ENSF 614

Advanced System Analysis and Software Design

LAB 3

Author:

Steven Duong
(30022492)

Affiliation
Department of Electrical and Software Engineering
University of Calgary
Calgary, Alberta

Lab Block: B01
Date of Report: Feb 4, 2023

Exercise C – Header File lab3Clock.h

```
/*
 *  File Name: lab3Clock.h
 *  Assignment: ENSF 614 Lab 3 Exercise C
 *  Lab Section: Lab B01
 *  Completed by: Steven Duong (30022492)
 *  Submission Date: Feb 4, 2023
 */

#ifndef LAB_3_LAB3CLOCK_H
#define LAB_3_LAB3CLOCK_H

class Clock {

  private: int hour,
  minute,
  second;

  void sec_to_hms(int seconds);
  // Converts total seconds into hours, minutes and seconds

  // REQUIRES:
  //      hour >= 0 and hour <= 23
  //      minute >= 0 and minute <= 59
  //      seconds >= 0 and seconds <= 59
  //
  // PROMISES:
  //      Initialize all member variables of the class to be within
the range required above

  int hms_to_sec() const;
  // Converts hours, minutes and seconds into a total number of
seconds

  // REQUIRES:
  //      NONE
  //
  // PROMISES:
  //      Return the total time as seconds

  public: Clock();
  // Default constructor which initializes all objects to 0

  // REQUIRES:
  //      NONE
  //
  // PROMISES:
  //      Initialize all member variables of the class to 0
```

```
  Clock(int seconds);
  // Class constructor which constructs a clock object with
  // the time value in seconds

  // REQUIRES:
  //      seconds >= 0
  //
  // PROMISES:
  //      An object of type Clock with a time value in seconds

  Clock(int hour, int minute, int seconds);
  // Class constructor which creates a Clock object containing
  // the hours, minutes and seconds

  // REQUIRES:
  //      hour >= 0 and hour <= 23
  //      minute >= 0 and minute <= 59
  //      seconds >= 0 and seconds <= 59
  //
  // PROMISES:
  //      An object of type Clock with a time in hours, minutes and
seconds

  int get_hour() const;
  // Getter method for the hour member variable

  // REQUIRES:
  //      NONE
  //
  // PROMISES:
  //      Return the hour of the Clock object

  int get_minute() const;
  // Getter method for the minute member variable

  // REQUIRES:
  //      NONE
  //
  // PROMISES:
  //      Return the minute of the Clock object

  int get_second() const;
  // Getter method for the second member variable

  // REQUIRES:
  //      NONE
  //
  // PROMISES:
  //      Return the second of the Clock object
```

```
void set_hour(int hour);
// Setter method for the hour member variable

// REQUIRES:
//      hour >= 0 and hour <= 23
//
// PROMISES:
//      Set the hour to the value that is passed in the parameter

void set_minute(int minute);
// Setter method for the minute member variable

// REQUIRES:
//      minute >= 0 and minute <= 59
//
// PROMISES:
//      Set the minute to the value that is passed in the parameter

void set_second(int second);
// Setter method for the second member variable

// REQUIRES:
//      second >= 0 and second <= 59
//
// PROMISES:
//      Set the second to the value that is passed in the parameter

void increment();
// Increment the time in the Clock object by one second

// REQUIRES:
//      NONE
//
// PROMISES:
//      Increases the time represented in Clock object by one second

void decrement();
// Decrement the time in the Clock object by one second

// REQUIRES:
//      NONE
//
// PROMISES:
//      Decreases the time represented in Clock object by one second

void add_seconds(int seconds);
// Add seconds to the time represented in a Clock object

// REQUIRES:
```

```
  //      NONE
  //
  // PROMISES:
  //      Total time is increased by the number of seconds passed in
the parameter

};

#endif //LAB_3_LAB3CLOCK_H
```

Exercise C – Source File lab3Clock.cpp

```
/*
 *  File Name: lab3Clock.cpp
 *  Assignment: ENSF 614 Lab 3 Exercise C
 *  Lab Section: Lab B01
 *  Completed by: Steven Duong (30022492)
 *  Submission Date: Feb 4, 2023
 */

#include "lab3Clock.h"

Clock::Clock() {
  hour = 0;
  minute = 0;
  second = 0;
}

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

}

Clock::Clock(int hour, int minute, int second) {
  if (second > 59 || second < 0 || minute > 59 || minute < 0 || hour >
23 || hour < 0) {
    this -> hour = this -> minute = this -> second = 0;
  } else {
    this -> hour = hour;
    this -> minute = minute;
    this -> second = second;
  }
}
```

```cpp
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 hours) {
  if (hours >= 0 && hours <= 23) {
    this -> hour = hours;
  }
}

void Clock::set_minute(int minutes) {
  if (minutes >= 0 && minutes <= 59) {
    this -> minute = minutes;
  }
}

void Clock::set_second(int seconds) {
  if (seconds >= 0 && seconds <= 59) {
    this -> second = seconds;
  }
}

void Clock::sec_to_hms(int seconds) {
  int h = (seconds / 3600) % 24;
  int m = (h % 60) - h;
  int s = seconds % 60;

  set_hour(h);
  set_minute(m);
  set_second(s);
}

int Clock::hms_to_sec() const {
  return (get_hour() * 3600 + get_minute() * 60 + get_second());
}

void Clock::increment() {
  sec_to_hms(hms_to_sec() + 1);
}

void Clock::decrement() {
  if (second > 0) {
```

```
    second--;
  } else {
    set_second(59);
    if (minute > 0) {
      minute--;
    } else {
      set_minute(59);
      if (hour > 0) {
        hour--;
      } else {
        set_hour(23);
      }
    }
  }
}

void Clock::add_seconds(int seconds) {
  sec_to_hms(hms_to_sec() + seconds);
}
```

Exercise 3 – Program Output

```
"/Users/stevenduong/CLionProjects/ENSF 614/Labs/Lab 3/cmake-build-debug/Lab_3"
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

Process finished with exit code 0
```