Course: Programming Fundamental – ENSF 337

Lab #: Lab 7

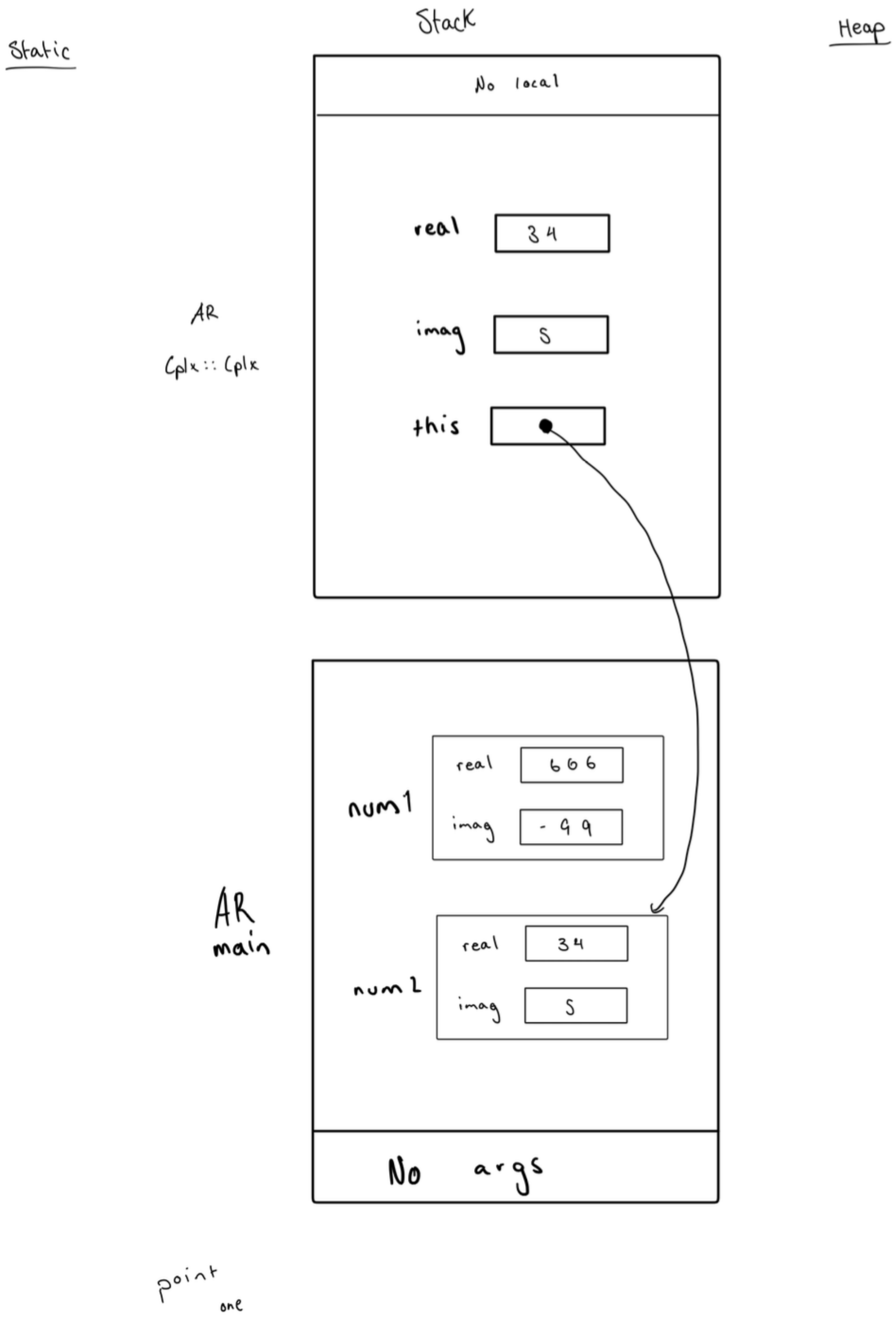
Instructor: M. Moussavi

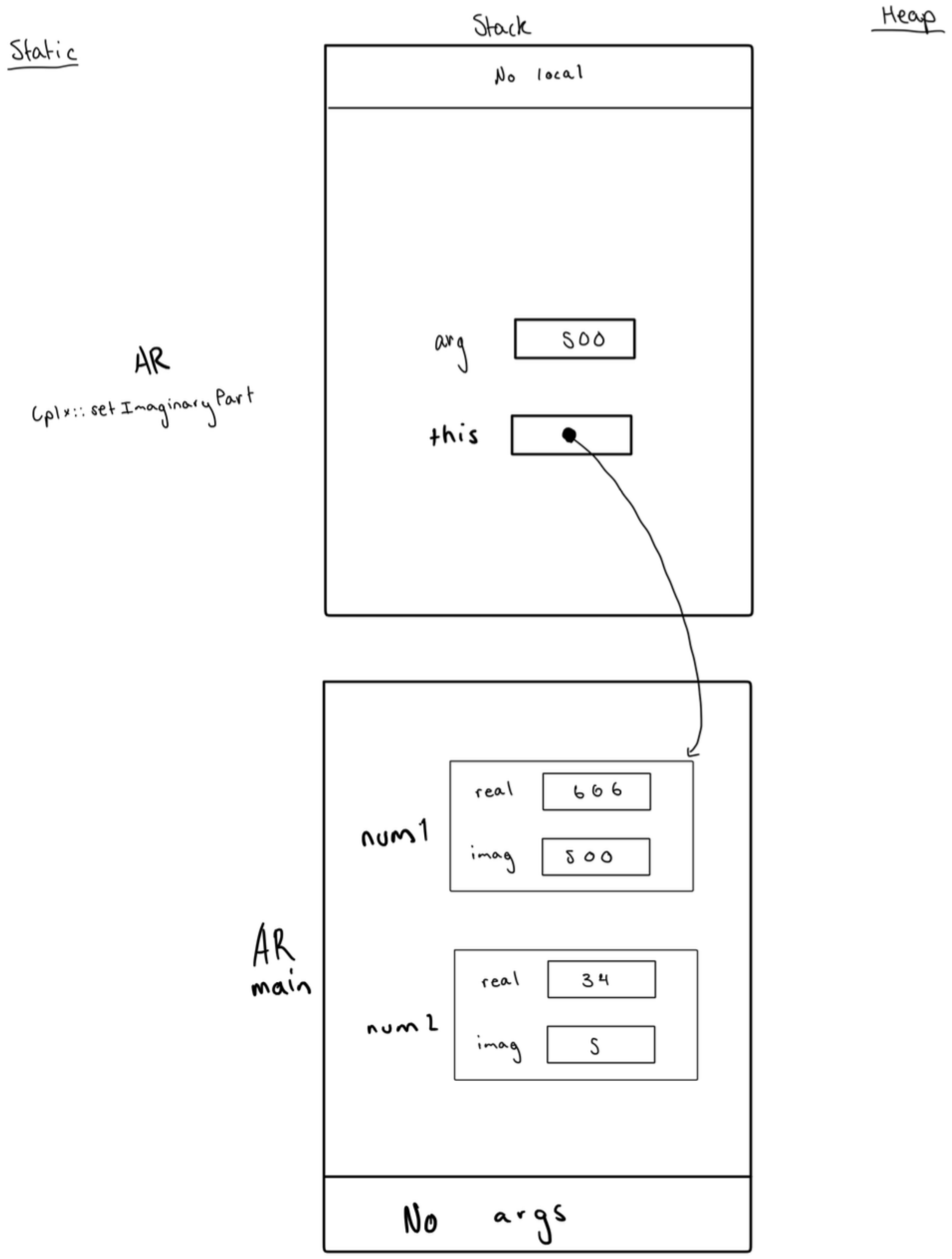
Student Name: Carl Soriano

Lab Section: B01

Date submitted: November 17, 2022

Exercise A


Static
Stack
Heap
No local
real
34
imag
5
this
AR
Cplx::Cplx
num1
real
666
imag
-99
AR
main
num2
real
34
imag
5
No args
point
one


Static
Stack
Heap
No local
arg
500
this
AR
Cplx::setImaginaryPart
real
666
num1
imag
500
AR
main
real
34
num2
imag
5
No args

