Assignment 1- Date class

Necessary skills: *Basic class design, constructors and member functions, basic iostream output formatting*

Description

In this assignment, you'll create a C++ `Date` class that stores a calendar date.. You'll test it using the supplied test `main()` function (attached below).

In your class, use three private integer data member variables to represent the date (month, day, and year).

Supply the following public member functions in your class.

- A default constructor (taking no arguments) that initializes the Date object to Jan 1, 2000.
- A constructor taking three arguments (month, day, year) that initializes the Date object to the parameter values.
  - It sets the Date's year to 1900 if the year parameter is less than 1900
  - It sets the Date's month to 1 if the month parameter is outside the range of 1 to 12.
  - It sets the Date's day to 1 if the day parameter is outside the range of days for the specific month. Assume February always has 28 days for this test.
- A `getDay` member function that returns the Date's day value.
- A `getMonth` member function that returns the Date's month value.
- A `getYear` member function that returns the Date's year value.
- A `getMonthName` member function that returns the name of the month for the Date's month (e.g. if the Date represents 2/14/2000, it returns "February"). You can return a `const char*` or a `std::string` object from this function.
- A `print` member function that prints the date in the numeric form MM/DD/YYYY to cout (e.g. 02/14/2000). Month and day must be two digits with leading zeros as needed.
- A `printLong` member function that prints the date with the month's name in the form dd month yyyy (e.g. 14 February 2000) to cout. This member function should call the `getMonthName()` member function to get the name. No leading zeroes required for the day.

The class data members should be set to correct values by the *constructor* methods so the get and print member functions simply return or print the data member values. The constructor methods must validate their parameter values (eg. verify the month parameter is within the range of 1 to 12) and only set the Date data members to represent a valid date, thus ensuring the Date object's data members (i.e. its state) always represent a valid date.

The `print` member function should output the date in the format MM/DD/YYYY with leading zeros as needed, using the C++ IOStreams cout object. To get formatting to work with C++ IOStreams (cout), look at the setw() and setfill() manipulator descriptions, or the width() and fill() functions in the chapter on the C++ I/O System.

```
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;  // or use individual directives, e.g. using std::string;

class Date
{

    // methods and data necessary
```

```
};
```

Use separate files for the `Date` class definition (in `Date.h`), implementation of the member functions (`Date.cpp`), and the attached test main() function (`DateDemo.cpp`). The shortest member functions (like getDay()) may be implemented in the class definition (so they will be inlined). Other member functions should be implemented in the `Date.cpp` file. Both `Date.cpp` and `DateDemo.cpp` will need to #include the `Date.h` file (since they both need the `Date` class definition in order to compile) and other include files that are needed (e.g. iostream, string, etc).

Sample code and output

```
// DateDemo.cpp
// Note - you may need to change the definition of the main function to
// be consistent with what your C++ compiler expects.
int main()
{
   cout << "DateDemo starting ..." << endl << endl;

   Date d1;             // default ctor
   Date d2(7, 4, 1976); // July 4'th 1976
   Date d3(0, 15, 1880);// Adjusted by ctor to January 15'th 1900

   d1.print();      // prints 01/01/2000
   d1.printLong();  // prints 1 January 2000
   cout << endl;

   d2.print();      // prints 07/04/1976
   d2.printLong();  // prints 4 July 1976
   cout << endl;

   d3.print();      // prints 01/15/1900
   d3.printLong();  // prints 15 January 1900
   cout << endl;

   cout << "object d2's day is " << d2.getDay() << endl;
   cout << "object d2's month is "  << d2.getMonth() << " which is " << d2.getMonthName() << endl;
   cout << "object d2's year is " << d2.getYear() << endl;
}
```

main() test code

Use the attached DateDemo.cpp source file to test your class.

Save this source as a separate DateDemo.cpp source file that contains the supplied main() function. Compile your Data.cpp file first, then this DateDemo.cpp next.

Links

- Reel Learning on YouTube - Programming Errors (C++)
- Reel Learning on YouTube - Header Files in C++
- Reel Learning on YouTube - First C++ Program
