// ========================

// Attached: HW\_7a, HW\_7b

// ========================

// Program: HW\_7a

// ========================

// Programmer: Ashley Gilliam

// Class: CS 1B

// ========================

#include<iostream>

#include"Date.h"

using namespace std;

int main()

{

Date today = {3,11,2019};

Date payday;

int month;

int day;

int year;

cout << "Enter the date of the payday:\nMonth: ";

cin >> month;

cout << "Day: ";

cin >> day;

cout << "Year: ";

cin >> year;

payday.setDate(month, day, year);

cout << "\nToday's date is ";

today.displayDate();

cout << "\nPayday is on ";

payday.displayDate();

payday.addDays(7);

cout << "\n\nThe next payday is on ";

payday.displayDate();

cout << endl;

system("pause");

return 0;

}

// ============== OUTPUT ================

/\*

Enter the date of the payday:

Month: 12

Day: 30

Year: 1999

Today's date is 3/11/2019

Payday is on 12/30/1999

The next payday is on 1/7/2000

Press any key to continue . . .

\*/

// ======================================

// ========================

// Program: Date.cpp

// ========================

#include<iostream>

#include"Date.h"

using namespace std;

Date::Date()

{

month = 0;

day = 0;

year = 0;

}

Date::Date(int month, int day, int year)

{

this->month = month;

this->day = day;

this->year = year;

}

Date::~Date() {}

void Date::setDate(int month, int day, int year)

{

this->month = month;

this->day = day;

this->year = year;

}

void Date::addDays(int days)

{

for (int i = 0; i < days; i++)

{

day++;

if (day > 30)

{

day = 1;

month++;

}

if (month > 12)

{

month = 1;

year++;

}

}

}

void Date::displayDate()

{

cout << month << '/' << day << '/' << year;

}

// ========================

// Program: Date.h

// ========================

#pragma once

#include<iostream>

using namespace std;

Class Date

{

private:

int month;

int day;

int year;

public:

Date();

Date(int,int,int);

~Date();

void setDate(int,int,int);

void addDays(int);

void displayDate();

};

// ========================

// Attached: HW\_7a, HW\_7b

// ========================

// Program: HW\_7b

// ========================

// Programmer: Ashley Gilliam

// Class: CS 1B

// ========================

#include<iostream>

#include"ClockType.h"

using namespace std;

int main()

{

ClockType myClock;

ClockType yourClock;

ClockType myOtherClock(0, 0, 0);

myClock.setTime(3, 30, 0);

yourClock.setTime(10, 0, 15);

cout << "The time on my clock is: ";

myClock.printTime();

cout << ".\n\nThe time on your clock is: ";

yourClock.printTime();

if (myClock.areTimesEqual(yourClock) == true)

{

cout << ".\n\nThe times of my clock and your clock are equal.\n\n";

}

else

{

cout << ".\n\nThe times of my clock and your clock are not equal.\n\n";

cout << "The time on my other clock is: ";

myOtherClock.printTime();

cout << ".\n\nI am going to add 1 minute and 1 second to my other clock.\n\nThe time on my other clock is: ";

myOtherClock.incrementMinutes();

myOtherClock.incrementSeconds();

myOtherClock.printTime();

cout << ".\n\n";

}

system("pause");

return 0;

}

// ============== OUTPUT ================

/\*

The time on my clock is: 03:30:00.

The time on your clock is: 10:00:15.

The times of my clock and your clock are not equal.

The time on my other clock is: 00:00:00.

I am going to add 1 minute and 1 second to my other clock.

The time on my other clock is: 00:01:01.

Press any key to continue . . .

\*/

// ======================================

// ========================

// Program: ClockType.ccp

// ========================

#include<iostream>

#include"ClockType.h"

using namespace std;

ClockType::ClockType()

{

hrs = 0;

mins = 0;

secs = 0;

}

ClockType::ClockType(int hrs, int mins, int secs)

{

this->hrs = hrs;

this->mins = mins;

this->secs = secs;

}

ClockType::~ClockType() {}

void ClockType::setTime(int hrs, int mins, int secs)

{

this->hrs = hrs;

this->mins = mins;

this->secs = secs;

}

void ClockType::incrementSeconds()

{

secs++;

if (secs > 59)

{

secs = 0;

mins++;

}

if (mins > 59)

{

mins = 0;

hrs++;

}

if (hrs > 12)

hrs = 1;

}

void ClockType::incrementMinutes()

{

Mins++;

if (mins > 59)

{

mins = 0;

hrs++;

}

if (hrs > 12)

hrs = 1;

}

void ClockType::incrementHours()

{

Hrs++;

if (hrs > 12)

hrs = 1;

}

void ClockType::printTime()

{

if (hrs < 10)

cout << "0" << hrs;

else

cout << hrs;

cout << ":";

if (mins < 10)

cout << "0" << mins;

else

cout << mins;

cout << ":";

if (secs < 10)

cout << "0" << secs;

else

cout << secs;

}

bool ClockType::areTimesEqual(ClockType clock)

{

if (hrs == clock.hrs && mins == clock.mins && secs == clock.secs)

{

return true;

}

else

{

return false;

}

}

// ========================

// Program: ClockType.h

// ========================

#pragma once

#include<iostream>

using namespace std;

class ClockType

{

private:

int hrs;

int mins;

int secs;

public:

ClockType();

ClockType(int,int,int);

~ClockType();

void setTime(int,int,int);

void incrementSeconds();

void incrementMinutes();

void incrementHours();

void printTime();

bool areTimesEqual(ClockType);

};