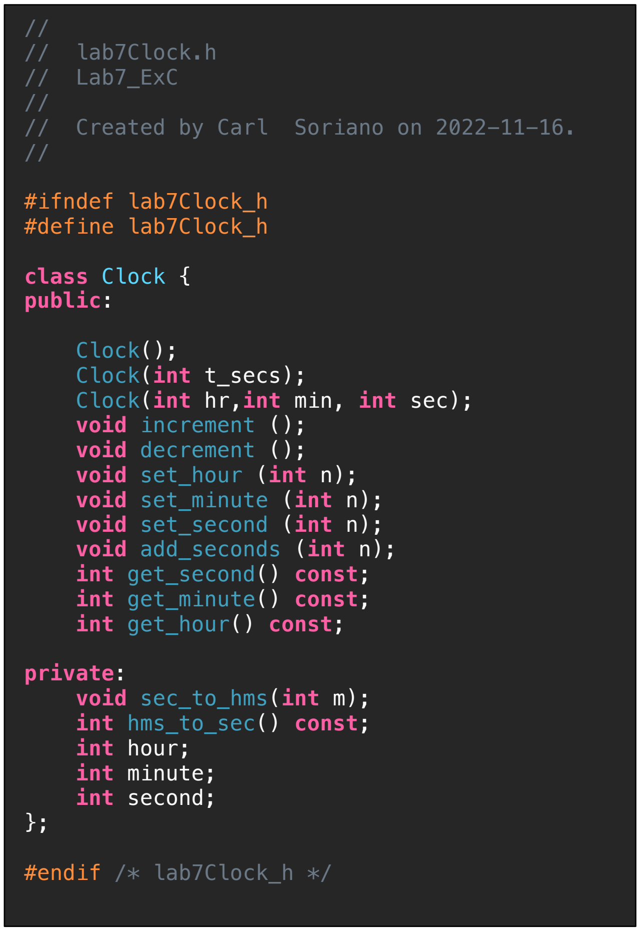
point

two

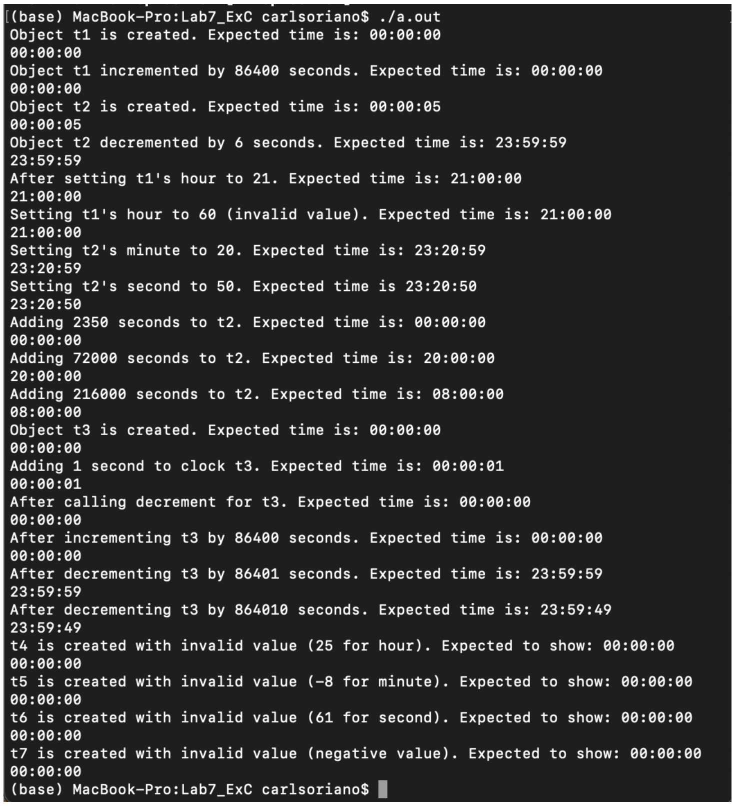
Exercise B

NOT Graded

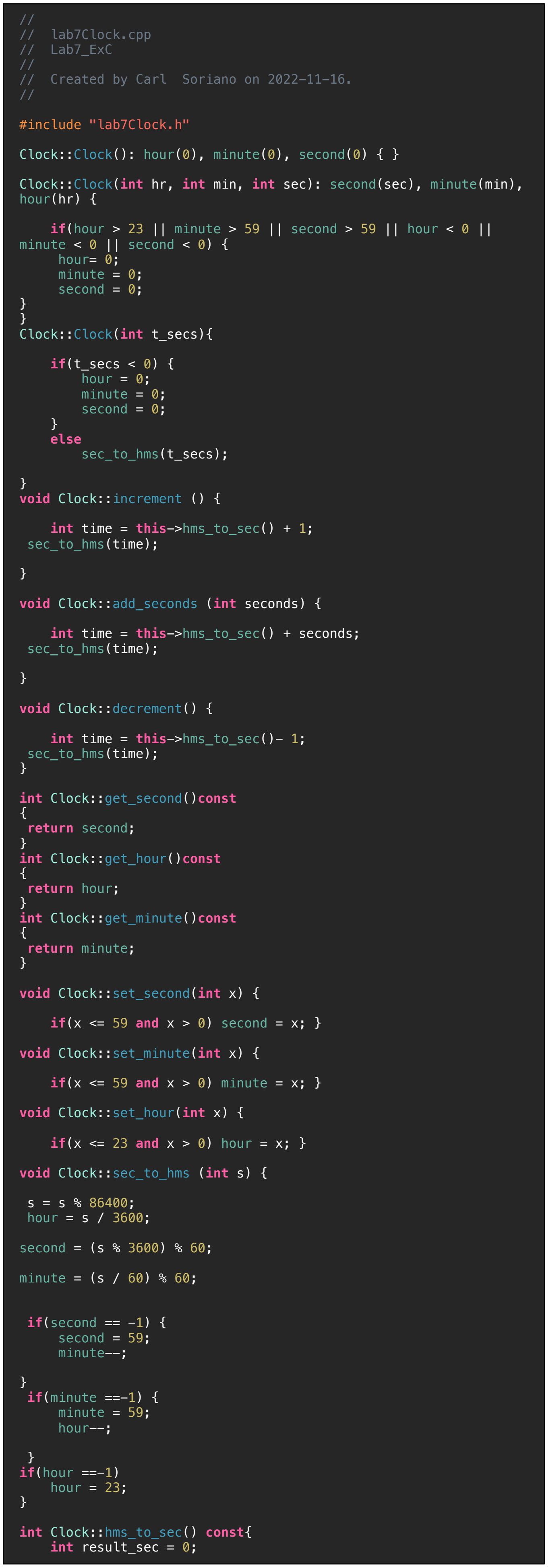
Exercise C



```
//
//  lab7Clock.h
//  Lab7_ExC
//
//  Created by Carl  Soriano on 2022-11-16.
//

#ifndef lab7Clock_h
#define lab7Clock_h

class Clock {
public:

    Clock();
    Clock(int t_secs);
    Clock(int hr,int min, int sec);
    void increment ();
    void decrement ();
    void set_hour (int n);
    void set_minute (int n);
    void set_second (int n);
    void add_seconds (int n);
    int get_second() const;
    int get_minute() const;
    int get_hour() const;

private:
    void sec_to_hms(int m);
    int hms_to_sec() const;
    int hour;
    int minute;
    int second;
};

#endif /* lab7Clock_h */
```

