Khoi Duong

Prof. Yang

CS360

7/26/2022

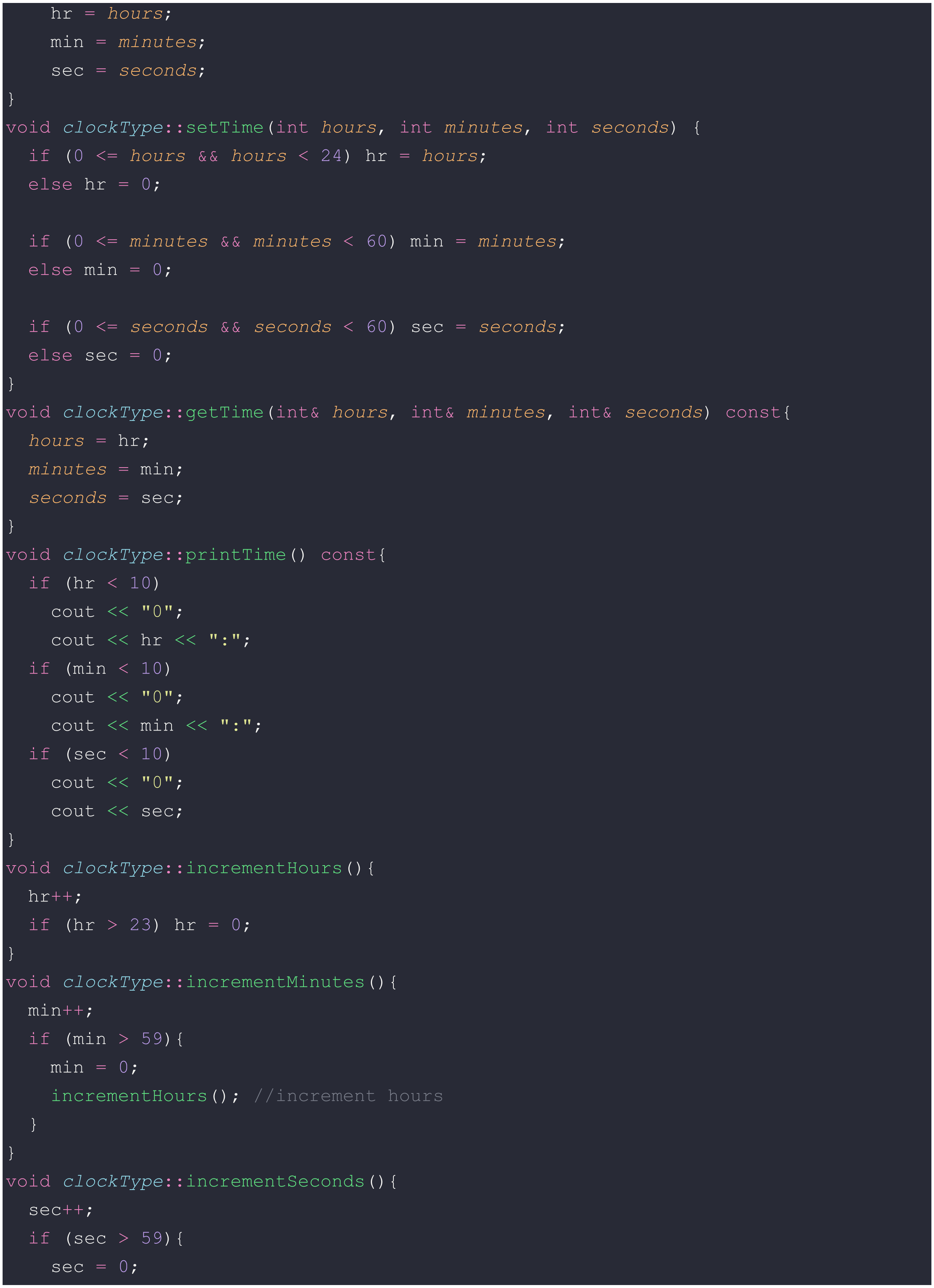
QUIZ#2

clockType.h

```
#ifndef CLOCKTYPE_H
#define CLOCKTYPE_H
#include <iostream>

class clockType{
  public:
    clockType(int hours, int minutes, int seconds);
    void setTime(int, int, int);
    void getTime(int&, int&, int&) const;
    void printTime() const;
    void incrementSeconds();
    void incrementMinutes();
    void incrementHours();
    bool equalTime(const clockType&) const;
  private:
    int hr;
    int min;
    int sec;
};
#endif
```

