Course: ENSF 614 - Fall 2023

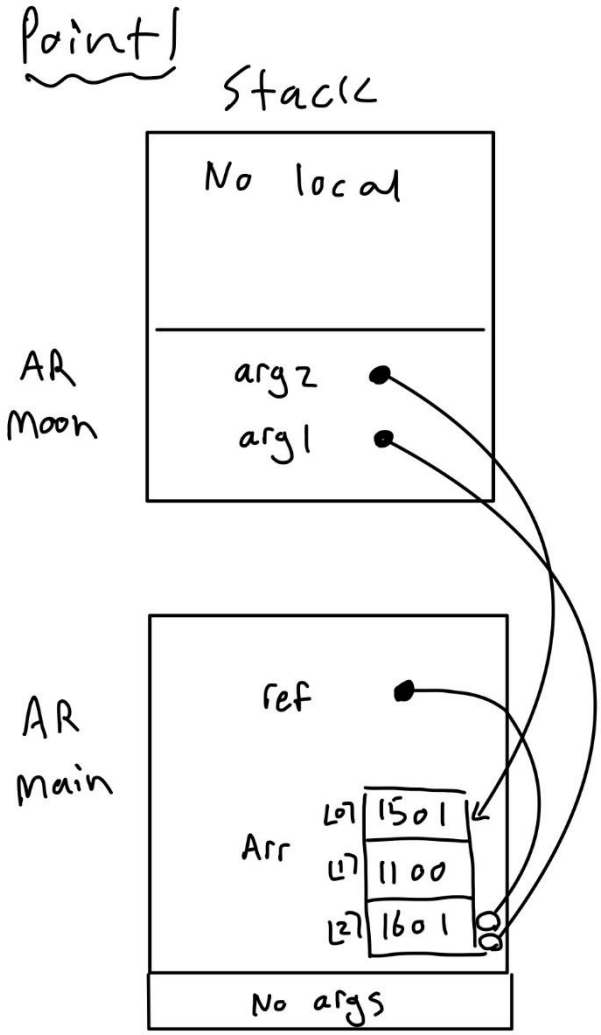
Lab #: Lab 3

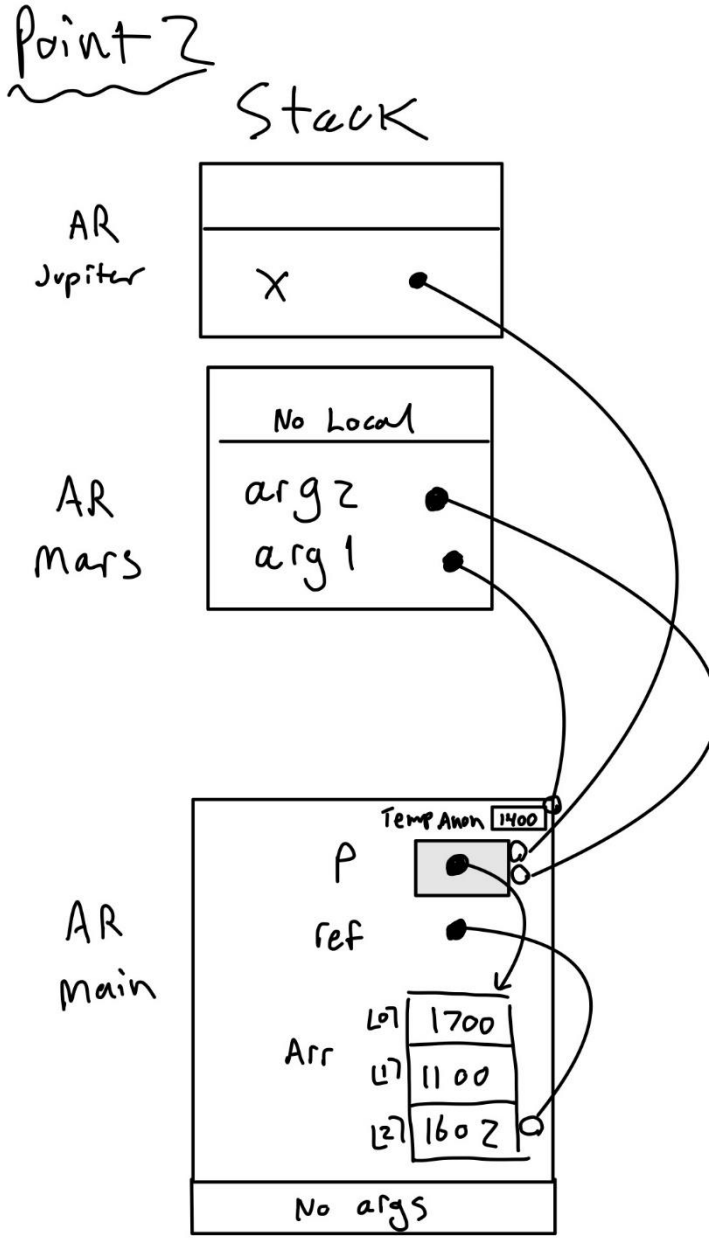
Instructor: Prof. M. Moussavi

Student Name: Jeremy Sugimoto

Submission Date: Oct 13, 2023

Exercise A:

