CMPSC121: Intro to Programming Techniques (Fall 2018)

Project 3 (100 points)

Due Friday, December 7 at 11:59pm

Objectives

After this project, students should be able to:

- Implement user-defined classes using given specifications, including
  - Member variables
  - Member functions
  - Constructors
- Use file input and output to save results

Background

While working to improve the LionPath website, you discovered a way to generate formatted text files that contains information about students' schedules for the upcoming semester. Consider the example below:

```
FNAME: Andrew
MINIT: S
LNAME: Yu
ID:    912345678
USER:  auy77

CMPSC121-002-003L
- M  9 05 AM  50 Lecture
- W  9 05 AM  50 Lecture
- T 10 05 AM 110 Lab
MATH140-001
- M 12 25 PM  50 Lecture
- T 12 25 PM  50 Lecture
- W 12 25 PM  50 Lecture
- R 12 25 PM  50 Lecture
PSU007-001
- W 06 00 PM  50 Lecture
ENGL101-004
- M 11 15 AM  50 Lecture
- W 11 15 AM  50 Lecture
- F 11 15 AM  50 Lecture
PHYS211-003-006L
- T  3 35 PM  90 Lecture
- R  3 35 PM  90 Lecture
- F 12 00 PM 110 Lab
```

Each line that begins with "-" refers to one of the meeting times of the course above. For example:

`- M  9 05 AM  50 Lecture`	Monday, 9:05AM, 50 minute lecture

Because LionPath can generate many of these files, you have decided to write a C++ program that can automatically process them. You will use object-oriented programming to keep track of the students, the courses, and the lecture times.

Instructions

Implement the classes described below:

Student

The `Student` class contains basic identifying information about one student:

- First name
- Middle initial
- Last name
- 9-digit PSU ID Number
- PSU Username

In addition, it contains a list of courses that the student is scheduled to take. Certain actions related to the student can be taken:

- Add a course
- Remove a course
- Print information about all courses (to console window and/or file)
- Check whether the student will be available (not in class) at a certain time and day of the week
- Check whether the student is available right now

Course

The `Course` class contains minimal information:

- Name of the course

It also contains a list of meeting times (labs and lectures). There is only one action related to the course:

- Print information about all meeting times (to console window and/or file)

Lectures

The `Lecture` class contains all the information about one meeting time (lab or lecture) for a course, of which there may be many:

- Day of the week
- Starting time
- Duration
- Whether it is a lab or lecture session

There is only one action related to the course:

- Print information about the meeting time (to console window and/or file)

You are required to implement but are NOT limited to the member variables and functions described above. You may find it beneficial to implement additional classes, member functions, and member variables that suit your needs.

Main Function

In the main function, provide the user with 6 options:

1. Select student by ID
    - The user is prompted to enter a PSU ID Number
        - If a corresponding file (e.g. "912345678.txt") exists, then open it
            - Create a new student object containing all the provided information
            - If there are pre-existing scheduling conflicts, then remove one of the courses
        - If not, then allow the user to create a new student object with that number
            - Prompt the user for the student's name and username
    - Options 2-5 should not be available until the user selects a student
2. Modify schedule
    - Delete an existing course (provide a list of course names to choose from)
    - Add a new course
        - Use user input to populate the course name and meeting times
        - If there is a scheduling conflict with an existing course:
            - Display the course and meeting time with which it conflicts
            - Do not add the new course
3. Check student availability
    - Determine whether the student should be in class right now
        - Use functions defined in the `<ctime>` library to get the current time
    - Determine whether the student will be in class at a certain time and day of the week
    - If the student is/will be in class, then display the course and meeting time
4. Display Schedule
    - Generate and print all the information in the same format as the given text file
5. Write to file
    - Generate and save all the information in the same format as the given text file
    - Overwrite the existing file (if it exists)
6. Quit
    - If the student's schedule has been modified but not written to file, give the user a warning
        - The user may quit anyways, losing any unsaved changes
        - The user may return to the menu

Sample Output

```
Welcome to the scheduling program!

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
1
Enter the student's 9-digit PSU ID: 912340000
"912340000.txt" does not exist. Would you like to create a new student (Y/N)? n

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
```

```
4. Display schedule
5. Save to file
6. Quit
4
No student has been selected!

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
1
Enter the student's 9-digit PSU ID: 912345678
---SCHEDULING CONFLICT---
EXISTING CLASS: ENGL101-004
- F 11 15 AM  50 Lecture
NEW CLASS: PHYS211-003 (NOT ADDED / REMOVED)
- F 12 00 PM 110 Lab

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
4
FNAME: Andrew
MINIT: S
LNAME: Yu
ID:    912345678
USER:  auy77

CMPSC121-002-003L
- M  9 05 AM  50 Lecture
- W  9 05 AM  50 Lecture
- T 10 05 AM 110 Lab
MATH140-001
- M 12 25 PM  50 Lecture
- T 12 25 PM  50 Lecture
- W 12 25 PM  50 Lecture
- R 12 25 PM  50 Lecture
PSU007-001
- W 18 00 PM  50 Lecture
ENGL101-004
- M 11 15 AM  50 Lecture
- W 11 15 AM  50 Lecture
- F 11 15 AM  50 Lecture

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
2
Add a new course or delete existing (A/D)? D
Select a course to remove:
1. CMPSC121-002-003L
2. MATH140-001
```

```
3. PSU007-001
4. ENGL101-004
4

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
4

CMPSC121-002-003L
- M  9 05 AM  50 Lecture
- W  9 05 AM  50 Lecture
- T 10 05 AM 110 Lab
MATH140-001
- M 12 25 PM  50 Lecture
- T 12 25 PM  50 Lecture
- W 12 25 PM  50 Lecture
- R 12 25 PM  50 Lecture
PSU007-001
- W 18 00 PM  50 Lecture

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
6
You have unsaved changes. Quit anyways (Y/N)? N

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
3
Check current day and time (Y/N)? Y
Student is available at: Monday 6:41 AM

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
3
Check current day and time (Y/N)? N
Enter day of the week(U, M, T, W, R, F, S): M
Enter hour: 9
Enter minute: 30
Enter AM or PM: am
---NOT AVAILABLE---
CLASS TIME: CMPSC121-002-003L
- M  9 05 AM  50 Lecture
REQUESTED TIME: Monday 9:30 AM
```

```
Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
5
Changes written to "912345678.txt"

Choose an option:
1. Select student
2. Modify schedule
3. Check student availability
4. Display schedule
5. Save to file
6. Quit
6
Thanks for using the scheduling program!
```

Submission

Submit the following file(s) to Canvas before the deadline:

1. `scheduler.cpp`