隣とひとつ以上開けて座ってください
出欠端末で出席登録してください

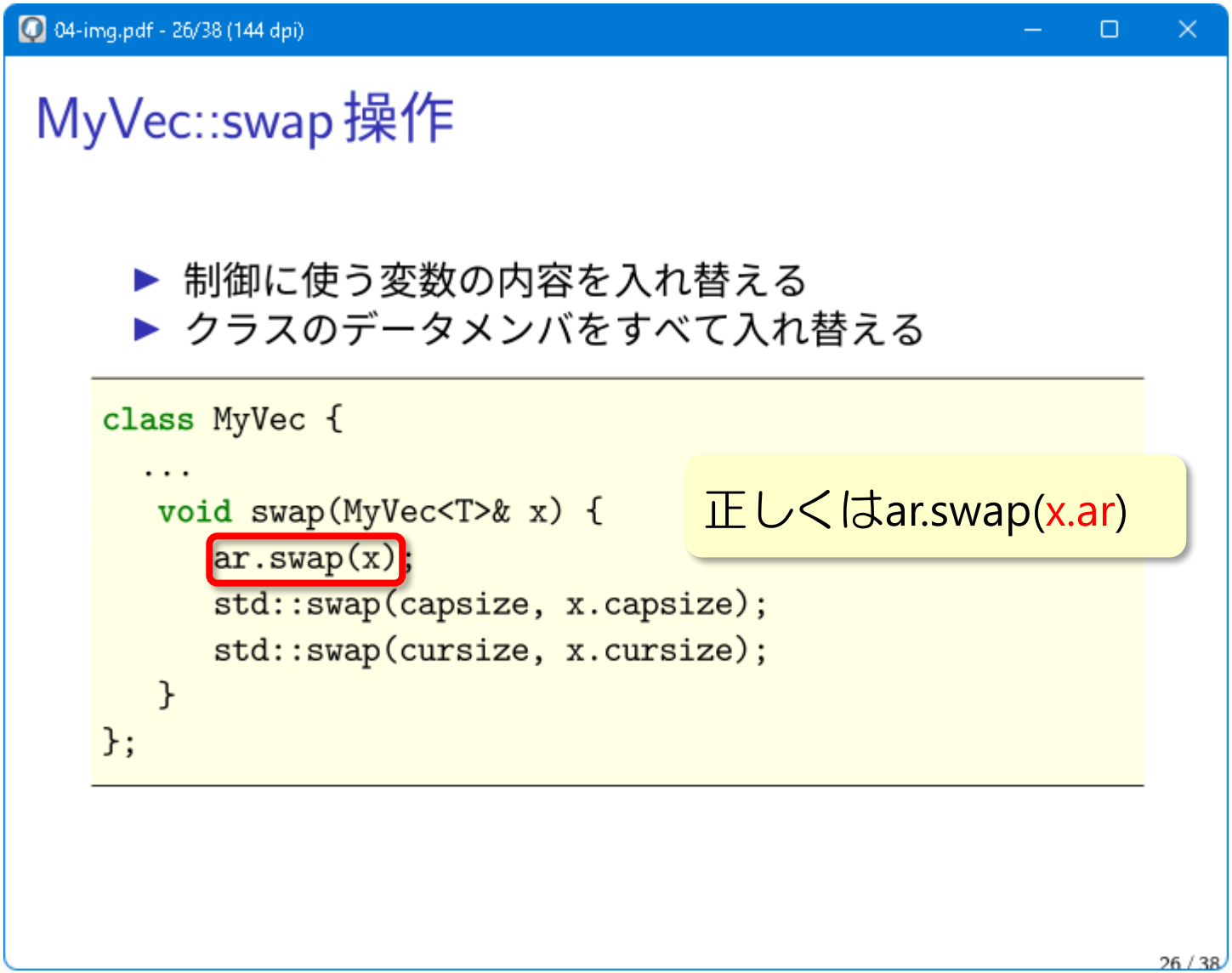
C++プログラミングII

第04回

今週のテーマ

- vector型のしくみ

資料の訂正


04-img.pdf - 26/38 (144 dpi)
MyVec::swap 操作
制御に使う変数の内容を入れ替える
クラスのデータメンバをすべて入れ替える
class MyVec {
...
void swap(MyVec<T>& x) {
ar.swap(x);
std::swap(capsize, x.capsize);
std::swap(cursize, x.cursize);
}
};
正しくはar.swap(x.ar)
26 / 38

