

Python Application Development Assignment – 2

Phone Organiser

Practical Work

Using Jupyter Notebook, develop an object-oriented application that can organize phone records in a phone book. You have to realize a class `phoneRec` holding the information of name, nickname, phone number (an 8-digit integer assuming not started with 0), email address, last-call date-time (e.g., 8 November 2022 13:28 can be represented by the string "2211081328".) You also need to realize a class `phoneBk` holding a number of `phoneRec` objects under three groups: Family, Friend, and Junk. Every group can be realized as a list of `phoneRec` objects. (Hence, a `phoneBk` object has three member lists of `phoneRec` objects.) The number of valid elements (`phoneRec` objects) of each list can be kept inside the `phoneBk` object. Upon the user's request, the application should allow a phone record input and store it into a `phoneBk` object under one of the three groups specified by the user. Any record can be under more than one group, and your application does not need to check this. Upon the user's request, the application can display the phone records' information under a user-given group.

Within the `phoneBk` class, you are required to write public member function(s) to realize one of the following five tasks for editing the phone book. The task to do should be determined by the remainder obtained from dividing your team number by 5. For example, if your team number is SEVEN, $7 \% 5 = 2$, and you should do Task 2) of this assignment.

Task:

- 0) The member function removes the phone record of a user-input phone number under a user-given group. After removing a phone record, all the elements in the list that follow the removed record should be moved forward to ensure the valid records are stored consecutively. Then, update the number of valid elements in that list accordingly. If, for whatever acceptable reason, the removal cannot be done, show an error message, and then do nothing.
 - 1) The member function displays a sorted list of phone records based on the last-call date and time (latest first) under a user-given group. [Hint: "Peter" is *larger* than "Mary" (because 'P' is larger than 'M' in Unicode.)] The items in the displayed sorted list include the nickname and last-call date and time in a proper output format, e.g., the string "2211081328" should be shown as "08 November 2022 13:28".
 - 2) The member function checks the validity of all input email addresses under a user given group. For a valid email address, the character '@' must be found in it once. Moreover, some characters must be before and after the character '@' (e.g., a12bc@de3f.com). There must be at least one '.' character in the string following the character '@'. If an email address is invalid, the function should display an error message that tells the nickname and the invalid email address. Otherwise, the function should display a message that tells no invalid email address is found in that user group.
 - 3) The member function displays a sorted list of phone records based on the nickname under a user-given group. [Hint: "Peter" is *larger* than "Mary" (because 'P' is larger than 'M' in Unicode.)] The user needs to specify whether the sorting is done in ascending order or descending order. The items in the displayed sorted list include the nickname and phone number.
 - 4) The member function copies the phone record of a user-input phone number from one user-given group to another user-given group. The copied record should be the new last valid element of the list. Then, update the number of valid elements in the list accordingly. If, for whatever acceptable reason, the copying cannot be done, show an error message, and then do nothing.
- It is required that the classes and the implementation of their member functions should be built as a separate module to be imported by the application.
 - Your application is expected to provide a text menu as a user interface. Through the interface, the user can repeatedly organize the phone book's content according to your task. The user can also choose to end the application through the interface.
 - Should you want to get a credit grade, you should design the application such that when a user wants to start it, he needs to login with a username first, which will be checked with a file that stores username-password pairs. If the username is new, the application will ask for a password from the user and store it into that file. If the username can be found in that file, the application will ask for a password from the user, and the program will start only when the password is correct. The application will end if the user

fails to provide the correct password in three consecutive trials.

- Should you want to get a distinction grade, you should add and implement the following member functions in the class that:
 - i) Save the content of the current phone book into a file
 - ii) Read the current phone book's content from a file and store the information into the three group lists.

Report

Your report should include:

Abstract: Summarise the objectives and achievements of your assignment in less than 100 words.

1. **Introduction:** Describe the assignment's objectives and requirements in detail and give a brief account of the methodology.
2. **Methodology:** It contains
 - how your team divides the work among the team members. It will be used as a reference for assessment.
 - the schedule and steps of developing the project,
 - the detail of the developed application, including
 - the specifications of the classes defined, and the member functions/variables inside - explain as far as possible why your group makes such choices for the class members
 - The flow of execution. (It is good to include a flow chart to help illustration.)
 - what problems your group encounters, and how your group solves the problems,
 - how you validate your application, i.e., confirm that the solution is correct.
3. **Results**
 - Include the results of executing your application captured from the screen.
4. **Conclusion and further development**
 - Summarize the experience gained in the assignment
 - Indicate how your application can be extended and improved if more time is allowed.

The report should be in PDF format with your class number, team number, student names, student IDs, and task number on the front page. It is NOT required to include the complete source code in the report. Instead, you should zip the folder(s) containing all your project folders and files (including your report file) into a zipped file for submission. (See the General Description below.)

General Description

1. Each team should comprise THREE students. Students must obtain prior approval from the subject lecturer if they want to form a team with fewer or more team members.
2. Unless you get prior approval from your subject lecturer, you must observe the following:
 - Do NOT use any technique or Python constructs not taught in the subject
 - Unless mentioned on this instruction sheet, any imported library function not mentioned in the subject must NOT be used.
3. Each team should upload the zipped file to the Blackboard (under Groups, select File Exchange after clicking your team number) on or before **26 Nov. 2022**.
4. Ensure your submitted code can successfully run in Jupyter Notebook. Assessment will only be made based on your submitted zipped file. To lower the chance of incomplete submission, you are advised to zip one whole folder that contains all sub-folders and files for this assignment.
5. The documentation for your assignment is important. Your ability to write good comments for the source code will also be an important factor in the final assessment of your assignment. A marking scheme for this assignment will be announced in due course.
6. It is compulsory to use a word processing tool to write your report. The font size must not be bigger than 12 or smaller than 10. Use 1.5 lines spacing. Including all figures and tables, if any, the length of the report should not be shorter than 7 pages.