ESS 201-Programming II

Assignment 5: C++

Submission: LMS and Domjudge on 23th November

Problem:

In this assignment, you are going to extend your model of a Simple Social Media Application, which consists of the following classes: System, Posts, and Users.

Descriptions of the Classes:

- **System**
  - Should maintain a list of all posts
  - Should maintain a list of all users
  - Should maintain the name of the application.
  - Should maintain the current User using the system. (Only one user is allowed at a time.)
  - (This should be initialised in a parameterised constructor.)
  - Getters and Setters should be implemented wherever you see fit.
  - Access Control:
    - There will be a signUpUser(std::string userName, std::string password) method that will add a new User into the system.
    - There will be a signInUser(int userID, std::string password) method that sets the current User variable to the user with the given parameters. All actions pertaining to posting and following/unfollowing are done in the context of this user. If the user does not exist or the password is incorrect, print **Access Denied** (in a new line.)
    - There will be a SignOutUser() method that resets the Current User Variable. After SignOutUser() is called, any user-related functions should fail; and you should print **Access Denied** (in a new line.)
- **User**
  - Should maintain a unique User ID (statically generated by the *User* class.) - The values will be 1,2,3,...
  - Should maintain a name
    - usernames will not contain spaces.
  - Should maintain a password
    - passwords will not contain spaces.
  - Should maintain a list of users they follow (List of *User* objects)
  - Should maintain a list of their posts (List of *Post* objects)
  - Should have methods to create Posts and follow/unfollow another user
  - Getters and Setters should be implemented wherever you see fit.
  - Overload the std::cout function such that std::cout called on an object of this class returns a string of the following format (Example:)
    - **User Id: 1, User Name: user1, Number of Posts: 3, Following Count: 2**
- **Post**
  - Should maintain a unique Post ID (statically generated by the *Post* class.) - The values will be 1,2,3,...
  - Each post will have a unique User who posts - the class will maintain a User ID Attribute.
  - Should maintain content
    - You may use std::string for this
  - Should maintain a year of posting
  - Getters and Setters should be implemented wherever you see fit.
  - Initialise posts with the help of parameterised constructors
  - Overload the std::cout function such that std::cout called on an object of this class returns a string of the following format (Example:)
    - **Post Id: 1, User Id: 1, Year: 2021, Post Content: this is a sample post**
- **Input/Output:**
  - All inputs are taken through std::cin and all outputs involve std::cout
  - Each line of the input will contain a keyword(in caps) followed by arguments.

        - CREATE nameOfSystem
        - SIGNUP usernameOfNewUserCreated passwordOfNewUserCreated
        - SIGNIN userIdOfUserSigningIn passwordOfUserSigningIn
        - SIGNOUT
        - UNFOLLOW usernameOfTheOneBeingUnfollowed
        - FOLLOW usernameOfTheOneBeingFollowed
        - POST YearOfPost ContentOfPost(can contain spaces)
        - PRINTUSER userIdOfUserObjectWhereOverloadedCoutIsBeingCalled
        - PRINTPOST postIdOfPostObjectWhereOverloadedCoutIsBeingCalled
        - EXIT
    - EXIT specifies the end of the input. You must call all destructors here.
    - You must first print the name of the System (i.e the parameter in CREATE)
    - SIGNUP calls the SignUpUser method of the system class - this essentially adds a new user to the System.
    - SIGNIN calls the SignInUser method of the system class - this sets the currentUser object of the System class and all future operations (i.e Post, Follow/Unfollow) are done by this user, until SIGNOUT occurs.
    - SIGNOUT resets the currentUser object of the System class.
    - PRINTUSER calls the overloaded std::cout function on the User with the ID given in the parameter of this call. Note that the working of this function is agnostic of which user is Signed in to the system.
    - PRINTPOST calls the overloaded std::cout function on the Post with the post ID given in the parameter of this call. Note that the working of this function is agnostic of which user is Signed in to the system.

- **Error Codes to be Printed:**

Note: In our implementation, SIGNOUT will never create any errors. Even if we call SIGNOUT where no user is Signed in, we just reset the current User object in System class and continue.

Input String	Condition	What to Print
SIGNUP	Username provided already exists for another user	**Access Denied**
SIGNIN	Username provided exists for a user in the system, but the password provided in the call does not match with the password of the user	**Access Denied**
SIGNIN	Username provided does not belong to any user in the System	**Access Denied**
UNFOLLOW	The username provided is not a username of any of the users in the current User's following List	**Invalid Input**
FOLLOW	The username provided is not the username of any user in the System	**Invalid Input**
POST	No valid user is Signed in (i.e Current User of System class is not set/default)	**Access Denied**
UNFOLLOW	No valid user is Signed in (i.e Current User of System class is not set/default)	**Access Denied**
FOLLOW	No valid user is Signed in (i.e Current User of System class is not set/default)	**Access Denied**
POST	Year of Post provided is <= 0	**Invalid Input**
PRINTUSER	The user ID provided does not belong to any user in the system	**Invalid Input**

PRINTPOST	The post ID provided does not belong to any post in the system	**Invalid Input**

- **Sample Input**

```
CREATE sma
SIGNUP user1 password1
SIGNUP user2 password2
SIGNIN user1 password3
SIGNOUT
SIGNIN user1 password1
FOLLOW user2
POST 2021 this is the first post
SIGNOUT
SIGNUP user3 password3
SIGNIN user2 password2
FOLLOW user1
UNFOLLOW user1
UNFOLLOW user1
POST 2020 this is the second post
SIGNOUT
PRINTUSER 1
PRINTUSER 2
PRINTUSER 3
PRINTUSER 4
SIGNUP user4 password4
PRINTUSER 4
SIGNOUT
POST 2022 this is the third post
PRINTPOST 1
PRINTPOST 2
PRINTPOST 3
EXIT
```

- **Sample Output**

```
Access Denied
Invalid Input
User Id: 1, User Name: user1, Number of Posts: 1, Following Count: 1
User Id: 2, User Name: user2, Number of Posts: 1, Following Count: 0
User Id: 3, User Name: user3, Number of Posts: 0, Following Count: 0
Invalid Input
User Id: 4, User Name: user4, Number of Posts: 0, Following Count: 0
Access Denied
Post Id: 1, User Id: 1, Year: 2021, Post Content: this is the first post
Post Id: 2, User Id: 2, Year: 2020, Post Content: this is the second post
Invalid Input
```