Term Project: Iteration 3 – GUI Application

(Fall 2023)

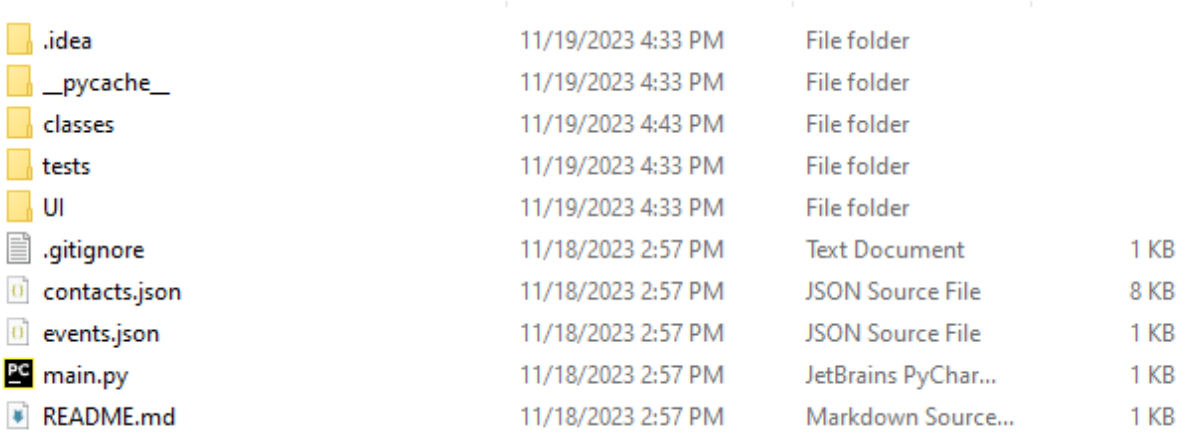
Project Description:

In this iteration, you will be given a rudimentary version of the application that uses a graphical user interface (GUI) rather than the terminal. This application contains options for creating and displaying contacts, creating and displaying events, associating contacts with an event, and creating a memo attached to an event attendee. An existing list of contacts and events are read into the application via a JSON, similar to Iteration 2.

Some parts of the code (in **Event_Manager.py** and **Event_Attendee.py**) have been removed. Your part in this assignment is to complete the classes and methods necessary for this application to run, such that it functions as shown in the demo video.

Instructions:

1. Download **iteration_03.zip** from iLearn and unzip the file. The contents should appear as follows:



.idea	11/19/2023 4:33 PM	File folder	
__pycache__	11/19/2023 4:33 PM	File folder	
classes	11/19/2023 4:43 PM	File folder	
tests	11/19/2023 4:33 PM	File folder	
UI	11/19/2023 4:33 PM	File folder	
.gitignore	11/18/2023 2:57 PM	Text Document	1 KB
contacts.json	11/18/2023 2:57 PM	JSON Source File	8 KB
events.json	11/18/2023 2:57 PM	JSON Source File	1 KB
main.py	11/18/2023 2:57 PM	JetBrains PyChar...	1 KB
README.md	11/18/2023 2:57 PM	Markdown Source...	1 KB

2. For this assignment, you will need to modify only two files:
    - Event_Manager.py
    - Event_Attendee.py

3. In **Event_Manager.py**, complete the following methods:
    - `def uid_to_contact(self, uid: int) -> Contact:`
    - `def uid_to_event(self, uid: int) -> Event:`
    - `def is_attending(self, c: Contact, e: Event) -> bool:`

4. In **Event_Attendee.py**, complete the constructor (`def __init__(self, e: Event, c: Contact):`) and any methods necessary for this class to function (getters, setters, etc.).

5. Run **main.py** and check whether the program functions correctly. Attempt to view and create contacts and events, associate contacts with events, add memos to event attendees, etc. If you do not encounter any bugs or errors, it is likely that you have completed the assignment correctly.

Notes:

If you have trouble, the following notes may help you in completing the assignment:

- The code that you must write to complete this iteration is neither lengthy nor complex. Most methods can be completed in one to three lines.

- The arguments needed for your methods are already included in the code. You should use this knowledge to inform your solution. For example, the **uid_to_contact** method clearly shows that it takes one argument (apart from *self*) which is an integer for the UID. Therefore, if your implementation requires more than one argument, it is likely incorrect.

- Reading the comments in the various files will help you understand what each part of the code is doing, which can then help you understand what your parts of the code need to do.

- Remember that you are not writing your own program from scratch. Whatever code you produce (variable names, method names, etc.) must be compatible with the rest of the application.

- Depending on which version of Python you are using and which IDE you are using, you may receive different errors / warnings for this application. If you are receiving an error that is **unrelated to the code you are writing for this assignment**, debug it yourself as best as you can or contact one of the TAs. If neither option is fruitful, contact me and I will do what I can to resolve the issue.

Deliverables:

Once you have completed your application, place all necessary files into a .zip file named "userid_iteration_3.zip" where "userid" is your Tech username. For example, "jstrickler_iteration_3.zip." Submit the .zip file to the Iteration 3 dropbox in iLearn. Make sure that all necessary files are included with your program and that it can be easily run by myself and/or the TAs.