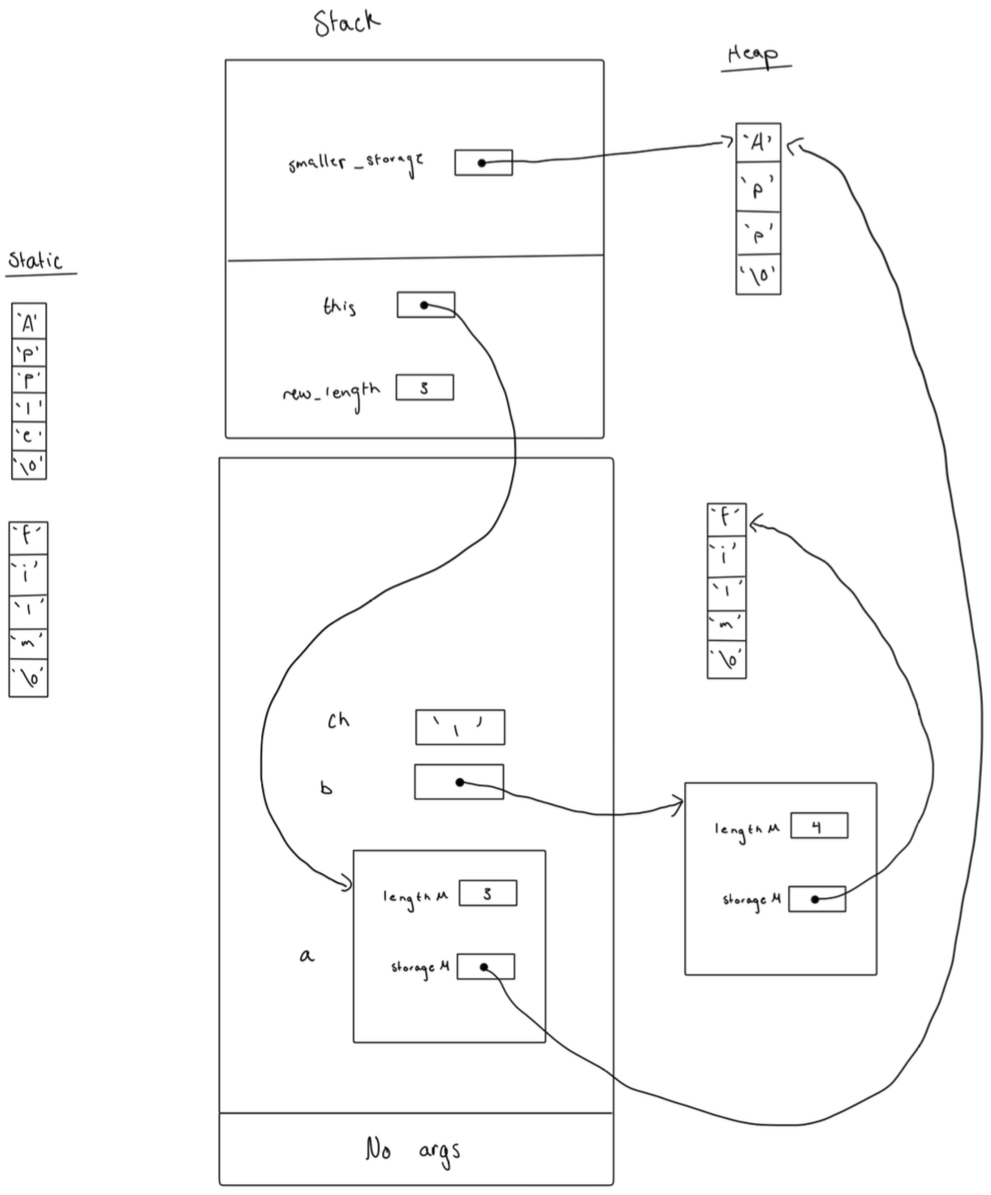


```
(base) MacBook-Pro:Lab7_ExC carlsoriano$ ./a.out
Object t1 is created. Expected time is: 00:00:00
00:00:00
Object t1 incremented by 86400 seconds. Expected time is: 00:00:00
00:00:00
Object t2 is created. Expected time is: 00:00:05
00:00:05
Object t2 decremented by 6 seconds. Expected time is: 23:59:59
23:59:59
After setting t1's hour to 21. Expected time is: 21:00:00
21:00:00
Setting t1's hour to 60 (invalid value). Expected time is: 21:00:00
21:00:00
Setting t2's minute to 20. Expected time is: 23:20:59
23:20:59
Setting t2's second to 50. Expected time is 23:20:50
23:20:50
Adding 2350 seconds to t2. Expected time is: 00:00:00
00:00:00
Adding 72000 seconds to t2. Expected time is: 20:00:00
20:00:00
Adding 216000 seconds to t2. Expected time is: 08:00:00
08:00:00
Object t3 is created. Expected time is: 00:00:00
00:00:00
Adding 1 second to clock t3. Expected time is: 00:00:01
00:00:01
After calling decrement for t3. Expected time is: 00:00:00
00:00:00
After incrementing t3 by 86400 seconds. Expected time is: 00:00:00
00:00:00
After decrementing t3 by 86401 seconds. Expected time is: 23:59:59
23:59:59
After decrementing t3 by 864010 seconds. Expected time is: 23:59:49
23:59:49
t4 is created with invalid value (25 for hour). Expected to show: 00:00:00
00:00:00
t5 is created with invalid value (-8 for minute). Expected to show: 00:00:00
00:00:00
t6 is created with invalid value (61 for second). Expected to show: 00:00:00
00:00:00
t7 is created with invalid value (negative value). Expected to show: 00:00:00
00:00:00
(base) MacBook-Pro:Lab7_ExC carlsoriano$ 
```



