2023, Spring CS-2 Advanced C++ Programming
Chapter #17: Social media Posts Final Exam Project

Bill Komanetsky
Las Positas College

May 12, 2023

Any printed or copied version of this document are for reference only and should not be considered the most up to date version of this document. Please see the version of this document located on the Canvas class management system for the most recent version

---

1 Project Overview

So, you have been given the task of writing a program that let's you query the social media posts of social media users. The data for this program will be stored in STL Map's and MultiMap's after being read from CSV files.

To do this, there will be two files:

- **SMUsers.csv**: This file will contain users of the social media company where you are working.
- **SMUserPosts.csv**: This file will contain posts from the users identified in the previous, SMUsers.csv file

Total Points: 100

1.1 Requirement #1

Create a class called **SMUser** (with it's own .cpp and .hpp files):

- User ID
- User First Name
- User Last Name
- Date of Membership (usr your previously written date class)

Create a class called **SMUserPost** (with it's own .cpp and .hpp files):

- User ID
- Date of the post (use your previously written date class)
- Time of the post. This should be a structure named **PostTime** defined in your SMUserPost class:
  - Hour (an integer using 24-hour format)
  - Minute (an integer)
  - Second (an integer)
- The post's text

All attributes must be private and all accessors and mutators for these attributes made public. Data validation in the mutators must throw a C-String error and include:

- Strings cannot be blank
- Numbers cannot be zero
- Dates must be valid

1.2 Requirement #2

In your main(), create a menu as shown below:

1. Display all Users
2. Display a Specific User
3. Display all Posts
4. Display Posts from a Specific User
5. Exit the program

1.3 Requirement #3

When your main() function is invoked:

- Read the SMUsers.csv file into a Map of SMUser **pointers** with the key for that map being the User ID in the SMUser class. If the file doesn't exist, display an error message and end the program. If the user already exists in the map, save that user's details in a simple text file named **UserErrors.txt** with some error text so the development team can fix those posts later and DO NOT add it to the Map (skip it).
- Read the SMUserPosts.csv file into a Multi-Map of SMUserPost **pointers**. If the file doesn't exist, display an error message and end the program. Before adding a post to this Multi-Map, make sure that the user the post belongs to exists in the SMUser Map. If it does not, write that post to a simple output file named **PostErrors.txt** with some error text so the development team can fix those posts later and DO NOT add it to the Multi-Map (skip it).

1.4 Requirement #4

When the user selects option #1, in a range-based for loop, print the user ID, User Name (both), Date of membership (using your Date classes operator<< overload function) and total number of posts which the user has (from the Multi-Map) in a nicely formatted report (using **iomanip**) and go back to the main menu.

1.5 Requirement #5

When the user select option #2:

1. Ask the user to type an SMUser ID. If that ID does not exist in the SMUser map, display an error message and go back to the menu
2. If the user does exist in the SMUser map, print the same information for that user as you did for all users when option #1 was selected from the menu
3. Go back to the main menu

1.6 Requirement #6

When the user selects option #3, in a range-based for loop, print the user ID, date of (using your Date's overloaded operator<< function), time of and the text of the post in a nicely formatted report (using **iomanip**).

1.7 Requirement #7

When the user selects option #4:

1. Ask the user to type an SMUser ID. If that ID does not exist in the SMUser map, display an error message and go back to the menu
2. If the user does exist in the SMUser Map, print the Userid and name of the user, the total number of posts the user has in the SMUserPost Multi-Map
3. Next, using the **equal_range** function, print all of the posts from the multi-map for the user ID input by the user under the user details in a nicely formatted report (using **iomanip**). If there are no posts for that user, just print "No Posts Found" for that user
4. Go back to the menu

1.8 Requirement #8

When the user selects option #5:

- Iterate through the Multi-Map using a non-range based for loop and delete each SMUserPost pointer
- Iterate through the Map using a non-range based for loop and delete each SMUser pointer
- Clear both the Map and the Multi-Map
- Open the Posterrors.txt file and display the errors to the user to remind them that there are problems with the csv file.
- Open the UserErrors.txt file and display the errors to the user to remind them that there are problems with the csv file.
- Print a nice farewell message that the program is ending

---

2 Data to be used

All data will be provided by files attached to this project.

Description of the SMUser.csv data file:

- User ID (a String)

- User First Name (a String)
- User Last Name (a String)
- Date of Membership (year (integer), month (integer), day (integer))

Description of the SMUserPosts.csv data file:

- User ID (a String)
- Date of the post (year (integer), month (integer), day (integer))
- Time of the post.
  - Hour (an integer using 24-hour format)
  - Minute (an integer)
  - Second (an integer)
- The post's text (a String)

```
User01, Smith, Jane, 2021, 08, 07
User02, Taylor, John, 2020, 12, 31
User02, Thompson, Carol, 2020, 4, 15
User03, Fillingame, Susan, 2019, 03, 30
User04, Samuels, Steven, 2020, 06, 27
```

Figure 1: Example SMusers.csv file - just an example

```
User01, 2021, 08, 07, 08, 14, 00, I think that racism in this country sucks!
User99, 2022, 12, 14, 13, 31, 58, I agree that we need to have higher taxes on the rich!
User02, 2020, 12, 23, 08, 00, 00, Happy Holidays everyone!
User02, 2021, 01, 01, 08, 00, 00, Happy New Year everyone!
User03, 2021, 01, 01, 08, 10, 00, I don't believe in celebrating New Year's
User02, 2021, 01, 02, 08, 15, 20, Ahhhhh don't be that way
User08, 2021, 01, 03, 10, 30, 14, Hey everyone I just got a raise!
User01, 2021, 01, 04, 08, 15, 23, Happy new Year to you too John!
User01, 2020, 12, 24, 13, 22, 47, Happy Holidays to you too John!
User03, 2022, 01, 05, 12, 01, 38, I think I'll go play soccer today with my friends
User01, 2022, 01, 05, 12, 07, 19, Let me know where that will be and I'll join you!
```

Figure 2: Example SMuserPosts.csv file - just an example

---

3 Sample output

- None - be creative!
- Make sure your report output is easy to read

---

4 Hints and Tips

- If you did what we did in class, you should be able to do this project with no problems. If you did not follow along, I wish you the best of luck.

- Remember, this is a final exam project. If you do not turn it in, you will fail the class. If you cannot finish it completely, make sure it compiles and make sure it does some of what you are required to do so you can get partial credit.

---

5 Submitting your project

- Export your Eclipse project as shown in class and on videos found in Canvas
- In Canvas, upload the archive .zip file when submitting the Canvas assignment
- After uploading to Canvas, it is suggested that you then download what you uploaded to assure it is correct

---

6 Grading

Table 1: Grading Criteria

Item	**Percentage Off**
Program does not compile or link	-100%
Some feature documented in this project description is not implemented into the code	-20%
Memory leaks found	-30%
Use of smart-pointers	-100%
Poorly formatted reports	-20%
Plagiarized code	-100%
Computer-Generated code of any kind	-100%
Late	-100%

---

7 Notes

- None

---

8 Help

- This is a final exam project and not a group project, so no discussion board will be provided to you