Object Oriented Programming Lab

Final Project - C++

Muhammad Yousaf

|  |
| --- |
| **Topics:**   * **Making an Object Oriented** Application * Use of **Inheritance, Polymorphism, Operator Overloading & templates** * **Three file structure** * **File Handling** * Everything you studied till now |

# Project Brief

This project will provide you an opportunity to test and review all the concepts that you studied during the semester. The goal for the final project is to extend the Mini Project: ContactsBook application, with some new features to make it more realistic and useful!

Extending the mini project might be rather easy using Windows Forms instead of console. Let's discuss!

The project requirement is to be developed on console. Graphical User Interface with Windows Forms is a bonus and recommended for top groups, you can discuss with me! Currently, your project will be only usable through one computer, which is yours. You can continue working on it to **turn it into a website or an Android app this summer ☺ , interesting, no?**

# Policies

1. Project will be done in groups of 2
2. You really need to understand this document. And strictly follow the instructions provided to you
3. Plagiarism policies you already know. ☺
4. Your Viva will be coding based in the Lab and you’ll be asked to modify/extend any function of this project.
5. Your final marks will be dependent upon the percentage you get in viva and quality of code. Even if your marks in code are 100%, viva and code quality marks will affect these marks directly.

**HAPPY CODING**

Extending the Mini Project: Advanced Contacts Book

Design Changes

1. Advanced Searching feature (details are mentioned below in features): Introduce a new searching method, that just takes a string and searches in all the contacts
2. For groups you will need to have some unique\_id (int or string suitable) to implement the groups feature. As you can make a Class Group that has a list of contact IDs. It will make listing all groups of a contact and other group related features easier.
3. You will need a resizable array (list) in multiple places such as for storing search history, storing viewed contacts and so on. So you should create a generic resizable list using Templates and reuse its objects in all the places where you need them. This list class will store objects, and have functions like AddItem, RemoveItem, GetSize, GetArray etc. Leave the contacts array in ContactsBook the same. **And create a reusable List class for SearchHistory, Frequently viewed contacts, Groups etc.**
4. As, you will need to do sorting in at least two places, you shall write a reusable Template method for sorting in some separate header file. Whatever class is using sorting shall provide a **compare method** (enforce the implementation of this function using an interface/abstract named ComparatorInterface, according to class contract binding as we discussed in Polymorphism lab, you can create an abstract class with a method named compare which is pure virtual and inherit classes from this class)**. You can take help from me in reusability stuff! (Optional**

**Requirement for section-2D)**

Design: The sorting will be a template method, it shall take an arrayof T type for sorting, its size, and a “compare” function(same as we did in lab) by function pointer. The class that needs to do sorting of its elements, will call this function and pass the array, size and its compare function by function pointer **(we will do this example in next last lab InShaaAllah).**

1. **Reuse newly introduced functions (example: delete contact) to improve your existing code of Mini-Project**

Managing Contacts

FEATURE-ID-1: View details of a single contact

When the list of contacts is being displayed, user shall be able to select any 1 contact and view all the stored details of the contact

FEATURE-ID-2: Update details of a single contact

After selecting any contact for view (*FEATURE-ID-1*), the user shall be able to update the contact details.

FEATURE-ID-3: Delete the contact

After selecting any contact for view (*FEATURE-ID-1*), the user shall be able to delete the contact

Search History

Design

You will need a separate class that stores the list of searched strings as searched history along with the data and time. You can make a HistoryObject class that can store a string and date/time, then make a class that stores a list of HistoryObject objects (Just like ContactsBook class).

FEATURE-ID-4: View Search History reverse sorted(latest first) by Date/Time, store a Date/Time attribute. It will need some Date/Time class, maybe built in class of C++ language class or you can write your own.

FEATURE-ID-5: View frequently searched top 5 contacts; count the number of times each contact is viewed and show top 5 viewed contacts. Again unique\_id of contact will help here.

Groups of Contacts

Design: A list of groups, and each group can have a list of contact\_id.

FEATURE-ID-6: Create a group with given name

FEATURE-ID-7: Add a contact to group; when this menu item is chosen, show a menu to choose from available groups, after group is chosen, show a menu to choose from contacts to add. Add the selected contact to the group.

FEATURE-ID-8: Remove a contact from a group; While viewing a group, the user can remove a contact from the group.

FEATURE-ID-9: View a contact’s groups; View all the groups a contact is in.

FEATURE-ID-10: Delete a group

Note: When a contact is deleted, or deleted during merge duplicates, its ID should be removed from all the groups as well(groups where contact is added).

Advance Search

FEATURE-ID-11: Display all the contacts that match the given string, string shall be searched in all the suitable attributes of a contact. User will just enter a string without telling your program to search by First Name or Last Name or Phone Number. You will have to search the string in all these attributes including the address.

FEATURE-ID-12: Searching by Name, or any other attribute should not be character by character match, but a character level match like:

Search query entered by user: sal

This contact has a match as the highlighted letters are shown(in order): Usman Ali Similarly, if some digits of a phone number are entered: 397 This can be found in multiple contacts’ phone numbers.

## Menu Changes due to Advanced Search

Search will now be a single item, not like Search By First Name, Search by Last Name. It will be just search now, as you will perform searching of the entered string in all these attributes everytime.

Persistence/Permanent Storage

1. You will reuse the save\_to\_file feature to make sure your contacts/contacts group and all other data is always saved in files, even if the program crashes, **data must not be lost**. It means you will have to save data frequently, maybe every time a change occurs. Use separate files for different data such as Contacts, Groups, SearchHistory, FrequentlySearched
2. When the program starts, all the data shall be automatically loaded. Loading can be done through the constructors and some loading functions in each class.

# **Graphical User Interface (GUI): Windows Forms (15% marks)**

1. Use Windows Forms with C++ instead of console and make this

application GUI app

1. You will do all the logic in separate classes, Windows Form files shall only be used to display data and obtain input from the user and pass the input to logic classes for processing. Also, input validation like empty strings check, or entering numbers in name field, or alphabets in phone number field etc. will be done in Windows Forms class functions (event listeners)
2. Discuss with me your group and details of this. You will have to actively take guidance from me to write clean and non-spaghetti code (spaghetti code contains “juggars and is dirty”). Meet me first!
3. Contacts Cloud Storage using Google Firebase
4. 1.If you go this route, you will have to implement two storage methods: file handling and Firebase both.
5. 2.Once you store data in files, the same data should also be stored in the Firebase database. You can use the method we discussed in the Polymorphism lab using OutputDevices. But you will have to download from Firebase and update local file data if Firebase data is changed from some other computer. As, using cloud storage your program can have the same data on multiple computers.
6. 3.You will need to learn and use some libraries and Google Firebase account, meet and discuss with me if you want to do this.
7. 4.This will be really interesting though!