Lab 17

In this lab, we learn to define overloaded functions and operators in a C++ Class. In particular, we will add the following to the "dateClass" defined in closed lab 16:

- Copy constructor. In lab 16, we have already defined two constructors for the dateClass, namely the default constructor and the value contructor. The copy constructor is used where a new object of this class is created as a copy of another existing object. The existing object is passed in the copy constructor as a value parameter, and
- 4 overloaded operators:
    - The == operator is used when comparing two dateClass objects: the dateClass object "self" to the left of the == operator, and the dateClass object to the right of the operator. It returns true if the "self" object represents the same date as the second object, and returns false otherwise.
    - The < operator is used when comparing two dateClass objects: the dateClass object "self" to the left of the < operator and the dateClass object to the right of the operator. It returns true if the "self" object represents the date prior to that of the second object, and returns false otherwise. A date is prior to another date if :
        - the first date's year is prior to that of the second date
        - the two dates have the same year value, and the first date's month is prior to that of the second date; or
        - the two dates have the same year and month values, and the first date's day is prior to that of the second date;
    - The >> operator is used to prompt the user to enter a date. The overloaded operator will prompt to read the month, day, and year values for the dateClass on the right side of the operator;
    - The << operator is used to display the date in the format "Current Date: Month/day/year".

You will need to modify both the specification file and the implementation file of the class "dateClass" to incorporate the declaration and definition of the new constructor and the overloaded operators.

Next, modify the client program ("main.cpp") to test out these new additions: (Note: The steps in black color were coded in lab 16. The steps in blue color are new steps to be added)

1. Declare an object named "firstDate" using the default constructor,
2. Declare a second object named "secondDate" using the value constructor, with month equals to 3, day equals to 15, year equals to 2019,
3. Change the date of the first object to month=7, day=17, year=2019,
4. Use the overloaded << operator to display the "first" dateClass object,
5. Write a C++ if-else statement to compare the first and the second dateClass object using the overloaded == operator. If the two objects do not represent the same date, output the message "Two different dates.", otherwise, output the message "Same date".
6. Declare the third dateClass object as a copy of the first dateClass object using the copy constructor,

7. Use the overloaded >> operator to read the values for the "first" dateClass object,
8. Write a C++ if-else statement to compare the second and the third dateClass objects using the overloaded < operator. If the second object represents the date prior to that of the third dateClass object, output the message: "Second date is prior to the third date", otherwise, output the message: "Second date is not prior to the third date".