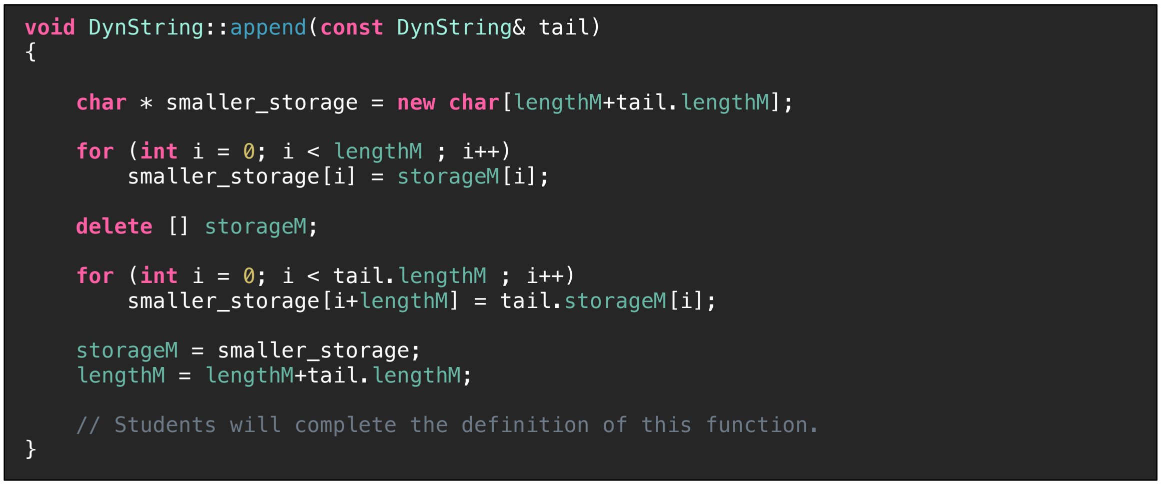
```
//
//  lab7Clock.cpp
//  Lab7_ExC
//
//  Created by Carl  Soriano on 2022-11-16.
//

#include "lab7Clock.h"

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

Clock::Clock(int hr, int min, int sec): second(sec), minute(min),
hour(hr) {

    if(hour > 23 || minute > 59 || second > 59 || hour < 0 ||
minute < 0 || second < 0) {
     hour= 0;
     minute = 0;
     second = 0;
}
}
Clock::Clock(int t_secs){

    if(t_secs < 0) {
        hour = 0;
        minute = 0;
        second = 0;
    }
    else
        sec_to_hms(t_secs);

}
void Clock::increment () {

    int time = this->hms_to_sec() + 1;
 sec_to_hms(time);

}

void Clock::add_seconds (int seconds) {

    int time = this->hms_to_sec() + seconds;
 sec_to_hms(time);

}

void Clock::decrement() {

    int time = this->hms_to_sec()- 1;
 sec_to_hms(time);
}

int Clock::get_second()const
{
 return second;
}
int Clock::get_hour()const
{
 return hour;
}
int Clock::get_minute()const
{
 return minute;
}

void Clock::set_second(int x) {

    if(x <= 59 and x > 0) second = x; }

void Clock::set_minute(int x) {

    if(x <= 59 and x > 0) minute = x; }

void Clock::set_hour(int x) {

    if(x <= 23 and x > 0) hour = x; }

void Clock::sec_to_hms (int s) {

 s = s % 86400;
 hour = s / 3600;

second = (s % 3600) % 60;

minute = (s / 60) % 60;


 if(second == -1) {
     second = 59;
     minute--;

}
 if(minute ==-1) {
     minute = 59;
     hour--;

 }
if(hour ==-1)
    hour = 23;
}

int Clock::hms_to_sec() const{
    int result_sec = 0;
```

