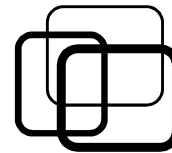


Thư viện STL, Tính đóng gói

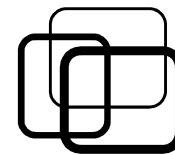
GV. Nguyễn Minh Huy



- Thư viện C++.
- Thư viện STL.
- Tính đóng gói.



- **Thư viện C++.**
- **Thư viện STL.**
- **Tính đóng gói.**



■ Khái niệm thư viện:

■ Thợ xây dựng dùng gì để xây nhà?

- Tay không.
- Công cụ hỗ trợ.

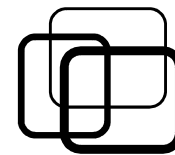
■ Thư viện lập trình:

- Bộ công cụ hỗ trợ của lập trình viên.
- Tập hợp tri thức của người đi trước.
➔ Các lớp, hàm dựng sẵn (file .lib).

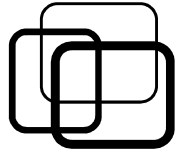
■ Các thư viện C++ phổ biến:

- Thư viện chuẩn C++.
- Thư viện boost: hơn 80 thư viện hỗ trợ lập trình C++.
- Thư viện MFC: hỗ trợ lập trình C++ trên Windows.

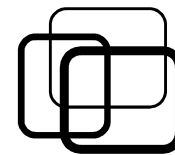




- Thư viện chuẩn (C++ Standard Library):
 - Phát triển và chuẩn hóa vào 1990s.
 - Bộ công cụ lập trình nền tảng.
 - Tập hợp các lớp trong namespace std.
 - Đặc điểm: file header không .h.
 - Cấu trúc:
 - Thư viện nhập xuất: `iostream`, `fstream`, `iomanip`, ...
 - Thư viện STL.
 - Thư viện chuẩn C: file header `cxxx.h`.

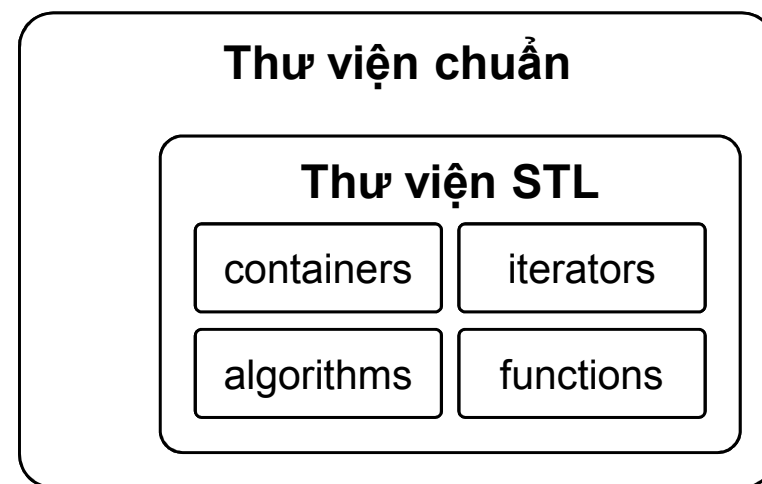


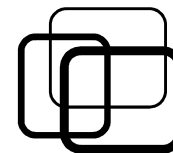
- Thư viện C++.
- **Thư viện STL.**
- Tính đóng gói.



■ Giới thiệu STL—Standard Template Library:

- Alexander Stepanov, 1994.
- Phần chính yếu của thư viện chuẩn.
- Tập hợp class template dựng sẵn.
- Hỗ trợ cấu trúc dữ liệu tổng quát.
- Cấu trúc thư viện:
 - Nhóm containers.
 - Nhóm iterators.
 - Nhóm algorithms.
 - Nhóm functors.

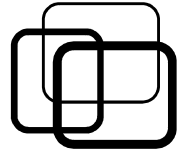




■ Lớp string (header <string>):

- Lớp đại diện cho chuỗi ký tự.
- **Giải quyết 3 vấn đề con trỏ.**
- Các phương thức chính:

Phương thức	Ý nghĩa
string(char *)	Khởi tạo từ một chuỗi char *.
length()	Lấy chiều dài chuỗi.
Toán tử []	Lấy ký tự tại vị trí nào đó.
Toán tử >, <, >=, <=, ==, !=	So sánh chuỗi theo thứ tự từ điển.
Toán tử +, +=	Nối chuỗi.
find(char *)	Tìm vị trí chuỗi con.
substr(int, int)	Trích chuỗi con.