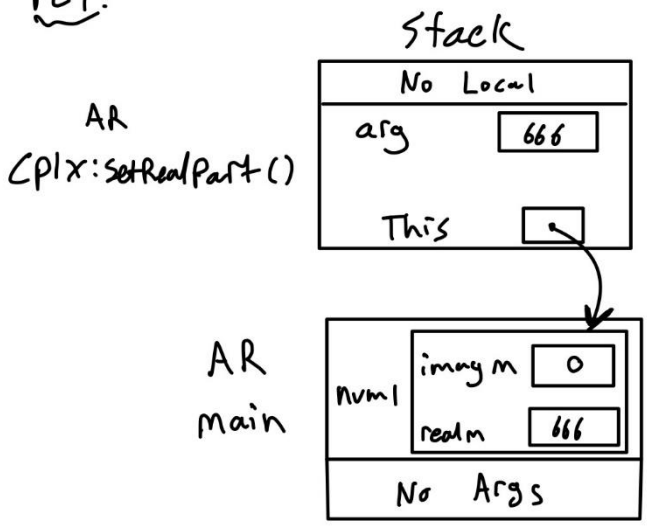

Point1
Stack
No local
AR
Moon
arg2
arg1
AR
main
ref
Arr
[0] 1501
[1] 1100
[2] 1601
No args


Point2
Stack
AR
Jupiter
X
No Local
AR
Mars
arg2
arg1
Temp Anon 1400
p
ref
AR
main
Arr
[0] 1700
[1] 1100
[2] 1602
No args

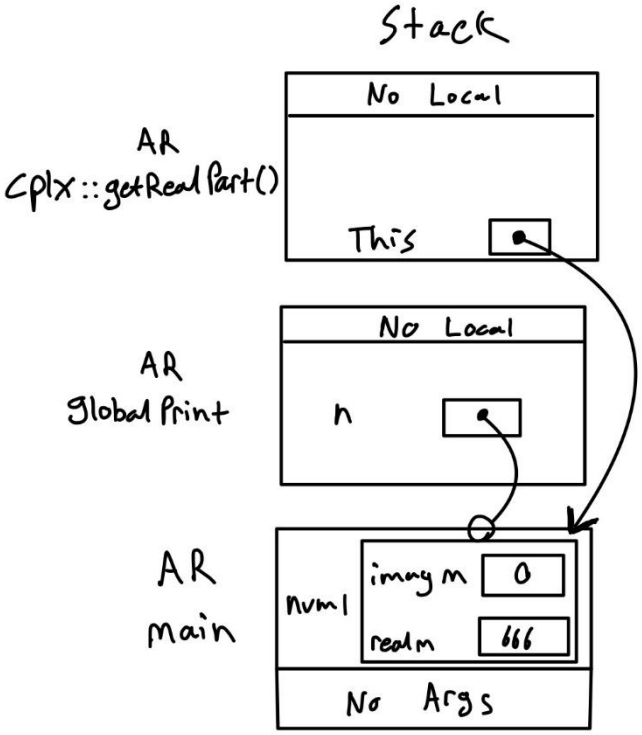
Exercise B:

Exercise B:

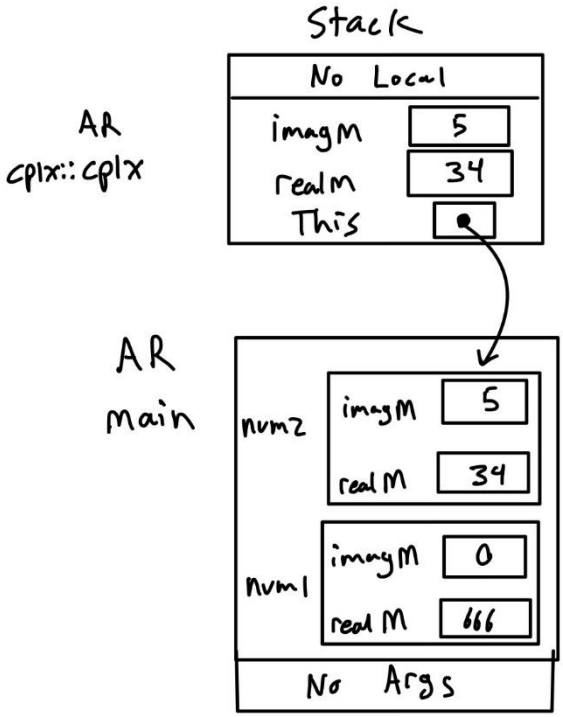
Pt 1:



Pt 2:



Pt 3:



Exercise C:

```
//lab3Clock.h
// ENSF 614 FALL 2023 LAB 3 - EXERCISE C
#ifndef LAB3CLOCK_H
#define LAB3CLOCK_H

class Clock{
private:
    int hour;
    int minute;
    int second;

    // Private Helper functions
    int hms_to_sec(int h, int m, int s);
    void sec_to_hms(int seconds);

public:
    // Constructor
    Clock();
    Clock(int s);
    Clock(int h, int m, int s);

    // Member Functions

    // Setter Funcs
    void set_time(int h, int m, int s);
    void set_hour(int h);
    void set_minute(int m);
    void set_second(int s);

    // Getter Funcs
    int get_hour() const;
    int get_minute() const;
    int get_second() const;

    // Additonal Funcs
    void increment();
    void decrement();
    void add_seconds(int seconds);

    // Display
    //void displayTime();

};


#endif
```

```cpp
//Lab3Clock.cpp
// ENSF 614 FALL 2023 LAB 3 - EXERCISE C

#include "lab3Clock.h"
#include <iostream>
using namespace std;

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

    }

    Clock::Clock(int s){
        if(59 < s || s < 0){
            second = 0;
            minute = 0;
            hour = 0;
        }
        else{
            sec_to_hms(s);
        }

    }

    Clock::Clock(int h, int m, int s){
        set_time(h,m,s);
    }

    void Clock::set_time(int h, int m, int s){
        if(59 < s || s < 0 || 59 < m || m < 0 || 23 < h || h <0){
            second = 0;
            minute = 0;
            hour = 0;
        }

        else{
            second = s;
            minute = m;
            hour = h;
        }

    }


    void Clock::set_second(int s){
        if(59 >= s && s > -1){
            second = s;
        }
    }

    void Clock::set_minute(int m){
        if(59 >= m && m > -1){
            minute = m;
```

```
    }
}

void Clock::set_hour(int h){
    if (23 >= h && h > -1){
        hour = h;
    }

}

int Clock::get_second() const{
    return second;
}

int Clock::get_minute() const{
    return minute;
}

int Clock::get_hour() const{
    return hour;
}

// Additonal Funcs
void Clock::decrement(){
    second--;
    if(second < 0){
        second = 59;
        minute--;
    }
    if(minute < 0){
        minute = 59;
        hour--;
    }
    if(hour < 0){
        hour = 23;
    }

}

void Clock::increment(){
    second++;
    if(second > 59){
        second = 0;
        minute++;
    }
    if(minute > 59){
        minute= 0;
        hour++;
    }
    if(hour > 23){
        hour = 0;
    }
```

```
}

void Clock::add_seconds(int seconds){
    second += (seconds % 60);
    if(second > 59){
        second = 0;
        minute++;
    }
    minute += (seconds % 3600 / 60);
    if(minute > 59){
        minute = 0;
        hour++;
    }
     hour += (seconds / 3600);
    if(hour > 23){
        hour %= 24;
    }
    }

int Clock::hms_to_sec(int h, int m, int s){
    int seconds;
    seconds = (h*3600) + (m*60) + s;

    return seconds;
}
void Clock::sec_to_hms(int seconds){
    hour = seconds / 3600;
    if (hour > 23){
        hour %= 24;
    }
    seconds %= 3600;
    minute = seconds / 60;
    second = seconds % 3600;
}
```

Exercise C Output:

```
Jeremy Sugimoto@DESKTOP-O7EHS1S /cygdrive/c/Users/Jeremy Sugimoto/OneDrive - Uni
versity of Calgary/ENSF 614 Adv Syst Analysis and Soft Design/Lab 3
$ ./a.exe
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
00:00:00
Object t2 decremented by 6 seconds. Expected time is: 23:59:59
23:59:54
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
23:20:54
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
```

Exercise D Output:

```
Jeremy Sugimoto@DESKTOP-O7EHS1S /cygdrive/c/Users/Jeremy Sugimoto/OneDrive - University
$ ./a.exe
Elements of a:  0.5 1.5 2.5 3.5 4.5
(Expected:      0.5 1.5 2.5 3.5 4.5)

Elements of b after first resize:  10.5 11.5 12.5 13.5 14.5 15.5 16.5
(Expected:                         10.5 11.5 12.5 13.5 14.5 15.5 16.5)

Elements of b after second resize:  10.5 11.5 12.5
(Expected:                          10.5 11.5 12.5)

Elements of b after copy ctor check:  10.5 11.5 12.5
(Expected:                            10.5 11.5 12.5)

Elements of c after copy ctor check:  -1.5 11.5 12.5
(Expected:                            -1.5 11.5 12.5)

Elements of a after operator = check:  -10.5 1.5 2.5 3.5 4.5
(Expected:                             -10.5 1.5 2.5 3.5 4.5)

Elements of b after operator = check:  -11.5 1.5 2.5 3.5 4.5
(Expected:                             -11.5 1.5 2.5 3.5 4.5)

Elements of c after operator = check:  0.5 1.5 2.5 3.5 4.5
(Expected:                             0.5 1.5 2.5 3.5 4.5)
```