clockType.cpp

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

clockType::clockType(int hours, int minutes, int seconds)
{
```



```
    hr = hours;
    min = minutes;
    sec = seconds;
}
void clockType::setTime(int hours, int minutes, int seconds) {
  if (0 <= hours && hours < 24) hr = hours;
  else hr = 0;

  if (0 <= minutes && minutes < 60) min = minutes;
  else min = 0;

  if (0 <= seconds && seconds < 60) sec = seconds;
  else sec = 0;
}
void clockType::getTime(int& hours, int& minutes, int& seconds) const{
  hours = hr;
  minutes = min;
  seconds = sec;
}
void clockType::printTime() const{
  if (hr < 10)
    cout << "0";
    cout << hr << ":";
  if (min < 10)
    cout << "0";
    cout << min << ":";
  if (sec < 10)
    cout << "0";
    cout << sec;
}
void clockType::incrementHours(){
  hr++;
  if (hr > 23) hr = 0;
}
void clockType::incrementMinutes(){
  min++;
  if (min > 59){
    min = 0;
    incrementHours(); //increment hours
  }
}
void clockType::incrementSeconds(){
  sec++;
  if (sec > 59){
    sec = 0;
    incrementMinutes(); //increment minutes
  }
}
bool clockType::equalTime(const clockType& otherClock) const {
  return (hr == otherClock.hr && min == otherClock.min && sec ==
otherClock.sec);
}
```

extclockType.h

```
#ifndef EXTCLOCKTYPE_H
#define EXTCLOCKTYPE_H

#include <string>

#include "clockType.h"

using namespace std;

class extClockType: public clockType
{
public:
void setTime(int hours, int minutes, int seconds, string tZone);
void printTime();

extClockType(int = 0, int = 0, int = 0, string = "EST");

private:
string timeZone;
};

#endif
```

extclockType.cpp

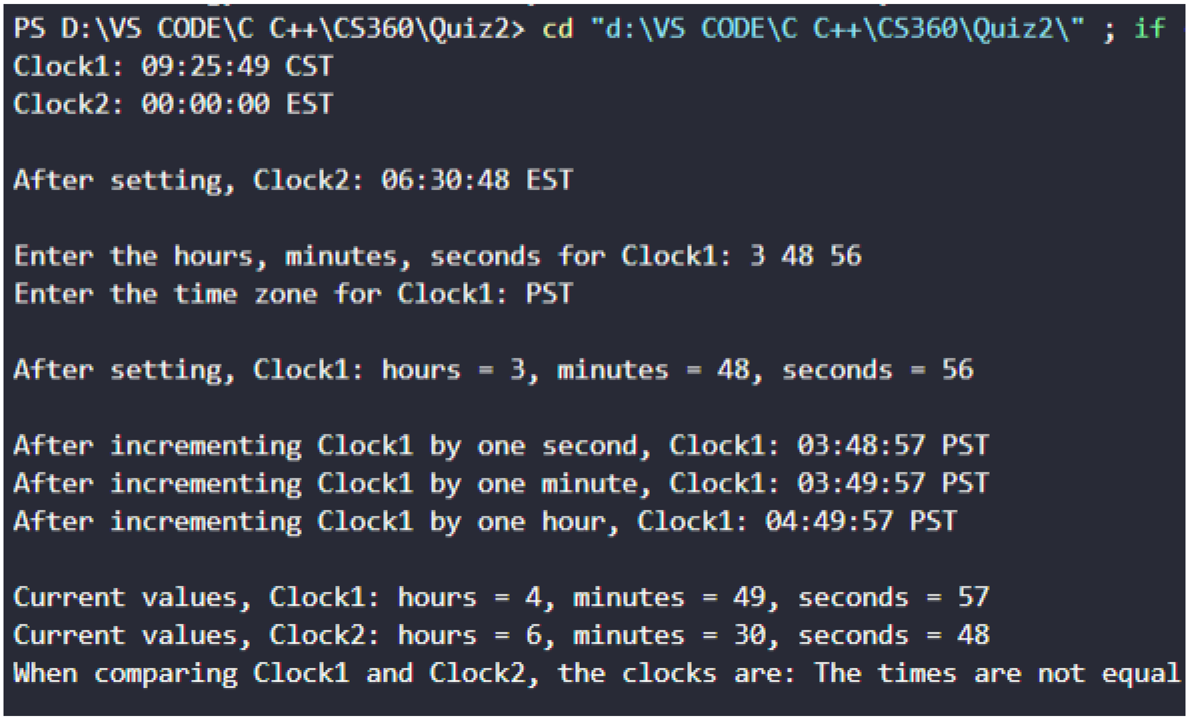
```
#include <iostream>
#include <string>

