Object Oriented Programming Lab

Lab 11

Special Section

Instructions:

- Work on this lab individually. Discussion is not allowed.
- Anyone caught being indulged in the act of plagiarism would be awarded an "F" grade in this lab.

Submission Details:

- Submit the zip folder of your lab source files via mail at oop.lab.specialsection@gmail.com
- On August 20, 2020, Before 09.59pm
- The email subject should be OOP Lab 11_Roll No.
- Late submissions are not allowed.

Note: Validate all the inputs in their correct format. Use **this pointer** where it can be used.

Task 01 **20 Marks**

Write a class named **Time** that stores the time in **24 hours** format; having following functionalities

- The class should have following **three private data members**.
  1. An **integer** named **second** that holds the value of the **seconds.**
  2. An **integer** named **minute** that holds the amount of the **minutes**.
  3. An **integer** named **hour** that holds the value of the **hours.**

Value should only be assigned to **second**, if it is in between **0 (default value)** and **59** both inclusive.
Value should only be assigned to **minute**, if it is in between **0 (default value)** and **59** both inclusive.
Value should only be assigned to **hour**, if it is in between **0 (default value)** and **23** both inclusive.

- Provide the implementation of **mutators** for all the data members (sec, minute and hour) of the class.
- Provide the implementation of **accessors** for all the data members (sec, minute and hour) of the class.
- Provide the implementation of following **constructors** and a **destructor**

  1. The constructor should accept the **Time's second**, **minute** and **hour** as arguments. These values should be assigned to the object's appropriate member variables.
  2. The constructor should accept the **Time's minute** and **hour** as arguments. These values should be assigned to the object's appropriate member variables. The **second** data member should be assigned to the **default value**.
  3. A **default constructor** that initializes all the data members of the class with **default values**.
- Provide the implementation of following **overloaded operators**

1. stream insertion (<<) to display the time in the form **16:50:45 (hour:minutes:seconds)**
2. stream extraction (>>) should prompt the user for a time to be stored in a **Time** object. The operator should ask the user to enter the time in the following format; **hour:minutes:seconds**
3. pre-increment (++) should increment the **second** data member of the object
4. post-increment (++) should increment the **second** data member of the object
5. pre-decrement (--) should decrement the **second** data member of the object
6. post-decrement (--) should decrement the **second** data member of the object
7. subtraction (-) binary should subtract the one time from another and **return the number of seconds** between two times. For example, if **16:50:45** is **subtracted** from **16:51:00**, the result will be **15**.

Tazeem Haider
(tazeem.haider@pucit.edu.pk)

8. addition (+) unary should return **true**, if the time is a **working hour** i.e. 09:00:00 to 17:00:00, **false** otherwise.

- The class should detect the following **conditions** and handle them accordingly:
  1. When a **time** is set to the **last sec of the minute and incremented**, it should **become the first sec** of the following **minute**.
  2. When a **time** is set to **59:59 (min:sec) and incremented**, it should become **00:00 (min:sec)** of the following **hour**.
  3. When a **sec** is set to the **first sec of the minute and decremented**, it should become the **last sec** of the **previous minute**.
  4. When a **time** is set to **00:00 (min:sec) and decremented**, it should become **59:59 (min:sec)** of the **previous hour**.
- Once you have written the class, write **main** function and test its functionality by creating some objects of **Time**.