CECS 277 – Lab 4 – Classes & Objects

Rolodex

Create a program that allows the user to view, search, and modify a contact list made up of contact objects. A contact has a name, phone number, address, city, and zip code. Contacts are initially read in from the file 'addresses.txt' and then are written back to the file when the program ends.

Create a Contact class (in a separate file named 'contact.py') with the following:

1. Attributes – `first_name`, `last_name`, `phone`, `address`, `city`, `zip`.
2. `__init__(self, fn, ln, ph, addr, city, zip)` – pass in the contact's information and assign each one to their corresponding attribute.
3. `__lt__(self, other)` – passes in and compares two contacts. Compare by last names, if they are the same, then compare by first names. This method will be automatically used when you sort your list of contacts.
4. `__str__(self)` – returns a string that is used to display the contact to the console.
5. `__repr__(self)` – returns a string that is used to write the contact to the file in the format 'first_name,last_name,phone,address,city,zip'.

In a the main file ('main.py'), create the following functions:

1. `read_file()` – open the file, read in each contact (one per line), construct a Contact object using the data, and then store the Contact object in the list of contacts. Sort and then return the filled list.
2. `write_file(contacts)` – passes in the list of contacts. Open the file for writing, loop through the contacts list and write each contact to the file using the repr method. Write each contact on a new line.
3. `get_menu_choice()` – display the main menu to the user and then take in and return the user's valid input.
4. `modify_contact(con)` – pass in a contact object. In a loop, display the modify menu to the user, get the user's valid input, then, based on the user's choice, prompt the user for the information they'd like to change, and then update the appropriate attribute for the contact. The user may repeatedly change any of the contact's values until they choose option 7 to end the loop.

In the main function, read in the file and then repeatedly display the main menu and allow the user to make a valid selection. Perform the selected option:

1. Display Contacts – display the number of contacts in the list. Then display an enumerated list of each of the contacts using the str() method.
2. Add Contact – prompt the user to enter each piece of data for a contact. Construct the contact using these values, then add it to the end of the contacts list. Sort the list.
3. Search Contacts – prompt the user to enter the type of search, by last name or by zip code. Prompt the user for the search data, then loop through the contacts list comparing the user's input to the appropriate attribute for a contact. Display all matches.

4. Modify Contact – prompt the user to enter the first and last name of the contact they'd like to modify. Print the contact and then call modify_contact to allow them to update any part of the contact they'd like. Sort the list of contacts.
5. Save and Quit – call the write_file function to write the contact list back to the file. Display a message that the file has been saved and the program is ending.

Example Output (user input is in italics):

```
Rolodex Menu:
1. Display Contacts
2. Add Contact
3. Search Contacts
4. Modify Contact
5. Save and Quit
> g
Invalid input - should be an
integer.
> 1
Number of contacts: 6
1. Ned Flanders
555-4983
740 Evergreen Terrace
Springfield 90122

2. Hank Hill
555-9834
84 Rainey Street
Arien 50667

3. Sherlock Holmes
207 224 3688
221 Baker Street
London 19303

4. Herman Munster
555-1313
1313 Mockingbird Lane
Pleasantville 10394

5. Homer Simpson
555-9802
742 Evergreen Terrace
Springfield 90122

6. Spongebob Squarepants
555-8942
124 Conch Street
Bikini Bottom 10022
```

```
Rolodex Menu:
1. Display Contacts
2. Add Contact
3. Search Contacts
4. Modify Contact
5. Save and Quit
> 2
Enter new contact:
First name: Lisa
Last name: Simpson
Phone #: 555-9803
Address: 742 Evergreen Terrace
City: Springfield
Zip: 90122

Rolodex Menu:
1. Display Contacts
2. Add Contact
3. Search Contacts
4. Modify Contact
5. Save and Quit
> 3
Search:
1. Search by last name
2. Search by zip
> 1
Enter last name: Simpson
Homer Simpson
555-9802
742 Evergreen Terrace
Springfield 90122

Lisa Simpson
555-9803
742 Evergreen Terrace
Springfield 90122

Rolodex Menu:
1. Display Contacts
```

```
2. Add Contact
3. Search Contacts
4. Modify Contact
5. Save and Quit
> 3
Search:
1. Search by last name
2. Search by zip
> 2
Enter zip code: 90122
Ned Flanders
555-4983
740 Evergreen Terrace
Springfield 90122

Homer Simpson
555-9802
742 Evergreen Terrace
Springfield 90122

Lisa Simpson
555-9803
742 Evergreen Terrace
Springfield 90122

Rolodex Menu:
1. Display Contacts
2. Add Contact
3. Search Contacts
4. Modify Contact
5. Save and Quit
> 4
Enter first name: Herman
Enter last name: Munster

Herman Munster
555-1313
1313 Mockingbird Lane
Pleasantville 10394

Modify Menu:
1. First name
2. Last name
3. Phone
4. Address
5. City
6. Zip
7. Save
> 3
Enter phone #:555-1234
Modify Menu:
1. First name
2. Last name
3. Phone
4. Address
5. City
6. Zip
7. Save
> 6
Enter zip:10313
Modify Menu:
1. First name
2. Last name
3. Phone
4. Address
5. City
6. Zip
7. Save
> 7

Rolodex Menu:
1. Display Contacts
2. Add Contact
3. Search Contacts
4. Modify Contact
5. Save and Quit
> 3
Search:
1. Search by last name
2. Search by zip
> 1
Enter last name: Munster
Herman Munster
555-1234
1313 Mockingbird Lane
Pleasantville 10313

Rolodex Menu:
1. Display Contacts
2. Add Contact
3. Search Contacts
4. Modify Contact
5. Save and Quit
> 5
Saving File...
Ending Program
```

