CSCI-15 Lab #5 due 10/23/18

Please work in groups of two on this. Do not work alone.

Write a class Date to implement a date on the modern Gregorian calendar, current era only (C.E. or A.D.), with three private member integers, year, month and day. Include:

- a default constructor (set date to Jan 1, 1 or 1/1/1, the default date)
- a normal constructor that takes year, month and day, verifying all are >= 1, the month is <= 12 and the day is <= the number of days in that month (accounting for February in a leap year), otherwise setting to the default date
- a SetAll method taking integer year, month and day, verifying all three are reasonable (same rules as above), otherwise setting to the default date
- GetYear, GetMonth and GetDay methods to return the appropriate values
- separate methods to print the date in each of these forms, given an output stream to print to:
    - 12/25/09
    - December 25, 2009
    - 25 December 2009

Demonstrate the class is correct by writing a client program to test all the features of the class.

Remember the number of days in each month from January: {31, 28 or 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}. February has 29 days in leap years only, otherwise 28. A leap year is one which is divisible by 4, unless it is divisible by 100, unless it is divisible by 400. For example, 1996 was a leap year, 1900 was NOT a leap year, but 2000 was a leap year.

After you have this working, write a method Difference that takes a reference to another Date and returns the number of days between the two Dates.

Do this as follows: use a table (or 2 tables) of the days of the months (you will need to account for February of a leap year). If the dates are not in the same month and year, get the number of days up to the end of this month, then loop through the table adding the number of days in each month until the end of the first year, then loop through the years adding either 365 or 366 as appropriate until you reach the year of the parameter, then loop through the parameter's months until you reach its month, then add the appropriate number of days. If the parameter's date is before the current date, then make copies of the dates, reverse them, calculate normally, and negate the resulting number of days. Remember to code these for efficiency, so don't reload the tables on each method call.

<u>There are formulae to directly calculate the number of days between two dates, but you MAY NOT use one of them.</u>

Test with several dates, including a test from January 1, 2000 to the date you test the program, and tests with dates a few days apart and a few hundred years apart.