Exercise D

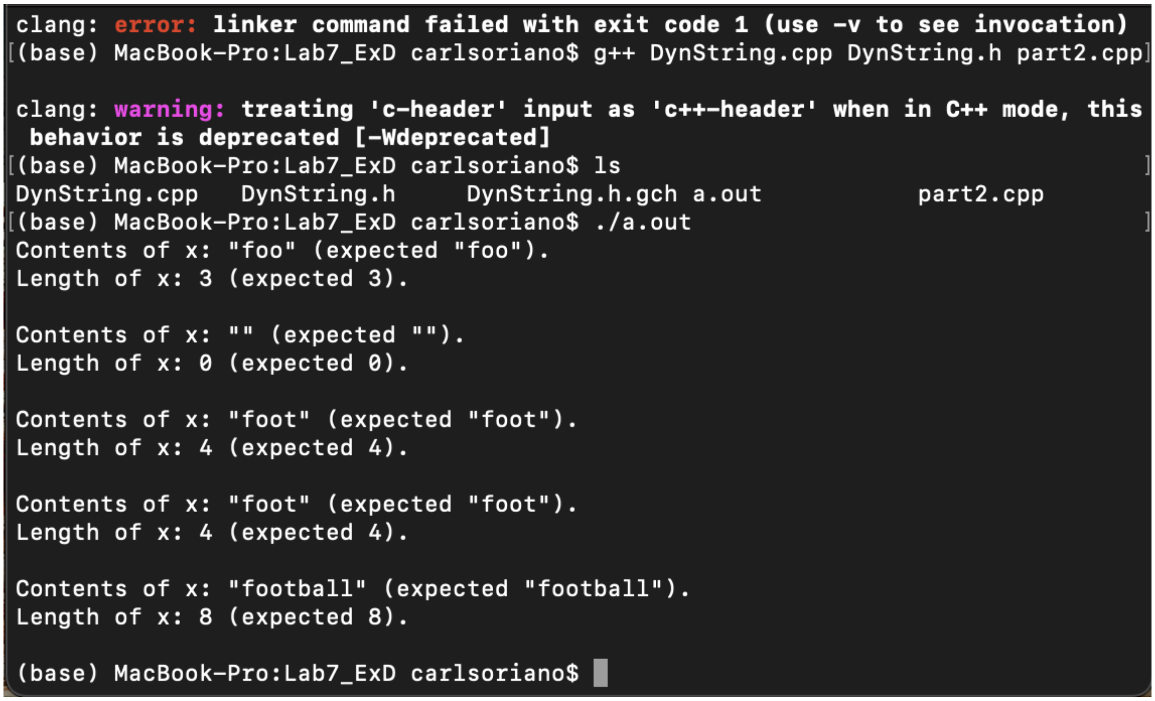


1. Constructor called 3 times and Destructor once
2. Constructor called 3 times and Destructor 3 times

Part 2

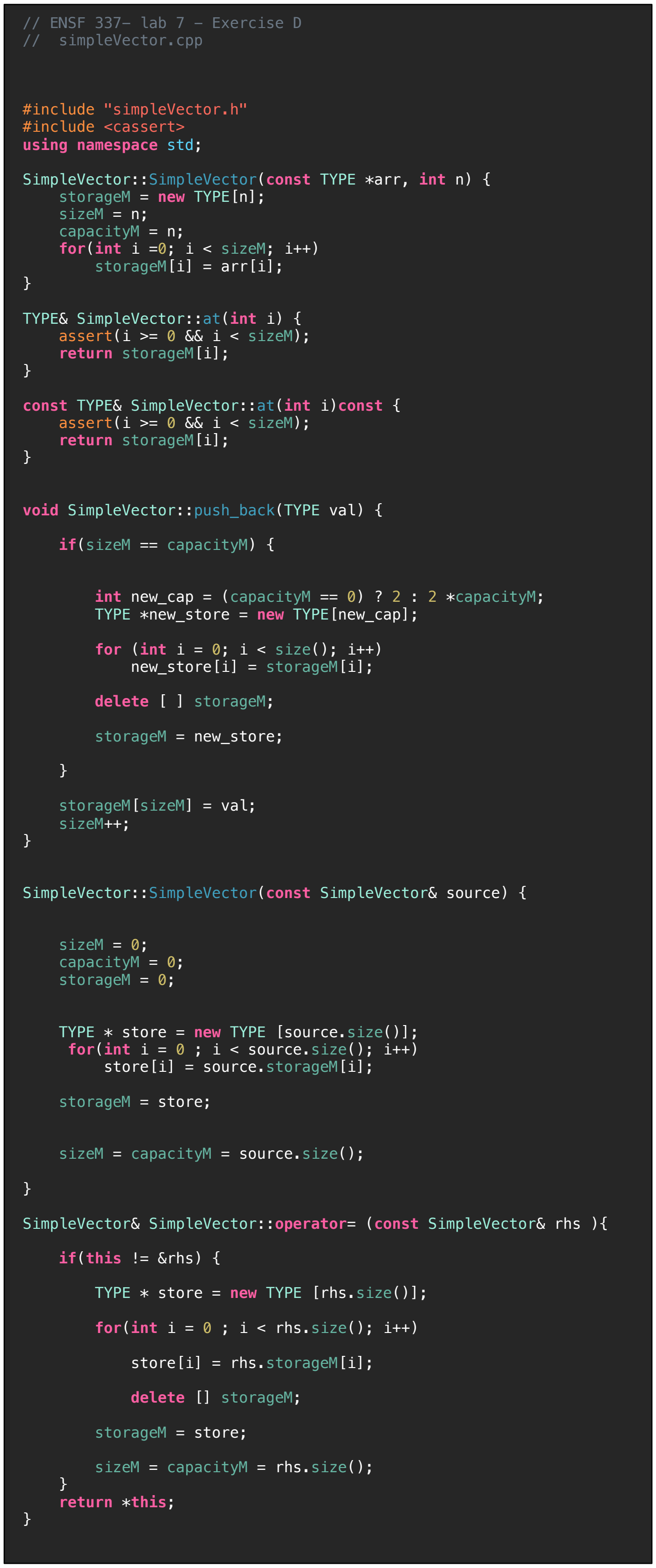


```cpp
void DynString::append(const DynString& tail)
{

    char * smaller_storage = new char[lengthM+tail.lengthM];

    for (int i = 0; i < lengthM ; i++)
        smaller_storage[i] = storageM[i];

    delete [] storageM;

    for (int i = 0; i < tail.lengthM ; i++)
        smaller_storage[i+lengthM] = tail.storageM[i];

    storageM = smaller_storage;
    lengthM = lengthM+tail.lengthM;

    // Students will complete the definition of this function.
}
```

