

演習

演習 1a

- ▶ 以下のプログラムを実行してみよ。
- ▶ コメントにあるファイル名を使う。
 - ▶ 以降の問題も同様

```
#include <iostream> // ex02-1a.cpp
class Loud {
public:
    Loud() { std::cout << "Hello!!\n"; }
};

int main()
{
    Loud a; // Hello!!
}
```

演習 1b

- ▶ 以下のプログラムを実行してみよ。

```
#include <iostream> // ex02-1b.cpp
class TimeData6 {
public:
    int min{}, sec{};
    TimeData6(int m, int s) {
        min = m + s/60;
        sec = s%60;
    }
};

int main()
{
    TimeData6 t{3, 50};
    std::cout << t.min << ":" << t.sec << "\n"; // 3:50
}
```

演習 1c

▶ 以下のプログラムを実行してみよ。

```
#include <iostream> // ex02-1c.cpp
class TimeData9 {
    int min{}, sec{};
public:
    TimeData9(int m, int s): min(m), sec(s){}
    int getseconds() const { return min*60 + sec; }
    bool operator<(const TimeData9& y) const {
        return getseconds() < y.getseconds();
    }
};

int main()
{
    TimeData9 a{2, 19}, b{3, 59};
    if (a < b)
        std::cout <<"a is small\n";
}
```

演習 1d

- ▶ 以下のプログラムを実行してみよ。

```
#include <iostream>    // ex02-1d.cpp
#include <iomanip>
class TimeData11 {
    int min{}, sec{};
public:
    TimeData11(int m, int s): min{m+s/60}, sec{s%60}{}
    int getm() const { return min; }
    int gets() const { return sec; }
    std::string str() const {
        std::ostringstream o;
        o << min << ":" << std::setw(2)
          << std::setfill('0') << sec;
        return o.str();
    }
};
```

```

// ex02-1d.cpp つづき
auto
operator+(const TimeData11& a, const TimeData11& b) {
    return TimeData11{a.getm()+b.getm(),
                      a.gets()+b.gets()};
}

TimeData11
operator-(const TimeData11& a, const TimeData11& b) {
    return {a.getm()-b.getm(), a.gets()-b.gets()};
}

auto&
operator<<(std::ostream& out, const TimeData11& t){
    return out << t.str();
}

int main() {
    TimeData11 x{3, 35}, y{2, 10};
    std::cout << x+y << "\n"; // 5:45
    std::cout << x-y << "\n"; // 1:25
}

```

演習 2

- ▶ コンパイラに対してデフォルトコンストラクタを生成
依頼するようにクラスを修正せよ。

```
#include <iostream> // ex02-2.cpp
class TimeData8 {
public:
    int min{1}, sec{30};
    // ここに 1 行追加する
    TimeData8(int m, int s) {
        min = m + s/60;
        sec = s%60;
    }
};

int main()
{
    TimeData8 t1{3, 65};
    std::cout << t1.min << ":" << t1.sec << "\n"; // 4:5
    TimeData8 t2;
    std::cout << t2.min << ":" << t2.sec << "\n"; // 1:30
}
```

演習 3

- ▶ 空欄を埋めてプログラムを完成させよ。

```
#include <iostream>    // ex02-3.cpp
class Ref {
    int &x, y;
public:
    // コンストラクタを作る
    void set(int a, int b) { x = a; y = b; }
    void print() const {
        std::cout << x << " " << y << "\n";
    }
};

int main() {
    int n{10};
    Ref s{n, 0}, t{n, 2};
    s.set(3,1);
    s.print(); // 3 1
    t.print(); // 3 2
}
```


演習 4

- ▶ 空欄を埋めてファイル 2 個でプログラムを完成させよ。

```
class TimeData12 { // timedata12.hpp
    int min{}, sec{};
public:
    TimeData12(int m,int s):min{m+s/60},sec{s%60}{}
    int getm() const { return min; }
    int gets() const { return sec; }
};
```

```
#include <iostream> // ex02-4.cpp
#include <iomanip>
#include "timedata12.hpp"
// ここを埋める
int main() {
    TimeData12 x{2, 5}, y{1, 75};
    std::cout << x <<"", " << y <<"\n"; // 2:05, 2:15
}
```

演習 5

- ▶ `timedata13-impl.cpp` を作って以下を完成させよ。
- ▶ 以下の 2 個はそのまま使い変えない。

```
class TimeData13 { // timedata13.hpp
    int min{}, sec{};
public:
    TimeData13(int m, int s);
    std::string str() const;
    bool operator<(const TimeData13&) const;
};
```

```
#include <iostream> // ex02-5.cpp
#include "timedata13.hpp"
int main() {
    TimeData13 x{2, 5}, y{1, 75};
    std::cout <<(x < y?x.str():y.str())<<"\n"; // 2.05
}
```