演習

質問はzoomのチャットから
ID: 645 671 3836
PW: 3205

- 課題はCourse Powerにあります
- 回答をCourse Powerにアップロードしてください
  - 問題ごとに提出先がわかれているので注意
- 提出締め切り: 5/02 12:30
  - 締め切り後に解答例が公開されます
- 提出した人は，退出して構いません

演習1

04prac-img.pdf - 2/12 (144 dpi)

演習 1

- 以下のコードを実行して結果を確認せよ。

```
#include <iostream> // prac04-01.cpp
#include <vector>
template<typename T>
void print(const T& a) {
   std::cout << a.size() <<", "<< a.capacity() <<": ";
   for (size_t i=0; i<a.size(); i++)
      std::cout << a[i] <<" ";
   std::cout <<"\n";
}

int main() {
   std::vector<int> a {1,2,3};
   print(a);                  // 3, 3: 1 2 3
   a.pop_back(); print(a); // 2, 3: 1 2
   a.pop_back(); print(a); // 1, 3: 1
   a.clear();    print(a); // 0, 3:
}
```

2 / 12

講義スライドp.10の例を実行してみる問題.

演習2

04prac-img.pdf - 3/12 (144 dpi)

演習 2

- 以下のコードを実行して結果を確認せよ。

```
#include <iostream> // prac04-02.cpp
#include <vector>
using std::cout;
int main() {
   std::vector a {3,2,3,4,8};
   a.front() = 1;
   a.back() = 5;
   cout << a.front() <<" "<< a.back() <<"\n"; // 1 5
   for (auto e : a)
      cout << e <<" ";
   cout <<"\n"; // 1 2 3 4 5
}
```

3 / 12

講義スライドp.11の例を実行してみる問題.

演習3

04prac-img.pdf - 4/12 (144 dpi)

演習 3

- 以下のコードを実行して it がポインタのように使用できること確認せよ。

```
#include <iostream> // prac04-03.cpp
#include <vector>
using std::cout;
int main() {
  std::vector a {1,2,3,4};
  auto it { a.begin() };   // イテレータ
  cout << *it <<" "<< *(it + 1)<<"\n"; // 1 2
  ++it;
  cout << *it <<"\n";                  // 2
  cout << a.end() - a.begin() <<"\n";  // 4
}
```

4 / 12

講義スライドp.12の例を実行してみる問題.

演習4

04prac-img.pdf - 5/12 (144 dpi)

演習 4

- 以下のコードを実行して結果を確認せよ。

```
#include <iostream> // prac04-04.cpp
#include <vector>

// 演習 1 と同じ print 関数

int main() {
   std::vector a {8,6,5,3,2,1};
   a.insert(a.begin(), 9);   print(a);
   a.insert(a.begin()+2, 7); print(a);
   a.erase( a.begin()+3 );   print(a);
}
```

```
7, 12: 9 8 6 5 3 2 1
8, 12: 9 8 7 6 5 3 2 1
7, 12: 9 8 7 5 3 2 1
```

5 / 12

講義スライドp.13の例を実行してみる問題.

