ECE 71/191T – Data Structures and Algorithms

Dr. Gregory R. Kriehn, Fresno State
C++ Homework Assignment: Chapter 11

Code Due By: Midnight on Mon, Feb 27
Write-up Due By: Class on Tue, Feb 28

HOMEWORK #25 – Calendar Program

Write a program to print the calendar for a particular month or a particular year. For example, the calendar for February 2017 is:

```
                    Feburary 2017
  Sun    Mon    Tue    Wed    Thu    Fri    Sat
                         1      2      3      4
    5      6      7      8      9     10     11
   12     13     14     15     16     17     18
   19     20     21     22     23     24     25
   26     27     28
```

Specifications:
To solve this problem, you will create a number of classes using inheritance and composition.

In Chapter 11 of Malik's book, he uses a class called **`dateType`** (see pages 768-770), which is designed to store the month, day, and year for calendar dates. Use the **`dateType`** class as the initial basis for your program. The class uses the following public and private members:

Public:
```
    void setDate(int month, int day, int year);
    int getDay() const;
    int getMonth() const;
    int getYear() const;
    void printDate() const;
    dateType(int month=1, int day=1, int year=1900);
```

Private:
```
    int dMonth;
    int dDay;
    int dYear;
```

Create the class, and copy the definitions of the class member functions into an appropriate .cpp file. Next, modify the definitions of the function **`setDate()`** and the constructor so that the values for the month, day, and year are checked before storing the date into the member variables. If an incorrect month, day, or year is used, use the default values of 1, 1, and 1900, respectively. Add a new public member function, `isLeapYear()`, to check weather a year is a leap year:

Public:
```
    bool isLeapYear();
```

You may want to write a small program to test the functionality of your class before going further. The rules for leap years are summarized on pages 460-461 of Malik's book – see Problem 6.19.

Next modify your **`dateType`** class so that it can additionally perform the following operations on a date:

a. Set the month individually.
b. Set the day individually.
c. Set the year individually.
d. Return the number of days in the month. For example, if the date is 3-12-2017, the number of days to be returned is 31 because there are 31 days in March.
e. Return the number of days passed in the year. For example, if the date is 3-18-2017, the number of days passed in the year is 77. Note that the number of days returned also includes the current day. Be sure to account for leap years.
f. Return the number of days remaining in the year. For example, if the date is 3-18-2017, the number of days remaining in the year is 288. Be sure to account for leap years.
g. Calculate the new date by adding a fixed number of days to the current date. For example, if the date is 3-18-2017 and the days to be added are 25, the new date is 4-12-2017.

Add the following member functions:

Public:
```
    void setMonth(int);
    void setDay(int);
    void setYear(int);
    int getDaysInMonth();
    int numberOfDaysPassed();
    int numberOfDaysLeft();
    void incrementDate(int nDays);
```

You may want to write a small test program to verify the operation of your class member functions.

The class member functions in **`dateType`** print the date in numerical form. Some applications may require the date to be printed in another form, such as "March 24, 2017". Use inheritance to drive a new class **`extDateType`** so the date can be printed in either form. Use the following members:

Public:
```
    void printLongDate();
    // Prints Month Day, Year using a string name for the
    // month.
    void printLongMonthYear();
    // Prints the Month and Year using a string name for the
    // month.
    void setDate(int month, int day, int year);
    // Function to store the month, day, and year in numerical
    // format.  Also stores the string name of the month
    void setMonth(int month);
    // Function to store the month in numerical format.
    // Also stores the string name of the month
    extDateType();
    // Default Constructor
    extDateType(int, int, int);
    // Constructor, should call dateType constructor, as
    // necessary, and also store month in string format.
```

Private:

```
    string month;
```

You may want to write a small test program to test the functionality of the class.

Next, design and implement a class called **dayType** that implements the day of the week in a program. The class should store the day of the week as a string. The program should be able to perform the following operations on an object of type **dayType**:

a. Set the day.
b. Print the day.
c. Return the day.
d. Return the next day.
e. Return the previous day.
f. Calculate and return the day by adding certain days to the current day. For example, if the current day is Monday and we add 4 days, the day to be returned is Friday. Similarly, if today is Tuesday and we add 13 days, the day to be returned is Monday.
g. Add appropriate constructors.

Use the following members:

Public:

```
    static string weekDays[7];
    // Stores the days of the week in string format
    void print() const;
    // Prints the weekday as a string.
    string nextDay() const;
    // Returns the string of the next day
    string prevDay() const;
    // Returns the string of the previous day
    void addDay(int nDays);
    // Calculates and stores the appropriate day by adding a
    // certain number of days to the current day.
    void setDay(string day);
    // Set the day to the input
    string getDay() const;
    // Get the current day.
    dayType();
    // Default constructor, set to Sunday.
    dayType(string day);
    // Constructor, set day to input.
```

Private:

```
    string weekDay;
    // Stores the current day as a string
```

You may want to write a small test program to test the functionality of the class.

Now use the classes **`extDateType`** and **`dayType`** to design a class **`calendarType`** using composition so that, given the month and the year, we can print the calendar for that month. To print a monthly calendar, you must know the first day of the month and the number of days in that month. Thus, you must store the first day of the month, which is of the form **`dayType`**, and the month and year of the calendar. The month and the year can be stored in an object of the form **`extDateType`** by setting the day component of the date to 1 and the month and year as specified by the user. Thus, the class **`calendarType`** has two member variables: an object of the type **`dayType`** and an object of the type **`extDateType`**.

Design the class **`calendarType`** so that the program can print a calendar for any month starting January 1, 1500, which is a Monday. To calculate the day of a month, you can add the appropriate days to Monday, January 1, 1500. Be sure to account for leap years. Use the following members for your class:

Public:

```
    void setMonth(int month);
    // Sets the month inside the firstDate object of type
    // extDateType
    void setYear(int year);
    // Sets the year inside the firstDate object of type
    // extDateType
    int getMonth();
    // Gets the month from the firstDate object
    int getYear();
    // Gets the year from the firstDate object
    void printCalendar();
    // Prints the calendar based upon the current month and
    // year in firstDate
    calendarType();
    // Default constructor, defaults to January 1, 1500, which
    // is a Monday.
    // Set the appropriate month, day, year, and weekday in
    // firstDate
    calendarType(int m, int y);
    // Sets the month, year, (and day) in firstDate.  The day
    // should default to 1.
    // The weekday should be set appropriately.
```

Private:

```
    dayType firstDayOfMonth();
    // Function to determine the weekday of the first day of
    // the month, based on the year.
    void printTitle();
    // Function to print the title (header information) for the
    // calendar
    void printDates();
    // Function to print the dates in the month
    extDateType firstDate;
    // firstDate object of type extDateType
    dayType firstDay;
    // firstDay object of type dayType
```

You may want to write a small test program to test the functionality of the class.

Finally, write a test program to print the calendar for a particular month and year.