driver:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name: Michael Quach Z#: Z23255459

Course: Date Structures and Algorithm Analysis (COP3530)

Professor: Dr. Lofton Bullard

Due Date: September 1, 2015 Due Time: 5 PM

Total Points: 100

Assignment 3: Date program

Description: The skeleton of a Date class is provided, which prints out the date.

I will fill in the skeleton with actual coding.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef DATE\_CPP

#include "date.cpp"

int main()

{

Date myDate;

Date yourDate(12,31, 1957);

Date test1Date(15, 1, 1962); //should generate error message that bad month

Date test2Date(2, 29, 1956); //ok, should say leep year

Date test3Date(2, 30, 1956); //should generate error message that bad day

Date test4Date(12,31,0000); //should generate error message that bad year

Date test5Date(80,40,0000); //should generate error message that bad month, day and year

yourDate.display();

cout<<yourDate.getMonth()<<endl;

cout<<yourDate.getDay()<<endl;

cout<<yourDate.getYear()<<endl;

cout<<"myDate: "<<myDate<<" test2Date: "<<test2Date<<" yourDate: "<<yourDate<<endl;

return 0;

}

#endif

.h:

#include <iostream>

#include <string>

//#include "date.h"

using namespace std;

//#ifndef DATE\_H

//#define DATE\_H

class Date

{

public:

Date(); //default constructor; sets m=01, d=01, y =0001

Date(unsigned m, unsigned d, unsigned y); //explicit-value constructor to set date equal to today's

//date. Use 2-digits for month (m) and day (d), and 4-digits for year (y); this function should //print a message if a leap year.

void display();//output Date object to the screen

int getMonth();//accessor to output the month

int getDay();//accessor to output the day

int getYear();//accessor to output the year

void setMonth(unsigned m);//mutator to change the month

void setDay(unsigned d);//mutator to change the day

void setYear(unsigned y);//mutation to change the year

friend ostream & operator<<(ostream & out, const Date & dateObj);//overloaded operator<< as a friend function with chaining

//you make add other functions if necessary

private:

int myMonth, myDay, myYear; //month, day, and year of a Date obj respectively

};

//#endif

date.cpp:

#include "date.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name: Date

//Precondition: The state of the object (private data) has not been initialized

//Postcondition: The state has been initialized to today's date

//Description: This is the default constructor which will be called automatically when

//an object is declared. It will initialize the state of the class

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Date::Date()

{

cout << "Default constructor has been called.\n";

myMonth = 01;

myDay = 01;

myYear = 0001;

//cout << "About to print Date\n";

//cout << Date; //OSTREAM TESTING GROUNDS

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name: Date

//Precondition: The state of the object has not been initialized.

//Postcondition: The state has been initialized to the given date.

//Description: This is the explicit-value constructor, which will be called if an object

//is declared with arguments. The state of the class is initialized based on the arguments.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Date::Date(unsigned m, unsigned d, unsigned y)

{

cout << "Explicit-value constructor has been called.\n";

if(m > 0 && m < 13){

myMonth = m;}

else{

cout << "Month " << m << " is invalid.";

myMonth = 01;}

if(m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12){

if(d > 0 && m < 32){

myDay = d;}

else{

cout << "Day " << d << " is invalid for month " << m << ".\n";

myDay = 01;}}

else if(m == 4 || m == 6 || m == 9 || m == 11){

if(d > 0 && m < 31){

myDay = d;}

else{

cout << "Day " << d << " is invalid for month " << m << ".\n";

myDay = 01;}}

else if(m == 2){

if(y%4 == 0){

if(d > 0 && m < 29){

myDay = d;}

else{

cout << "Day " << d << " is invalid for month " << m << ".\n";

myDay = 01;}}

else{

if(d > 0 && m < 29){

myDay = d;}

else{

cout << "Day " << d << " is invalid for month " << m << ".\n";

myDay = 01;}}}

if(y > 0){

myYear = y;}

else{

cout << "Year " << y << " is invalid.\n";

myYear = 0001;}

// cout << Date; //OSTREAM TESTING GROUNDS

if(y%4 == 0){

cout << "This is a leap year.\n";}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name: Display

//Precondition: Values aren't printed.

//Postcondition: The current values for the Date class are printed.

//Description: This is essentially an accessor for all three variables.

//It displays the current values for the month, day, and year in mm/dd/yyyy format.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Date::display()

{

cout << myMonth << "/" << myDay << "/" << myYear << "\n";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name: getMonth

//Precondition: Value not known.

//Postcondition: The value for the month is returned.

//Description: Returns the value of the month variable.

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int Date::getMonth()

{

cout << "Month == " << myMonth;

return myMonth;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name: getDay

//Precondition: Value not known.

//Postcondition: The value of the day variable is returned.

//Description: Returns the Day value.

//

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int Date::getDay()

{

cout << "Day == " << myDay;

return myDay;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name: getYear

//Precondition: Value not known.

//Postcondition: Value of Year returned.

//Description: The value of the Year variable is returned.

//

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int Date::getYear()

{

cout << "Year == " << myYear;

return myYear;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name: setMonth

//Precondition:

//Postcondition:

//Description:

//

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Date::setMonth(unsigned m)

{

myMonth = m;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name: setDay

//Precondition:

//Postcondition:

//Description:

//

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Date::setDay(unsigned d)

{

myDay = d;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Name: getYear

//Precondition:

//Postcondition:

//Description:

//

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Date::setYear(unsigned y)

{

myYear = y;

}

ostream & operator<<(ostream & out, const Date & dateObj)

{

cout << "In ostream function\n";

cout << dateObj.myMonth << "/" << dateObj.myDay << "/" << dateObj.myYear << "\n";

return out;

}