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>
```