

```
1 #include <string>
2 #include <sstream>
3 #include <iomanip>
4
5 using namespace std;
6
7 class Counter {
8 private:
9     int _count;
10    string _name;
11
12 public:
13    Counter(const string& name) : _count(0), _name(name) {}
14
15    void Increment() {
16        ++_count;
17    }
18
19    void Reset() {
20        _count = 0;
21    }
22
23    string Name() const {
24        return _name;
25    }
26
27    void SetName(const string& value) {
28        _name = value;
29    }
30
31    int Ticks() const {
32        return _count;
33    }
34 };
35
36 class Clock {
37 private:
38    Counter _seconds;
39    Counter _minutes;
40    Counter _hours;
41
42 public:
43    Clock() : _seconds("seconds"), _minutes("minutes"), _hours("hours") {}
44
45    void Tick() {
46        _seconds.Increment();
47        if (_seconds.Ticks() > 59) {
48            _minutes.Increment();
49            _seconds.Reset();
```

```
50         if (_minutes.Ticks() > 59) {
51             _hours.Increment();
52             _minutes.Reset();
53             if (_hours.Ticks() > 23) {
54                 Reset();
55             }
56         }
57     }
58 }
59
60 void Reset() {
61     _seconds.Reset();
62     _minutes.Reset();
63     _hours.Reset();
64 }
65
66 string Time() const {
67     ostringstream oss;
68     oss << setw(2) << setfill('0') << _hours.Ticks() << ":"
69         << setw(2) << setfill('0') << _minutes.Ticks() << ":"
70         << setw(2) << setfill('0') << _seconds.Ticks();
71     return oss.str();
72 }
73 };
74
```

```
1 #include <string>
2 #include <sstream>
3 #include <iomanip>
4
5 using namespace std;
6
7 class Counter {
8 private:
9     int _count;
10    string _name;
11
12 public:
13    Counter(const string& name) : _count(0), _name(name) {}
14
15    void Increment() {
16        ++_count;
17    }
18
19    void Reset() {
20        _count = 0;
21    }
22
23    string Name() const {
24        return _name;
25    }
26
27    void SetName(const string& value) {
28        _name = value;
29    }
30
31    int Ticks() const {
32        return _count;
33    }
34 };
35
36 class Clock {
37 private:
38    Counter _seconds;
39    Counter _minutes;
40    Counter _hours;
41
42 public:
43    Clock() : _seconds("seconds"), _minutes("minutes"), _hours("hours") {}
44
45    void Tick() {
46        _seconds.Increment();
47        if (_seconds.Ticks() > 59) {
48            _minutes.Increment();
49            _seconds.Reset();
```

```
50         if (_minutes.Ticks() > 59) {
51             _hours.Increment();
52             _minutes.Reset();
53             if (_hours.Ticks() > 23) {
54                 Reset();
55             }
56         }
57     }
58 }
59
60 void Reset() {
61     _seconds.Reset();
62     _minutes.Reset();
63     _hours.Reset();
64 }
65
66 string Time() const {
67     ostringstream oss;
68     oss << setw(2) << setfill('0') << _hours.Ticks() << ":"
69         << setw(2) << setfill('0') << _minutes.Ticks() << ":"
70         << setw(2) << setfill('0') << _seconds.Ticks();
71     return oss.str();
72 }
73 };
74
```

```
1 #include <string>
2 #include <sstream>
3 #include <iomanip>
4
5 using namespace std;
6
7 class Counter {
8 private:
9     int _count;
10    string _name;
11
12 public:
13    Counter(const string& name) : _count(0), _name(name) {}
14
15    void Increment() {
16        ++_count;
17    }
18
19    void Reset() {
20        _count = 0;
21    }
22
23    string Name() const {
24        return _name;
25    }
26
27    void SetName(const string& value) {
28        _name = value;
29    }
30
31    int Ticks() const {
32        return _count;
33    }
34 };
35
36 class Clock {
37 private:
38    Counter _seconds;
39    Counter _minutes;
40    Counter _hours;
41
42 public:
43    Clock() : _seconds("seconds"), _minutes("minutes"), _hours("hours") {}
44
45    void Tick() {
46        _seconds.Increment();
47        if (_seconds.Ticks() > 59) {
48            _minutes.Increment();
49            _seconds.Reset();
```

```
50         if (_minutes.Ticks() > 59) {
51             _hours.Increment();
52             _minutes.Reset();
53             if (_hours.Ticks() > 23) {
54                 Reset();
55             }
56         }
57     }
58 }
59
60 void Reset() {
61     _seconds.Reset();
62     _minutes.Reset();
63     _hours.Reset();
64 }
65
66 string Time() const {
67     ostringstream oss;
68     oss << setw(2) << setfill('0') << _hours.Ticks() << ":"
69         << setw(2) << setfill('0') << _minutes.Ticks() << ":"
70         << setw(2) << setfill('0') << _seconds.Ticks();
71     return oss.str();
72 }
73 };
74
```

00:04:42
00:04:43
00:04:44
00:04:45
00:04:46
00:04:47
00:04:48
00:04:49
00:04:50
00:04:51
00:04:52
00:04:53
00:04:54
00:04:55
00:04:56
00:04:57
00:04:58
00:04:59
00:05:00
00:05:01
00:05:02
00:05:03
00:05:04
00:05:05
00:05:06
00:05:07
00:05:08
00:05:09
00:05:10
00:05:11
00:05:12
00:05:13
00:05:14
00:05:15

PS D:\workspace\COS20007\COS20007-working\11.1P - Clock in Another Language> █