**Course: ENSF 614** - Fall 2023  
**Lab #:** Lab 3  
**Instructor:** Mahmood Moussavi  
**Student Names:** Alton Wong, Christian Valdez  
**Submission Date:** October 13, 2023

**Exercise A**

Point 1

A diagram of a graph

Description automatically generated with medium confidence

Point 2

A diagram of a line

Description automatically generated

**Exercise B**

**A diagram of a wireframe

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Exercise C**

//

// lab3Clock.h

// ENSF 614 - Lab 3 - Exercise C

// Section B01

// Completed by: Alton Wong and Christian Valdez

// Submission Date: October 13, 2023

//

#ifndef lab3Clock\_h

#define lab3Clock\_h

/\*The following class represents a 24-hour clock and contains three private integer data members called hour, minute, and second.

\*/

**class** Clock{

**private**:

**int** hour;

**int** minute;

**int** second;

**int** hms\_to\_sec();

**void** sec\_to\_hms(**int** seconds\_total);

**public**:

Clock();

//PROMISES: initializes the data-members: hour, minute, second to zero

Clock(**int** seconds\_total);

//PROMISES: initializes the data-members based on total seconds

Clock(**int** h, **int** m, **int** s);

//PROMISES: initializes the data-members with the supplied values for hour, minute, and second

**int** get\_hour() **const**;

//PROMISES: returns hour data member of the clock

**int** get\_minute() **const**;

//PROMISES: returns minute data member of the clock

**int** get\_second() **const**;

//PROMISES: returns minute data member of the clock

**void** set\_hour(**int** arg);

//PROMISES: sets a new value to the hour component of clock with the value of arg

**void** set\_minute(**int** arg);

//PROMISES: sets a new value to the minute component of clock with the value of arg

**void** set\_second(**int** arg);

//PROMISES: sets a new value to the second component of clock with the value of arg

**void** increment();

//PROMISES: increments clock by one second

**void** decrement();

//PROMISES: decrements clock by one second

**void** add\_seconds(**int** s);

//PROMISES: adds number of seconds to the clock

};

#endif /\* lab3Clock\_h \*/

//

// lab3Clock.cpp

// ENSF 614 - Lab 3 - Exercise C

// Section B01

// Completed by: Alton Wong and Christian Valdez

// Submissino Date: October 13, 2023

//

#include "lab3Clock.h"

#include <iostream>

**using** **namespace** std;

**int** Clock::hms\_to\_sec(){

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

}

**void** Clock::sec\_to\_hms(**int** seconds\_total){

seconds\_total %= 86400; // Wrap around every 24 hours (24 \* 60 \* 60 = 86400 seconds)

hour = seconds\_total / 3600;

seconds\_total %= 3600;

minute = seconds\_total / 60;

second = seconds\_total % 60;

}

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

}

Clock::Clock(**int** seconds\_total){

**if** (seconds\_total < 0){

hour = 0;

minute = 0;

second = 0;

}

**else**{

sec\_to\_hms(seconds\_total);

}

}

Clock::Clock(**int** h, **int** m, **int** s){

**if**((h > 23 || h < 0)||(m > 59 || m < 0)||(s > 59 || s < 0)){

hour = 0;

minute = 0;

second = 0;

}

**else**{

hour = h;

minute = m;

second = s;

}

}

**int** Clock::get\_hour() **const**{

**return** hour;

}

**int** Clock::get\_minute() **const**{

**return** minute;

}

**int** Clock::get\_second() **const**{

**return** second;

}

**void** Clock::set\_hour(**int** arg){

**if**(arg < 23 && arg > 0){

hour = arg;

}

}

**void** Clock::set\_minute(**int** arg){

**if**(arg < 59 && arg > 0){

minute = arg;

}

}

**void** Clock::set\_second(**int** arg){

**if**(arg < 59 && arg > 0){

second = arg;

}

}

**void** Clock::increment(){

second++;

**if**(second == 60){

second = 0;

minute++;

**if**(minute == 60){

minute = 0;

hour = (hour + 1) % 24;

}

}

}

**void** Clock::decrement(){

second--;

**if**(second < 0){

second = 59;

minute--;

**if**(minute < 0){

minute = 59;

hour--;

**if**(hour < 0){

hour = 23;

}

}

}

}

**void** Clock:: add\_seconds(**int** s){

**if** (s<0) **return**;

**int** total\_seconds = hms\_to\_sec() + s;

sec\_to\_hms(total\_seconds);

}

**Sample output**

**A screenshot of a computer

Description automatically generated**

**Exercise D**

**Sample output:**

**A screenshot of a computer program

Description automatically generated**