演習5

04prac-img.pdf - 6/12 (144 dpi)

演習 5

- 以下のコードを実行して結果を確認せよ。

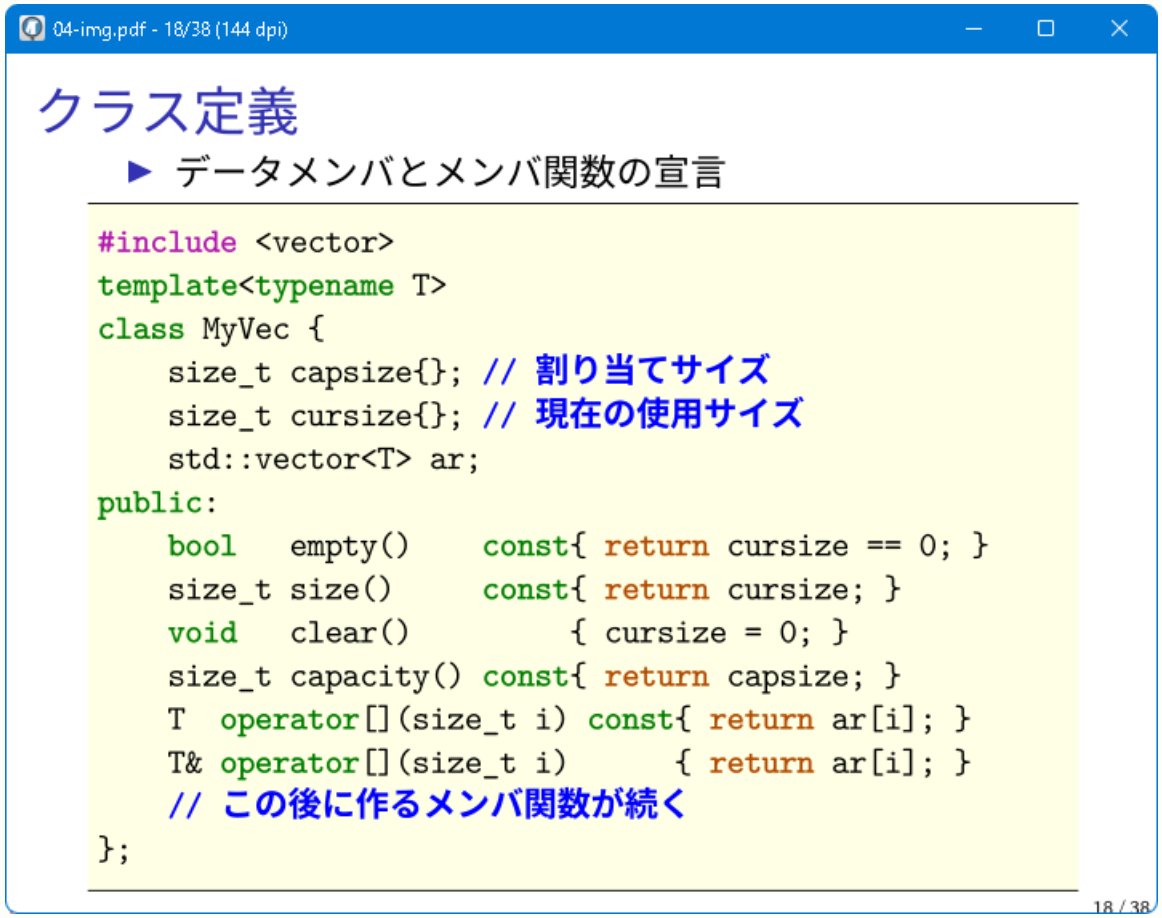
```
#include <iostream>  // prac04-05.cpp
#include <vector>
int main() {
  std::vector a {1,2,3,4,5};
  for (auto it=a.begin(); it!=a.end(); ++it) {
     auto e { *it };
     std::cout << e <<" ";
  }
  std::cout <<"\n";  // 1 2 3 4 5

  for (auto e : a) std::cout << e <<" ";
  std::cout <<"\n";  // 1 2 3 4 5
}
```

6 / 12

講義スライドp.14の例を実行してみる問題.

