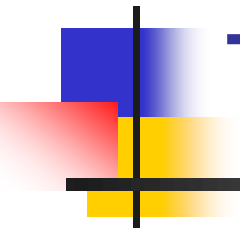


TIN ĐẠI CƯƠNG

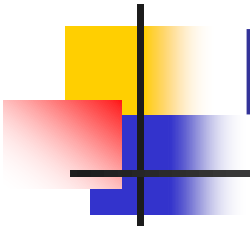
Bài 7: Khuôn mẫu & Chỉ mục

Nhắc lại nội dung bài trước

- Các kiểu dữ liệu (int, unsigned int, char, double, float, bool)
- Khai báo hằng số (const) và tham chiếu
- Phạm vi và vòng đời của biến
- Các kiểu dữ liệu tự tạo bằng cách ghép những kiểu dữ liệu khác với nhau

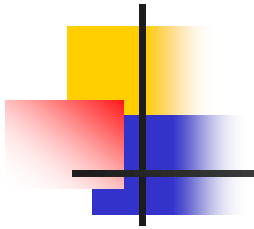
Nhắc lại nội dung bài trước

- Kiểu chuỗi (string):
  - Bản chất: Dãy các kí tự
  - #include <string>
  - Khai báo:
    - string str;
    - string w("Hello");
    - string s = "Hello";
  - Các hàm cơ bản: Tham khảo trang 473-474 của giáo trình



Bài 7: Khuôn mẫu & chỉ mục

- Khuôn mẫu (template)
- Chỉ mục (index)
- Sử dụng chỉ mục với chuỗi kí tự
- Vector
- Bài tập



Khuôn mẫu (template)

Khuôn mẫu (template)

- Nhiều thuật toán có tính tổng quát, có thể áp dụng được cho nhiều loại dữ liệu khác nhau
- Ví dụ: Tìm phần tử lớn nhất trong 2 phần tử

```
int max(int a, int b) {
    if (a > b) return a; else return b;
}
double max(double a, double b) {
    if (a > b) return a; else return b;
}
string max(string a, string b) {
    if (a > b) return a; else return b;
}
```

Khuôn mẫu (template)

- Ngôn ngữ C++ cho phép chúng ta "tổng quát hóa" các đoạn mã tương tự này bằng cách sử dụng template
- Ví dụ: Tìm phần tử lớn nhất trong 2 phần tử

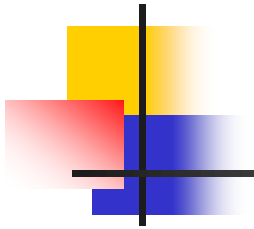
```
template <class T> T max(T a, T b) {
   if (a > b) return a; else return b;
}
```

- Sử dụng: Máy tính sẽ tự động thay thế kiểu dữ liệu thích hợp

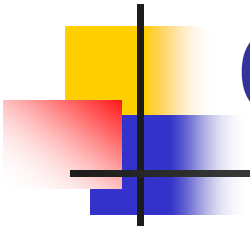
Tự động dùng hàm max với kiểu int

```
cout << max(100,200) << endl;
cout << max(1.5,1.3) << endl;
```

Tự động dùng hàm max với kiểu double

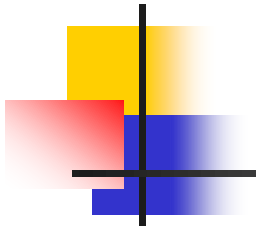


Chỉ mục (index)



Chỉ mục (index)

- Vấn đề: Về bản chất chuỗi kí tự thực chất là một dãy các chữ, liệu có cách nào thao tác đến 1 kí tự trong chuỗi hay không?
- Lời giải: Sử dụng hệ thống chỉ mục kèm với tên biến
- Chỉ mục là các số nguyên, bắt đầu từ 0



Sử dụng chỉ mục với chuỗi kí tự

Sử dụng chỉ mục với chuỗi kí tự

Dữ liệu	**H**	**A**	**P**	**P**	**Y**		**N**	**E**	**W**
Chỉ mục	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**…**

- Chuỗi s = "HAPPY NEW YEAR"
- s[0] ➔ 'H' s[1] ➔ 'A' s[2] ➔ 'P'
- Cách dùng:
  - Lấy ra: **cout << s[1] << endl;**
  - Ghi vào: **s[4] = 'I';**
  - Kết hợp: **for (int i = 0; i < 10; i++) cout << s[i] << endl;**

Vector

Vector

- Mẫu Vector cho phép tạo ra các loại danh sách các phần tử
- Ví dụ:

```
vector<int> x(10);
for (int i=0; i<x.size(); i++)
    x[i] = i*i;
for (int j=0; j<x.size(); j++)
    cout << x[j] << endl;
```

Danh sách 10 số nguyên x

Vector (cách dùng)

- Cần: #include <vector>
- Khai báo biến:
  - vector<bool> m;
  - vector<int> a(10);
  - vector<double> b(10,0.5);
- Sử dụng chỉ mục để truy cập từ phần tử bên trong biến, chỉ mục là số nguyên bắt đầu từ 0
- Các hàm do thư viện vector cung cấp để thao tác danh sách (xem bảng 6-2, trang 278 và phụ lục trang 474-475)

Một số hàm của vector

- v.clear(): Xóa rỗng vector v
- v.empty(): Trả về true nếu vector v rỗng
- v.pop_back(): Bỏ phần tử cuối cùng của vector v
- v.push_back(e): Chèn e vào cuối vector v
- v.size(): Trả về số phần tử của vector v

Bài tập

Một số bài tập cơ bản

1. Nhập số nguyên dương N và mảng N số thực, in ra các số vừa nhập
2. Nhập mảng N số thực và tính tổng
3. Nhập mảng N số thực và tính trung bình cộng của các số trong mảng
4. Nhập mảng N số nguyên và tính trung bình cộng các số dương trong mảng

Một số bài tập cơ bản

5. Nhập mảng N số thực, đếm và in ra màn hình các số trong mảng có giá trị nhỏ hơn trung bình cộng của mảng
6. Nhập mảng N số thực, sắp xếp lại các số trong mảng giảm dần theo giá trị
7. Nhập danh sách N sinh viên, sắp xếp lại danh sách theo sinh viên theo thứ tự từ điển