Course: Programming Fundamentals – **ENCM 339**

Lab #: Lab 7

Instructor: S. Norman

Student Name: **Mitchell Sawatzky**

Lab Section: **B02**

Date Submitted: **Nov 3, 2015**

**Exercise C**

lab7\_time.h

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

\* Title: lab7\_time.h \*

\* Name: Mitchell Sawatzky \*

\* UCID: 10146721 \*

\* Class: ENCM 339-T01/B02 \*

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

#ifndef lab7\_exe\_C\_Time

#define lab7\_exe\_C\_Time

class Time {

public:

Time ();

Time (int sec);

int get\_hour() const;

int get\_minute() const;

int get\_second() const;

void set\_time(int n);

void increment\_by\_n(int n);

void decrement\_by\_n(int n);

Time add(Time other\_time);

private:

int hour;

int minute;

int second;

int Time\_to\_sec();

Time sec\_to\_hms(int n);

};

#endif

lab7\_time.cpp

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

\* Title: lab7\_time.cpp \*

\* Name: Mitchell Sawatzky \*

\* UCID: 10146721 \*

\* Class: ENCM 339-T01/B02 \*

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

#include "lab7\_time.h"

Time::Time(): hour(0), minute(0), second(0) { }

Time::Time(int sec) {

this->set\_time(sec);

}

int Time::get\_hour() const {

return hour;

}

int Time::get\_minute() const {

return minute;

}

int Time::get\_second() const {

return second;

}

void Time::set\_time(int n) {

Time local = sec\_to\_hms(n);

this->hour = local.get\_hour();

this->minute = local.get\_minute();

this->second = local.get\_second();

return;

}

void Time::increment\_by\_n(int n) {

if (n <= 0)

return;

this->set\_time(n + this->Time\_to\_sec());

return;

}

void Time::decrement\_by\_n(int n) {

if (n <= 0)

return;

int cTime = this->Time\_to\_sec() - n;

this->set\_time(cTime < 0 ? 0 : cTime);

return;

}

Time Time::add(Time other\_time) {

Time local(other\_time.Time\_to\_sec() + this->Time\_to\_sec());

return local;

}

int Time::Time\_to\_sec() {

return hour \* 3600 + minute \* 60 + second;

}

Time Time::sec\_to\_hms(int n) {

Time local;

if (n <= 0)

return local;

local.hour = n / 3600;

local.minute = (n % 3600) / 60;

local.second = n % 3600 % 60;

return local;

}

Terminal Output:

Mitchell@ttys000 11:20 {0} [lab7]$ ./test.out

00:00:00

02:10:32

02:10:32

02:11:32

02:10:32

02:10:32

01:01:59

03:12:31

00:00:00

13:53:20

13:53:20

00:00:00

00:00:00

**Exercise D**

Function: average

Point average(const Point \*arr, int n)

{

Point local;

double xA = 0, yA = 0;

for (int i = 0; i < n; i++) {

xA += arr[i].getx();

yA += arr[i].gety();

}

local.setx(xA / (double)n);

local.sety(yA / (double)n);

return local;

}

Terminal Output:

Mitchell@ttys000 11:24 {0} [lab7]$ ./test.out

Point 9999: <-9999.00, -9999.00>

Point 100: <45.00, 30.00>

Point 9999: <-9999.00, -9999.00>

Point 9999: <-9999.00, -9999.00>

Array of points, gamma, contains:

Point 101: <5.00, 2.00>

Point 102: <10.00, 4.00>

Point 103: <15.00, 6.00>

Point 104: <20.00, 8.00>

Point 105: <25.00, 10.00>

Point 106: <30.00, 12.00>

Point 107: <35.00, 14.00>

Point 108: <40.00, 16.00>

Point 109: <45.00, 18.00>

Point 110: <50.00, 20.00>

The point with the average of points in array gamma is:

Point 9999: <27.50, 11.00>