演習6-9

- vector型を自作してみることで，しくみを理解する演習



演習6

04prac-img.pdf - 8/12 (144 dpi)

演習 6

myvec.htpp → myvec.hppの間違い

- 資料 p.18 に push_back() と pop_back() を加えた myvec.htpp を作成し以下の動作を確認せよ。

```
#include <iostream>  // prac04-06.cpp
#include "myvec.hpp"

// 演習 1 と同じ print 関数

int main() {
   MyVec<int> a;
   for (size_t i=0; i<5; i++)
     a.push_back(i+1);
   print(a);                  // 5, 8: 1 2 3 4 5
   a.pop_back(); print(a); // 4, 8: 1 2 3 4
   a.pop_back(); print(a); // 3, 8: 1 2 3
   a.clear();    print(a); // 0, 8:

}
```

8 / 12

演習6のヒント: myvec.hppの作り方

```
#include <vector>
template<typename T>
class MyVec {
  size_t capsize{};
  size_t cursize{};
  std::vector<T> ar;

public:
  bool empty() const { return cursize == 0; }
  size_t size() const { return cursize; }
  void clear() { cursize = 0; }
  size_t capacity() const { return capsize; }
  T operator[](size_t i) const { return ar[i]; }
  T& operator[](size_t i) { return ar[i]; }

  void push_back(const T& x) {
    // 講義資料p.24の定義を書く
  }

  void pop_back() {
    // 講義資料p.25の定義を書く
  }
};
```

演習7

04prac-img.pdf - 9/12 (144 dpi)

演習 7

- 前の演習の myvec.hpp に insert() と begin() を追加して以下を試せ。

```
#include <iostream>  // prac04-07.cpp
#include "myvec.hpp"

// 演習 1 と同じ print 関数

int main() {
   MyVec<int> a;
   a.push_back(2);
   a.push_back(3);
   a.push_back(5);
   a.insert(a.begin(), 1); print(a); // 4, 4: 1 2 3 5
   a.insert(a.begin()+3,4);print(a); // 5, 8: 1 2 3 4 5
}
```

9 / 12

演習7のヒント: myvec.hppの作り方

```
#include <vector>
template<typename T>
class MyVec {
  ...
public:
  ...
  size_t begin() {
    // 講義資料p.27の定義を書く
  }

  size_t insert(size_t pos, const T& v) {
    // 講義資料p.30の定義を書く
  }
};
```

演習6で作成したコードに以下の関数の定義を追加する

演習8

04prac-img.pdf - 10/12 (144 dpi)

演習 8

- 前の演習の myvec.hpp に erase() と end() を追加して以下を試せ。

```
#include <iostream>  // proc04-08.cpp
#include "myvec.hpp"

// 演習 1 と同じ print 関数

int main() {
   MyVec<int> a;
   for (size_t i=0; i<6; i++)
      a.push_back(i);
   a.erase(a.begin()+2); print(a); // 5, 8: 0 1 3 4 5
   a.erase(a.end()-3);   print(a); // 4, 8: 0 1 4 5
}
```

10 / 12

演習8のヒント: myvec.hppの作り方

```
#include <vector>
template<typename T>
class MyVec {
  ...
public:
  ...
  size_t end() {
    // 講義資料p.27の定義を書く
  }

  size_t erase(size_t pos) {
    // 講義資料p.28の定義を書く
  }
};
```

演習7で作成したコードに以下の関数の定義を追加する

演習9

04prac-img.pdf - 11/12 (144 dpi)

演習 9

- 以下を試してみよ。

```
#include <iostream>  // proc04-09.cpp
#include "myvec.hpp"
// 演習 1 と同じ print 関数
template<typename T>
void test(T& a) {
   for (size_t i=0; i<6; i++) a.push_back(i);
   print(a);
   a.insert(a.begin()+4, 8); print(a);
   a.erase(a.begin()+2);     print(a);
   a.erase(a.end()-3);       print(a);
   a.clear();                print(a);
}
int main(){
   std::vector<int> x; test(x);
   MyVec<int> y;       test(y);
}
```

11 / 12

標準のvector型と自作のMyVecの動作を比較する問題