11.1P Clock in C++

clock.cpp

```
#include "Clock.h"

#include <iomanip>

#include <sstream>


Clock::Clock() : _hours("hours"), _minutes("minutes"), _seconds("seconds") {}


void Clock::Tick() {
   _seconds.Increment();
   if (_seconds.GetTick() > 59) {
      _seconds.Reset();
      _minutes.Increment();
      if (_minutes.GetTick() > 59) {
         _minutes.Reset();
         _hours.Increment();
         if (_hours.GetTick() > 23) {
            ResetTime();
         }
      }
   }
}


void Clock::ResetTime() {
   _hours.Reset();
   _minutes.Reset();
   _seconds.Reset();
}
```

```
std::string Clock::CurrentTime() const {
   std::ostringstream timeStream;
   timeStream << std::setw(2) << std::setfill('0') << _hours.GetTick() << ":"
      << std::setw(2) << std::setfill('0') << _minutes.GetTick() << ":"
      << std::setw(2) << std::setfill('0') << _seconds.GetTick();
   return timeStream.str();
}
```

counter.cpp

```
#include "Counter.h"

Counter::Counter(std::string name, int count) : _name(name), _count(count) {}

void Counter::Increment() {
   _count++;
}

void Counter::Reset() {
   _count = 0;
}

int Counter::GetTick() const {
   return _count;
}
```

Main.cpp

```
#include <iostream>
#include <thread>
#include <chrono>
```

```
#include "Clock.h"

int main() {
   Clock clock;

   for (int i = 0; i < 86400; ++i) { // Simulate a full day (86400 seconds)
      std::this_thread::sleep_for(std::chrono::seconds(1)); // Wait for 1 second
      system("CLS"); // Clear the console (works on Windows)
      clock.Tick();
      std::cout << clock.CurrentTime() << std::endl;
   }

   return 0;
}
```

Header file
clock.h

```
#pragma once
#include "Counter.h"
#include <string>

class Clock {
private:
   Counter _hours;
   Counter _minutes;
   Counter _seconds;

public:
   Clock();
```

```
    void Tick();
    void ResetTime();
    std::string CurrentTime() const;
};
```

Counter.h

```
#pragma once
#include <string>

class Counter {
private:
    int _count;
    std::string _name;

public:
    Counter(std::string name, int count = 0);

    void Increment();
    void Reset();
    int GetTick() const;
};
```

Outcome