#include "extClockType.h"
#include "clockType.h"
#include "clockType.cpp"
using namespace std;
```

```
void extClockType::setTime(int hours, int minutes, int seconds, string tZone)
{
clockType::setTime(hours, minutes, seconds);
timeZone = tZone;
}

void extClockType::printTime()
{
clockType::printTime();
cout << " " << timeZone;
}

extClockType::extClockType(int hours, int minutes, int seconds, string tZone) :
clockType(hours, minutes, seconds)
{
timeZone=tZone;
}
```

main.cpp

```
#include <string>
#include <iomanip>
#include <iostream>
#include "extClockType.h"
#include "extclockType.cpp"

using namespace std;

int main()
{
    extClockType Clock1;
    extClockType Clock2;
    int hours, minutes, seconds;
    string timeZone;
    Clock1.setTime(9, 25, 49, "CST");
    cout << "Clock1: ";
    Clock1.printTime();
    cout << endl;
    cout << "Clock2: ";
    Clock2.printTime();
    cout << endl << endl;
    Clock2.setTime(6, 30, 48, "EST");
    cout << "After setting, Clock2: ";
    Clock2.printTime();
    cout << endl << endl;
    cout << "Enter the hours, minutes, seconds for Clock1: ";
    cin >> hours >> minutes >> seconds;
    cout << "Enter the time zone for Clock1: ";
    cin >> timeZone;
    cout << endl;
    Clock1.setTime(hours, minutes, seconds, timeZone);
    cout << "After setting, Clock1: hours = " << hours
        << ", minutes = " << minutes << ", seconds = " << seconds << endl << endl;
    cout << "After incrementing Clock1 by one second, Clock1: ";
    Clock1.incrementSeconds();
    Clock1.printTime();
    cout << endl;
    cout << "After incrementing Clock1 by one minute, Clock1: ";
    Clock1.incrementMinutes();
    Clock1.printTime();
    cout << endl;
    cout << "After incrementing Clock1 by one hour, Clock1: ";
    Clock1.incrementHours();
    Clock1.printTime();
    cout << endl << endl;
    Clock1.getTime(hours, minutes, seconds);
    cout << "Current values, Clock1: hours = " << hours
        << ", minutes = " << minutes << ", seconds = " << seconds << endl;
    Clock2.getTime(hours, minutes, seconds);
    cout << "Current values, Clock2: hours = " << hours
        << ", minutes = " << minutes << ", seconds = " << seconds << endl;
    cout << "When comparing Clock1 and Clock2, the clocks are: ";
    if (Clock1.equalTime(Clock2))
    cout << "The times are equal" << endl << endl;
    else
    cout << "The times are not equal" << endl << endl;

    return 0;

}
```

Run program & result:



```
PS D:\VS CODE\C C++\CS360\Quiz2> cd "d:\VS CODE\C C++\CS360\Quiz2\" ; if
Clock1: 09:25:49 CST
Clock2: 00:00:00 EST

After setting, Clock2: 06:30:48 EST

Enter the hours, minutes, seconds for Clock1: 3 48 56
Enter the time zone for Clock1: PST

After setting, Clock1: hours = 3, minutes = 48, seconds = 56

After incrementing Clock1 by one second, Clock1: 03:48:57 PST
After incrementing Clock1 by one minute, Clock1: 03:49:57 PST
After incrementing Clock1 by one hour, Clock1: 04:49:57 PST

Current values, Clock1: hours = 4, minutes = 49, seconds = 57
Current values, Clock2: hours = 6, minutes = 30, seconds = 48
When comparing Clock1 and Clock2, the clocks are: The times are not equal
```