OOP Lab 6
Topic: Operator Overloading
Marks: (25)
Release Date: 05-Nov-20 Thursday
Submission Date & Time: 07-Nov-20 11:59 PM Saturday

You may send your lab queries at: asktoknow42@gmail.com

Write a C++ program, to create a class called as Date having day, month and year represented by integers and default values as 1,1 and 1900 respectively. Write an input function for date by overloading the extraction operator >>. Input is given in the format of dd/mm/yyyy. Do appropriate validations for the input.
For validations, you may assume
A valid year is between 1900 – 2100
A valid month is between 1-12
A valid day can be between 1-31 (according to the respective month)
Make following `daysInMonth` array as class's private data member to know the number of days in each month. (No need to handle leap year)

```
static const int daysInMonth[ 13 ] = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }
```

Also overload the operators >=, -, +, ++ (pre increment) ++(post increment), --(pre decrement) and --(post decrement). Display the results by overloading the insertion operator << after each operation. Following calls should be possible.
Make appropriate constructor and member functions to support following calls.

1) cin>>d1 where d1 is a **Date** object. Input is given in the format dd/mm/yyyy and also validated (3 pts)
2) **cout<<d1;** where d1 is a **Date** object. It outputs the date in the format dd/mm/yyyy. (3 pts)
3) **bool isGreaterOrEqual = d1>=d2**; where d1 & d2 are Date objects and isGreaterOrEqual is a Boolean value. (3 pts)
4) **int no_of_days = d1 - d2;** where d1 & d2 are **Date** objects. Assume d1> = d2 and no_of_days is an integer. (4 pts)
5) **Date d2 = d1 + no_of_days;** where d1 and d2 are **Date** objects and no-of-days is an integer. **Also make it commutative. i.e. d2 = no_of_days + d1** (4 pts)
6) **d1++** and **d1--** where d1 is a **Date** object. Increments and decrements date by 1 day respectively. (2+2 pts)
7) **++d1** and **--d1** where d1 is a **Date** object. Increments and decrements date by 1 day respectively. (2+2 pts)

The operators that allow cascading calls for primitive data types should also allow cascading calls for your Date class. For example cin>>d1>>d2; where d1 and d2 are Date objects and input is being taken through cascading calls for objects d1 and d2

Note: No need to create a menu for this program. Simply call each of your functions in order, in the main program.

STEPS TO DETERMINE LEAP YEAR

To determine whether a year is a leap year, follow these steps:

1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
4. The year is a leap year (it has 366 days).
5. The year is not a leap year (it has 365 days).

Write a **private** member function in your date class, that receives an integer value of year as parameter and returns a true boolean value if it is a leap year satisfying the above steps and false otherwise.