CS200 Introduction to Programming

Spring 2022

Monday 21 February 2022

Assignment 01

Assignment Guidelines

Rules:

1. You must put all your work (source only) into a folder named Assignment01_YourRollNo (e.g, Assignment01_22100167), compress it into a "zip" file and submit the zip file on LMS before **Thursday 3rd March.**
2. Communicating with each other is NOT permitted. If you have a question, ask the TAs.
3. The object is not simply to get the job done, but to get it done in the way that is asked for in the lab.
4. All submissions are subjected to run in Moss, hence prevent sharing your code with anyone in the class.
5. Any cheating case will be reported to Disciplinary Committee without any delay.

Coding Conventions:

- Constants are "ALLCAPS" or "ALL_CAPS".
- Variables are "allsmall" or "all_small".
- All function names must be "firstWordSmallAllOtherWordsCamelCase".
- All class names must be "CamelCaseWords".
- All curly brackets defining a block must be vertically aligned.
- Use tabs where needed instead of spaces.
- File naming: Lab02_YourRollNo_TAname.zip

Course Learning Objective (CLOs):

1. CLO1: Enabling Knowledge:
    a. (C1) use object-oriented programming model: abstract data types, encapsulation, inheritance, and polymorphism to code algorithmic solutions using standard coding conventions.
    b. (C1) use of fundamental features of an object-oriented language like C++.
2. CLO2: Critical Thinking and Analysis:
    a. (C4) analyze the requirements for solving simple algorithmic problems.
3. CLO3: Problem Solving:
    a. (C6) design algorithm and implement program code in an object-oriented programming language such as C++ to solve simple algorithmic computing problems, based on the analysis of the requirements.
    b. (C5) evaluate the correctness of the proposed solution.
4. CLO4: Communication:
    a. (C2) explain key concepts of algorithmic design in written form.
5. CLO5: Responsibility:
    a. (C3) apply relevant conventions, standards and ethical considerations to writing computer programs.

Task: Database [100 marks]

You will be making a database system in this assignment with Login and SignUp features where you will be creating user accounts and then implementing the authentication for username and passwords.

Consider a class *Database* which has the following members:

- Array of class of *User*
- Number of users
- Default Constructor
- Parametrized Constructor
- Copy Constructor
- ++ Operator
- -- Operator
- Setters
- Getters
- Login()
- SignUp()
- Delete()
- Print()

The array is to be considered of size 20 by default. The number of users will increment whenever a new account is created.

In Login, you will be taking username and password from the user and authenticate it with already existing array. You will be required to search the username first, and then match its password.

In Sign Up, you will be taking input the details of the User class from the terminal. Incrementing the second member “Number of users” using ++ operator overloading.

In Delete, you will be searching for the username that is to be deleted, ask for its password, it both matches, delete the User. You will use – operator overloading for decrementing the “Number of users”.

For deletion, this is what you are supposed to do: when you find the username and password, let’s say user1 and you find it at index 12, you will set the details of the user1 default values and move all values in the array that are on the right side (index>12) to the left by 1. Meaning, the user stored at index 13 would now move to index 12, and so on. If there is no user on the right side of 12, leave it as it is. Inshort, the array should be filled from left to right without gaps.

In Print, you will print each username and password one by one.

The *User* class has the following members:

- Name
- CNIC
- Password
- Default Constructor
- Parametrized Constructor
- Copy Constructor
- Setters
- Getters
- << Operator
- >> Operator

The << and >> Operators should take input and print the details of a User, respectively.

The program should start with a menu having following options:

1) Login
2) Sign Up

3) Delete
4) Exit

After Login, the program should display an error message if:

- If the username and password doesn't match
- If the username doesn't exist

After the error, the program should display the menu again.

After successful Login, user would be given a welcome message and the details of the user printed on screen using << operator overloading. User would be given the following options after displaying the details of the user:

1) Logout
2) Exit Program

After Logout, the program should go back to the Login/Sign up menu. The Exit in all menus terminates the program.

For Sign Up, the program should take input for all details of the User class using >> operator overloading. Then add this instance of user to the array of users in the class *Database*.

For Delete, repeat the above mentioned process.

Marks division:

User Class

- Default, parametrized, and copy constructor 10 marks
- Getters and Setters 05 marks
- << and >> Operator 05 marks

Database Class

- Default, parametrized, and copy constructor 10 marks
- Getters and Setters 05 marks
- Correct declaration and initialization of array 05 marks
- ++ and -- Operator 05 marks
- Login 05 marks
- Sign Up 10 marks
- Delete 10 marks
- Print 05 marks

Main Program

- Correct menu display with error handling (both menus) 05 marks
- Correct program flow (only exit when user asks to exit) 05 marks
- Correct error messages 05 marks
- Proper coding conventions 05 marks
- Comments with proper descriptions 05 marks

You must write comments for each section of code mentioning the logic you used in that part of code.

Note: Do not use inline functions

……

Good Luck!