University of Central Punjab

Faculty of Information Technology

Object Oriented Programming

Lab 07	
Topic	Classes in C++
Objective	The basic purpose of this lab is to revise some preliminary concepts of C++ that have been covered in the course Introduction to Computing and Programming Fundamentals. Its objective is to recall previously learned basic concepts like a revision of arrays, functions, and pointers.

Instructions:

- Indent your code.
- Comment your code.
- Use meaningful variable names.
- Plan your code carefully on a piece of paper before you implement it.
- Name of the program should be the same as the task name. i.e. the first program should be Task_1.cpp

Students are required to complete the following tasks in lab timings.

Task 1:

Let's create a class `Time` to work with a 12-hour formatted time having the following attributes:

```
int* hours
int* minutes
int* seconds
```

Provide the following functions:

1. According to the rule of three:
   i. Copy Constructor
   ii. `operator =`
   iii. Destructor
2. Overloaded constructor with default values
3. Getter of each attribute (setters are NOT required)
4. `operator ==`
5. `operator >`
6. As discussed in class, only using `operator ==` and `operator >`, provide implementation of:
   i. `operator !=`
   ii. `operator <`
   iii. `operator <=`
   iv. `operator >=`
7. `operator <<` `//Example Output` **`2:13:47`**
8. `operator >>`
9. `operator +` `//to add two values of time`
10. `operator -` `//to subtract two values of time`
11. `operator ++`
    i. post-increment
    ii. pre-increment
12. `operator --`
    i. post-increment
    ii. pre-increment

Make sure to check that the time remains valid in all the functions that are:

A. `1 <= hours <=12`
B. `0 <= minutes < 60`
C. `0 <= seconds < 60`

Task 2:

Repeat task 1, to create a class for `Date;` however, no need to take pointer attributes; the attributes will therefore be:

```
int day;
int month;
int year;
```

1. Rule of three is NOT required (we don't have any pointer attribute). Rule of three is required only if there is at least one pointer attribute.
2. Overloaded constructor with default values
3. Getter of each attribute (setters are NOT required)
4. `operator ==`
5. `operator >`
6. As discussed in class, only using `operator ==` and `operator >`, provide implementation of:
   i. `operator !=`
   ii. `operator <`
   iii. `operator <=`
   iv. `operator >=`
7. `operator <<` `//Example Output: 30-4-2013`
8. `operator >>`
9. `operator +` `//to add integer days in a Date`
   Hint: The porotype should be:
   ```
   Date operator + (int d)     //where d is days
   ```
   Example: **20-3-1985** + **16** days will return date as: **5-4-1985**
10. `operator -` `//to subtract two values of time`
11. Hint: The porotype should be:
    ```
    Date operator - (int d)     //where d is days
    ```
    Example: **20-3-1985** - **155** days will return date as: **16-10-1984**
12. `operator ++`
    i. post-increment
    ii. pre-increment
13. `operator --`
    i. post-increment
    ii. pre-increment

Make sure to check that the date remains valid in all the functions that are:

A. `1 <= day <= 31`
B. `0 <= month <= 12`
C. `0 <= year <= n` `where n is a positive integer`