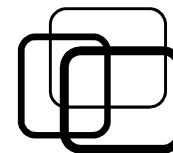


■ Lớp string:

```
void main()
{
    string s1("software");
    string s2("SoftWare");
    if (s1 == s2)
        cout << "equal." << endl;
    else
        cout << "not equal." << endl;

    s2 = s1.substr(4, 4);
    cout << s2;

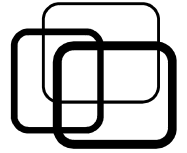
    string s3 = s1 + s2;
    cout << s3 << endl;
}
```



■ Lớp vector (header <vector>):

- Lớp mảng kiểu tổng quát T.
- **Giải quyết 3 vấn đề con trỏ.**
- Các phương thức chính:

Phương thức	Ý nghĩa
vector<T>()	Khởi tạo mảng kiểu T.
size()	Lấy kích thước mảng.
resize(int)	Thay đổi kích thước mảng.
Toán tử []	Lấy phần tử tại vị trí nào đó.
push_back(T)	Thêm phần tử vào cuối mảng.
insert(position, T)	Chèn phần tử vào vị trí nào đó.

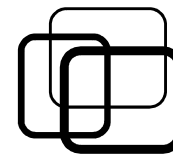


■ Lớp vector:

```
void main()
{
    vector<int>    v1;
    v1.push_back(1);
    v1.push_back(2);

    for (int i = 0; i < v1.size(); i++)
        cout << v1[i] << " ";

    vector<PhanSo *> v2;
    v2.push_back(new PhanSo(1, 3));
    v2[0]->rutGon();
}
```



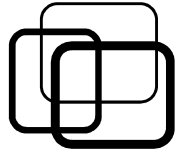
■ Luyện tập:

Viết chương trình thực hiện việc sau:

- Nhập vào danh sách N chuỗi ký tự.
- Nối các chuỗi vừa nhập và xuất kết quả.

Cài đặt theo hai cách:

- Sử dụng lớp mảng động thuần túy.
- Sử dụng lớp string và vector của thư viện STL.



- Thư viện C++.
- Thư viện STL.
- **Tính đóng gói.**

Tính đóng gói



■ Quy tắc hộp đen:

- Thuộc tính: **private** để hạn chế truy xuất.
- Phương thức: **public** để cung cấp tính năng.

```
class HocSinh
```

```
{
```

```
  private:
```

```
    char*    m_hoTen;
```

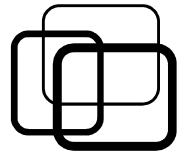
```
    float    m_diemVan;
```

```
    float    m_diemToan;
```

```
};
```

Dữ liệu
(Thuộc tính)

Tính năng
(Phương thức)



- Nhu cầu xử lý vs. quy tắc hộp đen:
 - Có nhu cầu truy xuất thuộc tính đối tượng?
 - Phương án 1: private → public.
 - Phương án 2: phương thức truy xuất.
 - ➔ Vẫn vi phạm quy tắc hộp đen!!

```
class HocSinh
{
public:
    char*    m_hoTen;
    float    m_diemVan;
    float    m_diemToan;
};
```

```
class HocSinh
{
    private:
        char*    m_hoTen;
        float    m_diemVan;
        float    m_diemToan;

    public:
        float&   layDiemVan();
        float&   layDiemToan();
};
```



- Làm sao vẫn tuân thủ quy tắc hộp đen?
 - Đưa xử lý cho đối tượng thực hiện.

```
class HocSinh
{
private:
    char*    m_hoTen;
    float    m_diemVan;
    float    m_diemToan;
public:
    float tinhDTB();
    int xepLoai();
};
```




- Dr. Guru khuyên: quy tắc **“Tell, Don’t Ask”**.
 - Thông tin đối tượng
 - ➔ Không thể truy xuất từ bên ngoài.
 - Đối tượng nắm giữ thông tin
 - ➔ Chịu trách nhiệm xử lý thông tin.
 - “Đừng hỏi tôi thông tin để thực hiện công việc”
 - ➔ “Hãy nhờ tôi làm giùm!!”
 - Đã giao thông tin
 - ➔ Giao luôn xử lý thông tin.





■ Luyện tập:

// Tính trọng tâm tam giác??

