Lab Assignment 12, Object-Oriented Programming, CSE 271, Spring 2020
Department of Computer Science and Engineering, Miami University

Object Stream and Binary File

In this lab, you will use object streams (ObjectInputStream and ObjectOutputStream) to read and write information from and to a binary file, respectively. Create a project in Eclipse named **Lab_12.** You are going to design two classes in this project. Your program will consist of the following two classes: **Contact** and **AddressBook**.

Class Contact:
Create a class **Contact** that can store a single contact information. The Contact class should store contact's first and last name (separately), phone number, email and address. Write toString(), equals(Object), and accessor and mutator methods for all instance variables. The toString() forms a string in the following format: "Sarah Williams, 100 Main St, Miami, FL 30309, swilliams@gmail.com, 123-45-6789". Two contacts are equal when they have the same name and the same phone number.

Class AddressBook:
Create a class **AddressBook** with the main() method. It uses an instance variable **ArrayList<Contact>** to store a list (database) of contacts. It offers the following options, in an infinite loop, to the user:

```
Address Book Operations:
    1)  Add
    2)  Remove
    3)  Save
    4)  Load
    5)  Display All
    6)  Search
    7)  Exit
Select an option (number): 2
```

The options are: (the green color 2 in the above figure is the user input)

1. Add a contact to the list (ArrayList object)
    - Ask (prompt user) for all the contact information and take input from the keyboard. Then create a Contact object and add it to the list.
2. Remove a contact from the list (ArrayList object)
    - Ask (prompt user) for the phone number and remove the associated contact from the list. The phone number is unique.
3. Save all contacts to a file (file name: "**addressbook.dat**")
4. Load all the contacts from a file (file name: "**addressbook.dat**")
5. Display all contacts (Name, Address, Email and Phone Number). You can use the toString() method to display a contact.
6. Search for a specific contact and display it
    - Take a string input from the keyboard as a search string
    - The searches should find all contacts where any of the instance variables contains the target search string. For example, if "**abc**" is the search string, then any contact where the first name, last name, phone number, email, or address contains the search string "**abc**" should be shown on the display. You can use the toString() method to display contacts.
7. Exit from the program
    - Save all the contacts from the list to the file and exit.

Lab Assignment 12, Object-Oriented Programming, CSE 271, Spring 2020
Department of Computer Science and Engineering, Miami University

Object Stream and Binary File

Your program should keep running while performing these operations and only quit if the user selects Exit from the menu. In that case, you should save all the contacts from the list to the file (before you quit from the program). When you restart your program, you should retrieve all the contacts from the file. The first time you run the program (file does not exist!) you should create an empty file (**addressbook.dat**).

Important Note:

- There is no formatting guideline for this lab. Please use appropriate prompts (print text) when you ask for input and display information.
- You need to offer the options in an infinite loop until the user selects the option 7 (exit).
- This is a console-based application, you do not need to create any GUI.
- You should use ObjectInputStream to read each Contact object from the file and ObjectOutputStream to write each Contact object to the file. You can also read or write the ArrayList object rather than reading or writing individual Contact objects, choice is yours.
- Add appropriate exception handling when performing file operations.

Javadoc Style Comments:
You have to make Javadoc style comments for all classes and methods including parameter and return description. You don't need to submit "doc" with your code submission.

Submission:
Make sure the file names are correct and code is well commented. No late submission. You will get zero for late submission. Submit the java files to the appropriate submission folder on the Canvas. You can zip all the java files together and submit one zip file.

Grading Rubric:

Correct Contact class	10
AddressBook Class	
ArrayList object of Contact class	2
Prints menu in a loop	2
Exits only if the user selects option 7	2
Add operation	8
Remove operation	8
Save operation (saves using ObjectOutputStream and works correctly)	15
Load operation (reads using ObjectInputStream and works correctly)	15
Display All operation	8
Search operation (finds all contacts using the search string by comparing all instance variables)	10
Exit operation (Writes to the file when program quits)	10
Reads from the file when program starts	5
Creates an empty file if file doesn't exist	5
Javadoc	
Javadoc style comments (Only deduction, -10)	0
Total	**100**