CS215: Introduction to Program Design, Abstraction and Problem Solving
(Spring, 2025)
Lab Assignment 8
(20 points)
Today's Date: Monday, March 24
Demonstration Due Date: the end of Lab9 class
Submission Due Date: Friday, April 4

The purpose of this lab assignment is

- to understand the concept of class, object, encapsulation, OOP
- to practice designing your own class

Problem Statement

The following class **Clock** implements some basic operations on time. The following shows the declaration of the class **Clock**. In this lab assignment, you need to complete the class definition and write the main function.

```
class Clock  {
   public:
     Clock();                              // default constructor
     Clock(int hh, int mm, int ss);   // alternative constructor

     // set time to hh:mm:ss
     void setClock(int hh, int mm, int ss);

     //increase time by sec seconds
     void incrementSeconds(int sec);

     //increase time by min minutes
     void incrementMinutes(int min);

     //increase time by hh hours,
     //if hours reach 24, simply wrap around to 0.
     void incrementHours(int hh);

     void addTime(Clock C);     //add C into the current clock time

     //print time in hours:minutes:seconds in 24-hour format
     void printTime() const;

     //compare with C, if it is earlier than C, return -1;
     // if it is the same time as C, return 0;
     // if it is later than C, return 1.
     int compareTime(Clock C) const;

   private:
     // Declare data members of the class
```

```
        // 0 <= hours < 24, 0 <= minutes < 60, 0 <= seconds < 60
        int hours, minutes, seconds;

        // Helper function to validate the data members
        // it only serves the member function of this class, hence private
        int hours, minutes, seconds;

};
```

You can either download (you need to save the file under the same folder as your solution file, and add it into the solution) or copy the content of the header file, which contains the declaration of the class **`Clock`**, from the following link:
https://www.cs.uky.edu/~yipike/CS215/Clock.h

Note that for this Lab assignment, you need to create your own header file, which contains the declaration of the class **`Clock`**, and its implementation .cpp file, which completes the definition of class **`Clock`**, and then your main function source file. After you create a new project from Visual Studio IDE, right click on **Source Files** and select **Add → New Item...,** and choose "C++ File (.cpp)" for the implementation file of the class **`Clock`**, or click on **Header Files** and select **Add → New Item...,** and choose "Header File (.h)" for the declaration of the class **`Clock`**.

Use the following instructions to write the main function:

```
int main()
{
      //write each statement for each operation below in ()
      (create Clock object C1)
      (set C1 with h:m:s = 3:-5:16)
      (set C1 with h:m:s = 0:0:5)
      (create second Clock object C2 with h:m:s=12:35:59)
      (print C1)
      (print C2)

      //compare C1 with C2
      if (C1.compareTime(C2) < 0)
            cout << "C1 is earlier than C2" << endl;
      else if (C1.compareTime(C2) > 0)
                  cout << "C1 is later than C2" << endl;
            else
                  cout << "C1 is the same as C2" << endl;

      (add C2 into C1)
      (print C1)
      (print C2)
      (compare C1 with C2)
      (increase clock C1 by 55 seconds)
      (print C1)
      (increase clock C1 by 119 minutes)
```

```
      (print C1)
      (increase clock C1 by 22 hours)
      (print C1)
      (print C2)

      (compare C2 with C1)

     return 0;
}
```

Here is the sample output of running your program:

```
Invalid time!
Clock C1 -- 00 : 00 : 05
Clock C2 -- 12 : 35 : 59
C1 is earlier than C2
Clock C1 -- 12 : 36 : 04
Clock C2 -- 12 : 35 : 59
C1 is later than C2
Clock C1 -- 12 : 36 : 59
Clock C1 -- 14 : 35 : 59
Clock C1 -- 12 : 35 : 59
Clock C2 -- 12 : 35 : 59
C2 is the same as C1
Press any key to close this window...
```

(Note: for this Lab assignment, we use 24-hour clock (or you can call it military time). The complete definition can be found at (https://en.wikipedia.org/wiki/24-hour clock).
Hence, for the data members of the clock class, they are in the following ranges respectively: 0 <= hours < 24, 0 <= minutes < 60, 0 <= seconds < 60. if the hours of the clock reach 24, simply wrap around to 0.
For the printTime function, a leading zero is added for the numbers under 10. For example, if the time of the clock is 7:5:9, it should display 07:05:09)

Demonstration and Submission

1. Each Lab assignment needs to demonstrate to your TA to be graded. You can demonstrate Lab8 during Lab8 class (with possible bonus 3 points) or no later than the end of Lab9 class (this is **the demonstration deadline** for Lab8).

If you finish Lab8 assignment during Lab8 class, you may demonstrate your program to your TA and answer your TA's questions, you can get up to 3 extra points for this lab assignment. (Note you can also demonstrate your program to your TA during Lab9 class. However, any demonstration later than the end of the Lab8 class cannot get bonus 3 points.)

If you need extra time, you can continue working on Lab8 assignment after the Lab class, and try to finish it before the next Lab class. Then demonstrate your Lab8 during Lab9 class.

If you do not demonstrate your code, even if you submit it in Canvas, you will receive a grade of 0!! *The TA may ask you to make some corrections. If so, make the corrections and demonstrate again…repeat until you have 100%!*

2. After the successful demonstration, submit the code in Canvas. Open the link to Course Canvas page (https://www.uky.edu/canvas), and log in to your account using your LinkBlue ID and password. **Please submit TWO source files, named Clock.cpp and Lab8.cpp through the link "Lab 8".**
Even if you successfully demonstrated it to the TA, if you do not submit in Canvas by* the submission deadline*, you will receive a grade of 0!

Grading (20 points + Bonus 3 points)

1. Attend the lab session or have a documented excused absence. (5 points)
2. Your program correctly solves the problem.
   - Include comments as specified in the lecture notes. (1 point)
   - Class definition is correct. (9 points)
     - ✓ each member function is correct. (1 point * 7)
     - ✓ the data member in the valid range (1 point)
     - ✓ add the leading 0 when the digit is less than 10 (1 point)
   - main function follows instruction and correctly generate the output (3 points)
   - Separate source file and header file for the class definition. (2 points)

Demonstrate your program to your TA and answer TA's questions during Lab class when the same Lab assignment is given. (Bonus 3 points)