```
=== 演習1 解答例 ============================
--- compile.txt ---
$ g++ -std=c++17 ex02-1a.cpp
$ ./a.out
Hello!!
$ g++ -std=c++17 ex02-1b.cpp
$ ./a.out
3:50
$ g++ -std=c++17 ex02-1c.cpp
$ ./a.out
a is small
$ g++ -std=c++17 ex02-1d.cpp
$ ./a.out
5:45
1:25


--- ex02-1a.cpp ---
// コンストラクタ
#include <iostream>

class Loud {
  public:
    Loud() { std::cout << "Hello!!\n"; }
};

int main()
{
   Loud a;
}



--- ex02-1b.cpp ---
// コンストラクタ
#include <iostream>

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";
}


--- ex02-1c.cpp ---
// 分秒処理: less than
#include <iostream>
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";
```

```
}


--- ex02-1d.cpp ---
// 分秒処理: add/sub
#include <iostream>
#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();
   }
};

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 解答例 ============================
--- compile.txt ---
$ g++ -std=c++17 ex02-2.cpp
$ ./a.out
4:5
1:30


--- ex02-2.cpp ---
#include <iostream>
class TimeData8 {
public:
   int min{1}, sec{30};
   TimeData8() =default; // デフォルトコンストラクタ
   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";
   TimeData8 t2;
   std::cout << t2.min <<":"<< t2.sec <<"\n";
}
```

```
=== 演習3 解答例 ============================
--- compile.txt ---
$ g++ -std=c++17 ex02-3.cpp
$ ./a.out
3 1
3 2


--- ex02-3.cpp ---
// メンバ初期化リスト
#include <iostream>
class Ref {
    int &x, y;
  public:
    Ref(int& a, int b): x(a),y(b){}
    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 解答例 ============================
--- compile.txt ---
$ g++ -std=c++17 ex02-4.cpp
$ ./a.out
2:05, 2:15


--- timedata12.hpp ---
class TimeData12 {
   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; }
};


--- ex02-4.cpp ---
// 分秒処理:
#include <iostream>
#include <iomanip>
#include "timedata12.hpp"

auto&
operator<<(std::ostream& o, const TimeData12& t)
{
   return o << t.getm() <<":"<< std::setw(2)
            << std::setfill('0')<< t.gets();
}

int main()
{
   TimeData12 x{2, 5}, y{1, 75};
   std::cout << x <<", "<< y <<"\n"; // 2:05, 2:15
}
```

```
=== 演習5 解答例 ============================
--- compile.txt ---
$ g++ -std=c++17 ex02-5.cpp timedata13-impl.cpp
$ ./a.out
2:05



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


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


--- timedata13-impl.cpp ---
#include <sstream>
#include <iomanip>
#include "timedata13.hpp"

TimeData13::TimeData13(int m, int s)
   : min{m+s/60}, sec{s%60} {}

std::string
TimeData13::str() const
{
   std::ostringstream o;
   o << min <<":"<< std::setw(2)
     << std::setfill('0')<< sec;
   return o.str();
}

bool
TimeData13::operator<(const TimeData13& t) const
{
   return min < t.min || (min == t.min && sec < t.sec);
}
```