Lab 16

In this lab, we will write a C++ Abstract Data Type(ADT) named **"dateclass"**, and a client program "main.cpp" that uses this ADT.

An ADT consists of both DATA and METHODs that operate on the data. For the **dateclass** ADT, there are three data elements:

- month (integer value in the range 1-12)
- day (integer value in the range 0-31)
- year (integer value > 0)

and the operations on these data include:

- **two constructors**
    - the default constructor which initializes the date to month equals 1, day equals 1 and year equals to 1970;
    - the value constructor that initializes the date to the month value, the day value and the year value that passed in as value parameters
- **Mutators**, aka the **Set methods**, methods that change one or more values of the data of an object of this ADT:
    - SetYear method that changes the **year** value of the object of this ADT. The new value is passed in as a value parameter.
    - SetMonth method that changes the **month** value of the object of this ADT. The new value is passed in as a value parameter.
    - SetDay method that changes the **day** value of the object of this ADT. The new value is passed in as a value parameter.
    - SetDate method that changes the **month, day, and year** values of the object of this ADT. The new values are passed in as value parameters.
- **Accessors**, aka the **Get methods**, methods that use one or more values of the data of an object of this ADT. These methods should be defined as *constant* methods.
    - GetYear method that returns the **year** value of a date object
    - GetMonth method that returns the **month** value of a date object
    - GetDay method that returns the **day** value of a date object
- Other methods
    - DisplayStandard method that displays the date in "Date: month/day/year" format, i.e., "Date: 2/14/1988".
    - DisplayFull method that displays the date in "Date: month day year" format, where the month is spelled out in full, i.e., "Date: October 17th, 1998.
    - SameDate method that compares two dates: the "self" date and a date object passed in as parameter to this method. It returns true if the "self" date is the same as the second date, and returns false otherwise.
    - PriorTo method that compares two dates: the "self" date and a date object passed in as parameter to this method. It returns true if the "self" date is prior to the second date, and returns false otherwise.

Define the dateclass in two files: the header file named dateclass.h and the implementation file named dateclass.cpp.

After the ADT is defined, write the client program in file main.cpp. Your client program should include the C++ statements to perform the following:

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 2017,
3. Change the date of the first object to month=4, day=17, year=2018.
4. Display the date of the first object
5. Display the month of the second object by calling the "GetMonth" method of the second object,
6. Set the year of the "secondDate" to 2019.
7. Set the month of the "secondDate" to 7.
8. Display the second date in the standard format: "Date: 7/15/2019" by calling the method "DisplayStandard".
9. Display the second date in the full name format: "Date: July 15th, 2019" by calling the method "DisplayFull".
10. Compare "firstDate" and "secondDate" and see if they represent the same date. Treat "firstDate" object as the "self" object, and call the "sameDate" method from this object. Pass "secondDate" object as the parameter of this method. If the two objects represent the same date, output the message: "Same date", otherwise, output the message: "Different dates".
11. Compare "firstDate" and "secondDate" and see if "firstDate" represents a date prior to "secondDate". Treat "firstDate" object as the "self" object, and call the "PriorTo" method from this object. Pass "secondDate" object as the parameter of this method. If the "self" object is prior to the second object, output the message: "First date is prior to the second date", otherwise, output the message: "First date is not prior to the second date".