it's saved for use in functions like add contacts.

Then, display a menu to the user. Loop while they want to continue in the program.

- Exit
 - Exits the menu/program
- Display Contacts
 - Calls display function
- Add Contacts
 - Calls add contacts function
- Incorrect menu option
 - Displays error in case the user has entered an invalid option

Hints:

Your vector should NOT be a global – but, also, no functions return the vector. Keep in mind, your vector should retain changes made in functions. How can we grant this behavior? Passing by...?

If you're passing an opened file in read mode to your read contacts function, this must also mimic the parameter type referenced in the first part of this hint!

BONUS / Honors Student Credit:

If you're an honors student, you must complete this portion. If you do not, it will be counted against you. If you're not an honors student, you may complete this section for an additional 10 bonus points.

Create either a delete contact or edit contact function, the choice is yours (only one will count, though). The parameters on both would be the contacts vector and a string file name. Then, create a corresponding menu option to be able to call the function.

Both delete and edit should utilize the view contacts function to show what's currently present and allow the user to choose which to edit/remove. Edit at least two categories to count for an edit function. Both delete and edit should then change the contacts.txt file accordingly (you'll want to overwrite here).

Example Execution:

```
Welcome to your address book manager!
Please enter a file to read your contacts from (include extension): contacts
That file doesn't exist. Try again.

Please enter a file to read your contacts from (include extension): contacts.txt

File opening ...
File read. Closing the file from read mode.

Menu:
0. Exit
1. Display Address Book
2. Add Contact
What would you like to do? 1

Address book:

Contact 1:
Name: John Lennon
Phone number: 555-5555
Email address: frontman@beatles.com

Contact 2:
Name: Ringo Starr
Phone number: 456-123-7890
Email address: drums@beatles.com

Contact 3:
Name: Paul McCartney
Phone number: 123-456-7890
Email address: newband@wings.com

Contact 4:
Name: George Harrison
Phone number: 789-654-4321
Email address: aguy@beatles.com

Menu:
0. Exit
1. Display Address Book
2. Add Contact
What would you like to do? 2

Here's where we'll gather the information.
What is the first name? Stevie
What is the last name? Wonder
What's the phone number (no spaces)? 543-678-0631
What's the email address? adude@music.gov

Menu:
0. Exit
1. Display Address Book
2. Add Contact
What would you like to do? 3

That's not a valid option. Try again.
Menu:
0. Exit
1. Display Address Book
2. Add Contact
What would you like to do? 0

Good-bye!
```

Comment Block:

Your code should contain a comment block at the top containing information on who wrote the code, what the assignment is, when it is due, etc. Here is an example of a good comment block to put:

```
/*  
Name: <your name>                                NetID: <your netID>  
Date: <current date>                              Due Date: <enter in due date>  
  
Description: <What is the program?>  
*/
```

Deliverables:

- C++ code (.cpp file)
- A document (.pdf) with two screenshots showing the program running
 - The two program screenshots should have completely different inputs from each other
 - The two screenshot must be *legible* to count (too small or pixelated text will not be interpreted)
 - Show *all* possible error messages

Point Breakdown:

(100 points total)

A submission that doesn't contain any code will receive a 0.

- 5pts - no global variables
- 5pts - IO handled correctly
- 15pts - file handling
 - 5pts - looped correctly while non-existent file
 - 5pts - read from the file correctly
 - 5pts - appended to the file correctly (did *not* overwrite contents)
- 20pts - classes
 - 5pts - created objects correctly
 - 5pts - included class correctly
 - 10pts - correctly utilized getters/setters for data
- 10pts - vectors
 - 5pts - vector and items are of type *Person* not strings
 - 5pts - correctly added/viewed *Person* object in vector
- 15pts - user defined functions (3)
 - 6pts - parameters/return values correct (2 points per function)
 - 9pts - functions correct (handling what they should) (3 points per function)
- 10pts - menu handling
- 10pts - two unique screenshots
- 10pts - programming style *
- (penalty) -15pts if the class provided must change for it to work with your code
- Honors / BONUS (dependent on Honors status):
 - Honors: 10pts penalty for not completing
 - BONUS: 10pts added for completing

* Programming style includes good commenting, variable nomenclature, good whitespace, etc.