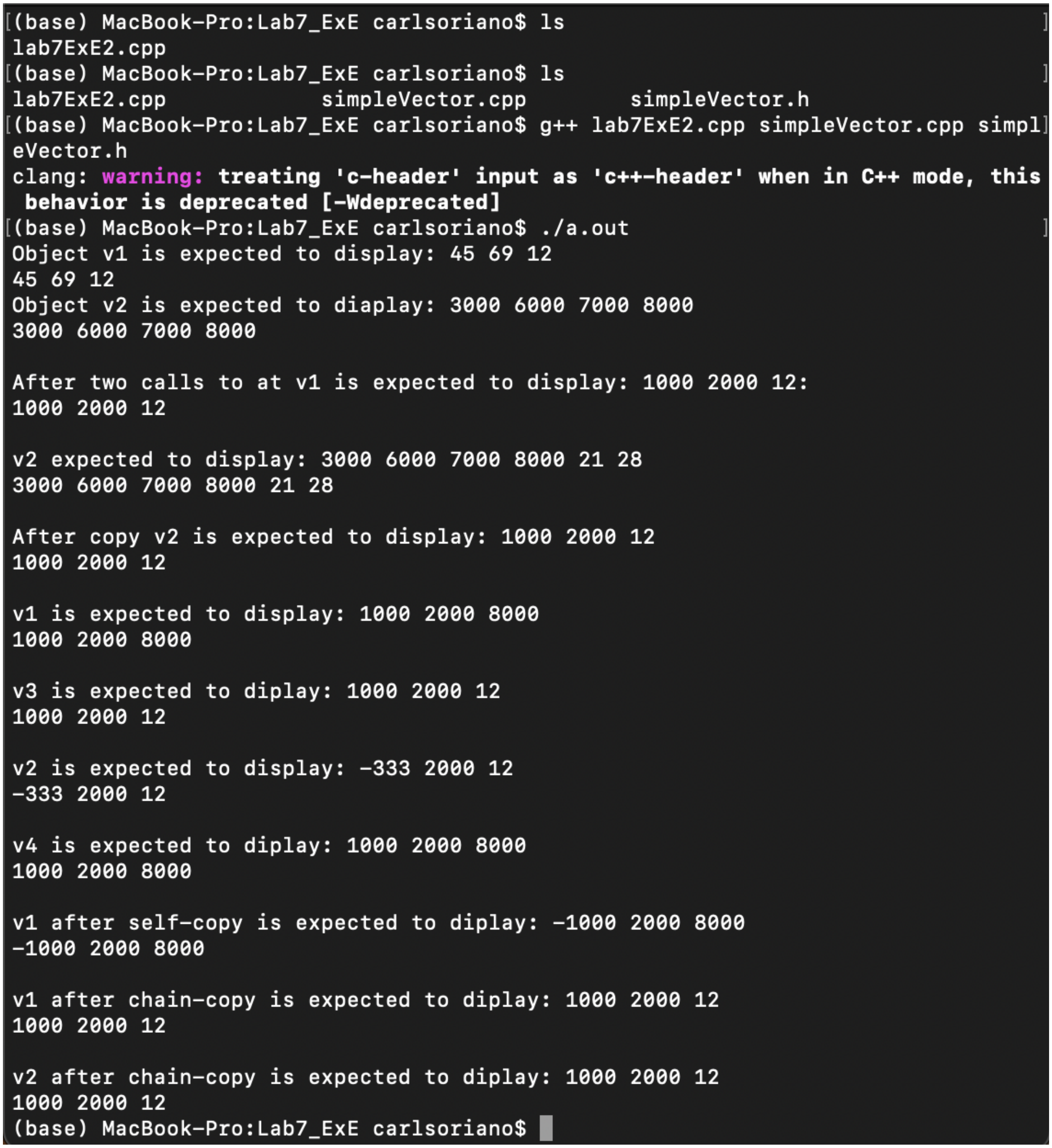


```
clang: error: linker command failed with exit code 1 (use -v to see invocation)
[(base) MacBook-Pro:Lab7_ExD carlsoriano$ g++ DynString.cpp DynString.h part2.cpp]

clang: warning: treating 'c-header' input as 'c++-header' when in C++ mode, this
 behavior is deprecated [-Wdeprecated]
[(base) MacBook-Pro:Lab7_ExD carlsoriano$ ls                                     ]
DynString.cpp   DynString.h     DynString.h.gch a.out           part2.cpp
[(base) MacBook-Pro:Lab7_ExD carlsoriano$ ./a.out                                ]
Contents of x: "foo" (expected "foo").
Length of x: 3 (expected 3).

Contents of x: "" (expected "").
Length of x: 0 (expected 0).

Contents of x: "foot" (expected "foot").
Length of x: 4 (expected 4).

Contents of x: "foot" (expected "foot").
Length of x: 4 (expected 4).

Contents of x: "football" (expected "football").
Length of x: 8 (expected 8).

(base) MacBook-Pro:Lab7_ExD carlsoriano$ █
```