Notes:

1. You should have 3 files in your project: contact.py, check_input.py, and main.py.
2. Check all user input using the get_int_range function in the check_input module.
3. Please do not create any extra attributes, methods, parameters, or functions.
4. Please do not create any global variables. You may access the attributes of the Contact object directly for this assignment because they are all strings.
5. Use docstrings to document the class, each of its methods, and the functions in main.py. See the lecture notes and the Coding Standards reference document for examples.
6. Place your names, date, and a brief description of the program in a comment block at the top of your main. Place brief comments throughout your code.
7. Thoroughly test your program before submitting:
    a. Make sure that all of the contacts are created from the file (ie. 18 contacts).
    b. Make sure that all of the menu inputs are validated. You do not need to validate the contact's data since they are all strings.
    c. Make sure that all of the contacts are numbered and displayed in sorted order by last name, and if the last names are the same, then they are in order by first name.
    d. Make sure that the new contact's data is added correctly (ie. the phone number isn't where the first name should be), and the contact is added to the list.
    e. Make sure that when searching, all of the matches appear, not just the first one.
    f. Make sure that when modifying, the user can repeatedly update any of the contact's values until they leave the modify menu.
    g. Make sure that when saving and quitting, the contacts are written in the correct format from the repr() method and that it overwrites the contents of the old file (doesn't append to the file). Check by running the program again to ensure that the contacts can still be read in properly and that there are the correct number.

Rolodex Rubric – Time estimate: 3 hours

Rolodex **10 points**	Correct. 2 points	A minor mistake. 1.5 points	A few mistakes. 1 point	Several mistakes. 0.5 points	No attempt. 0 points
Contact class (in a separate file): 1. Has attributes first_name, last_name, phone, address, city, zip. 2. Has methods: __init__, __lt__, __str__, and __repr__. 3. __lt__ compares by last name and if they're the same, by first name. 4. __lt__ returns true if self < other. 5. __str__ returns string for display. 6. __repr__ returns string for file.					
read_file and write_file functions: 1. opens file, constructs a contact from each line, and adds it to contact list. 2. returns list of contacts. 3. writes all contacts to file with repr().					
get_menu_choice and modify_contact functions: 1. Displays menu. 2. Validates user input. 3. Repeatedly updates contact data until user exits submenu.					
Main Function: 1. Creates list of contacts. 2. Displays enumerated list of contacts. 3. Correctly adds new contact to list. 4. Correctly searches contacts by name. 5. Correctly searches contacts by zip. 6. Correctly finds matching contact for updating. 7. Correctly updates contact. 8. Validates all menu input. 9. Repeats until user quits. 10. Correctly writes back to file 11. Correctly reads in updated file.					
Code Formatting: 1. Code is in functions/methods. 2. Correct spacing. 3. Meaningful variable names. 4. No global variables. 5. Correct documentation.					