CSE 107 Midterm Exam Spring 2015

NMT Computer Science Department

Name: ____

Due Date: March 13 5pm.

Submit: Online to Canvas

By printing your name, you agree to not share any of the information contained within this test in any form including but not limited to written, oral, and electronic. Failure to do so may result in disciplinary measures.

Note: When writing any code you will be graded on clarity of code not just correctness.

Questions: If you have a question and cannot find any in-person help, email me at jirhiker@gmail.com (subject: Midterm) for prompt responses. If you message me using Canvas I may not respond as quick as you hope.

Print this page out, sign, and hand in with your completed exam

	Student Score	Possible Score
Part 1. Design		25
Part 2. Implementation		75
Extra Credit		
% Total		100%

The Problem

You are an employee of, RossLabs, a small software consultation firm for noble gas mass spectrometry. At this week's weekly developer's meeting you were tasked with developing an application to manage the firm's contacts. The CEO and CTO of RossLabs are looking for something that is text-based and easy, so the firm can quickly start calling contacts for the latest round of venture capital fundraising. Heretofore the firm has been using a simple CSV file to save all of its contacts. Everyone in the organization is looking for an improved solution and are counting on **you** to have a **working prototype** by the end of next week (Friday, March 13).

The CTO has specified a set of tasks your contacts application must perform in order for you to get paid (pass your exam). Please examine the requirements carefully, implement as many of the tasks as possible including the Extra Credit (E.C worth 1% each)

1 Design

Provide a description of how you plan to solve this problem. Draw figures, flowcharts, and tables with detailed annotations (notes). Consider and describe typical use cases. For example,

1. application starts.
2. user wants to add a contact.
3. program sequentially asks for required information validating each inputted value.
4. program adds contact to its stored contacts.
5. the added contact information is displayed to the user.

2 Implementation

Make your code clean and readable. You will be graded on style and functionality. **You must follow PEP8 and PEP257**.

README

Write a README.txt file. This is common practice in the open source community and provides a consistent location for users to find introductory information about your application. This file should contain a brief description of what the application does and some information on how to use it. For example, you should provide a list of valid commands, their function, and examples of their use.

Contact Application Requirements

- Basic
  - Think of and display a unique name for your application
  - Display a welcome message when the application starts.
  - Display a goodbye message when the application quits.
  - E.C use a random welcome and goodbye message chosen from a set of available messages.
- Required functionality
  - Minimum attributes to store
    - name
    - phone
    - company
    - email
    - note
  - Get input from user
  - Load contacts from file.
    - Load a default contacts file when the application starts
    - Warn user if entered invalid file
  - Save contacts to file
    - Write contacts as a CSV file
    - E.C verify path can be written to, i.e parent directory exits
  - Add a contact

    - report contacts updated
    - E.C validate user input. verify phone, email correct formats
  - Edit contact
    - report contact's updated info
    - E.C validate user input.
  - Remove contact
    - Report contacts updated
    - Warn user if trying to remove contact that doesnt exist
  - List contacts
    - E.C filter contacts. e.g show all contacts with name starting with __________.
  - Info
    - Number of contacts
    - Number of companies
    - Number of contacts per company
  - About
    - Show developer information. developer, date, ... etc
  - Lookup
    - Display the contacts information to the user in a nicely formatted view.
    - Warn user if specified contact does not exist
  - Note
    - Allow user to edit a note associated with a specified contact
    - Allow user to see current note
    - E.C Allow multiple notes for a given contact. User should be able to list, add, remove notes.
  - Load a set of commands from a file and execute them. An example command file **might** look like the following.

```
add joe
  phone: 555-5555
add jane
  phone: 444-4444
  company: Four
remove bob
```

- Extra Credit
  - E.C Stay in each mode before asking user for another command. For example if user wants to add a bunch of contacts, enter 'add' mode, add contacts manual, and explicitly exit add mode with 'exit'
  - E.C plot contacts. e.g plot bar graph of number contacts vs company
  - E.C group contacts. add ability to manage groups of contacts. Contacts may exist in multiple groups. Add an additional command set for working with groups, i.e add, remove, list groups, list contacts in group etc.
  - E.C Import contacts from a csv file
  - E.C Export contacts to YAML, XML, JSON, etc...
  - E.C Associate an image with each contact. Add a command to display the contact's image.
  - E.C Add user login. Each user has a distinct set of contacts that is automatically loaded when the user logs in.

Example Session

An example session using the Contacts app is show below. A user starts the application with Line 1, enters the commands, `about, info, list` and finally `exit`. The application displays a farewell message and quits. **Note: Lines 1 and 17 are terminal prompts**.

```
1  ross$ python contacts.py
2  Welcome to the Contact application
3  Please enter a command: about
4  Contacts App. Developed by Jake Ross for CSE107 2015
5  Please enter a command: info
6  contacts path: contacts.txt
7  number contacts: 3
8  
9  Please enter a command: list
10 Name                  Phone         Company              email
11 ---------------------------------------------------------------------
12 Elon Musk           : 453-6723      SpaceX              emusk@spacex.com
13 Larry Page          : 853-0653      Google              lpage@gmail.com
14 Tim Cook            : 133-0419      Apple               tcook@apple.com
15 Please enter a command: exit
16 Goodbye
17 ross$
```

Submitting

Submit your contacts application files as a tarball to canvas. Include a README file with a brief description of the application and how to use it. Include your design notes as a text file or PDF. **(Note: If you want to include hand drawings, scan and include as PDFs or drop off hardcopy at my office)**. Print out the first page of this document, sign, and drop off at my office or email a scanned copy.