Exercise E



```
// ENSF 337- lab 7 - Exercise D
//  simpleVector.cpp



#include "simpleVector.h"
#include <cassert>
using namespace std;

SimpleVector::SimpleVector(const TYPE *arr, int n) {
    storageM = new TYPE[n];
    sizeM = n;
    capacityM = n;
    for(int i =0; i < sizeM; i++)
        storageM[i] = arr[i];
}

TYPE& SimpleVector::at(int i) {
    assert(i >= 0 && i < sizeM);
    return storageM[i];
}

const TYPE& SimpleVector::at(int i)const {
    assert(i >= 0 && i < sizeM);
    return storageM[i];
}


void SimpleVector::push_back(TYPE val) {

    if(sizeM == capacityM) {


        int new_cap = (capacityM == 0) ? 2 : 2 *capacityM;
        TYPE *new_store = new TYPE[new_cap];

        for (int i = 0; i < size(); i++)
            new_store[i] = storageM[i];

        delete [ ] storageM;

        storageM = new_store;

    }

    storageM[sizeM] = val;
    sizeM++;
}


SimpleVector::SimpleVector(const SimpleVector& source) {


    sizeM = 0;
    capacityM = 0;
    storageM = 0;


    TYPE * store = new TYPE [source.size()];
     for(int i = 0 ; i < source.size(); i++)
         store[i] = source.storageM[i];

    storageM = store;


    sizeM = capacityM = source.size();

}

SimpleVector& SimpleVector::operator= (const SimpleVector& rhs ){

    if(this != &rhs) {

        TYPE * store = new TYPE [rhs.size()];

        for(int i = 0 ; i < rhs.size(); i++)

            store[i] = rhs.storageM[i];

            delete [] storageM;

        storageM = store;

        sizeM = capacityM = rhs.size();
    }
    return *this;
}
```



```
[(base) MacBook-Pro:Lab7_ExE carlsoriano$ ls                                  ]
lab7ExE2.cpp
[(base) MacBook-Pro:Lab7_ExE carlsoriano$ ls                                  ]
lab7ExE2.cpp            simpleVector.cpp        simpleVector.h
[(base) MacBook-Pro:Lab7_ExE carlsoriano$ g++ lab7ExE2.cpp simpleVector.cpp simpl]
eVector.h
clang: warning: treating 'c-header' input as 'c++-header' when in C++ mode, this
 behavior is deprecated [-Wdeprecated]
[(base) MacBook-Pro:Lab7_ExE carlsoriano$ ./a.out                             ]
Object v1 is expected to display: 45 69 12
45 69 12
Object v2 is expected to diaplay: 3000 6000 7000 8000
3000 6000 7000 8000

After two calls to at v1 is expected to display: 1000 2000 12:
1000 2000 12

v2 expected to display: 3000 6000 7000 8000 21 28
3000 6000 7000 8000 21 28

After copy v2 is expected to display: 1000 2000 12
1000 2000 12

v1 is expected to display: 1000 2000 8000
1000 2000 8000

v3 is expected to diplay: 1000 2000 12
1000 2000 12

v2 is expected to display: -333 2000 12
-333 2000 12

v4 is expected to diplay: 1000 2000 8000
1000 2000 8000

v1 after self-copy is expected to diplay: -1000 2000 8000
-1000 2000 8000

v1 after chain-copy is expected to diplay: 1000 2000 12
1000 2000 12

v2 after chain-copy is expected to diplay: 1000 2000 12
1000 2000 12
(base) MacBook-Pro:Lab7_ExE carlsoriano$ 
```