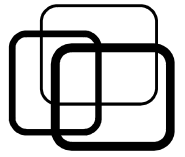
```
class Diem
{
private:
    float    m_X;
    float    m_Y;
};

class TamGiac
{
private:
    Diem    m_A;
    Diem    m_B;
    Diem    m_C;
};
```

*// Xuất danh sách học sinh đậu??
// (ĐTB >= 5.0)*

```
class HocSinh
{
private:
    char    *m_hoTen;
    float    m_diemVan;
    float    m_diemToan;
};

class BangDiem
{
private:
    vector<HocSinh>    m_danhSach;
};
```



■ Thư viện C++:

- Bộ công cụ dựng sẵn hỗ trợ lập trình C++.
- Các thư viện phổ biến:
 - Thư viện chuẩn:
 - Thư viện boost.
 - Thư viện MFC.

■ Thư viện STL:

- Phần chính yếu của thư viện chuẩn.
- Các lớp dựng sẵn: string, vector.

■ Tính đóng gói:

- Tell, Don't Ask.

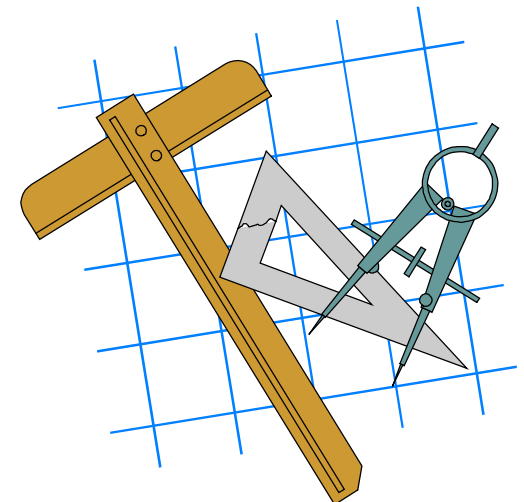




■ Bài tập 5.1:

Với mỗi thư viện bên dưới, hãy liệt kê tên và cho biết tính năng của ít nhất 3 lớp đối tượng trong thư viện:

- Thư viện chuẩn.
- Thư viện boost.
- Thư viện MFC.

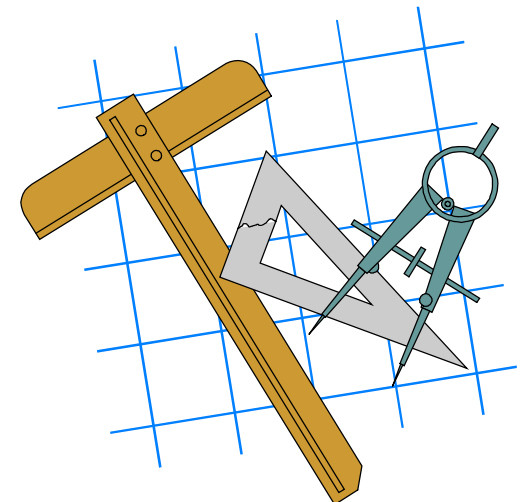




■ Bài tập 5.2:

Một đoạn văn bao gồm nhiều câu. Mỗi câu bao gồm nhiều từ và kết thúc bằng dấu câu (dấu ., ?, !). Xây dựng các lớp cần thiết để:

- Đọc một đoạn văn từ file văn bản.
- Đếm số lượng câu và số lượng từ có trong đoạn văn.
- Tìm từ xuất hiện nhiều nhất trong đoạn văn.
- Sắp xếp tăng dần (thứ tự từ điển) các từ trong mỗi câu.
- Ghi đoạn văn đã chỉnh sửa vào file văn bản.





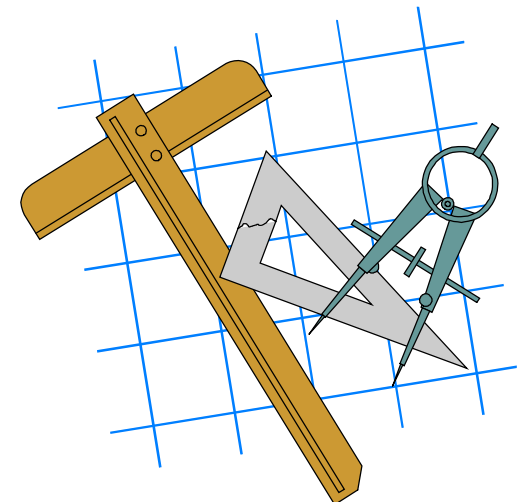
■ Bài tập 5.3:

Xây dựng lớp **đa thức** có những phương thức sau:
(Nhóm tạo hủy)

- Khởi tạo mặc định đa thức = 0.
- Khởi tạo với bậc và mảng hệ số cho trước.
- Khởi tạo từ một đối tượng đa thức khác.
- Hủy đa thức, thu hồi bộ nhớ.

(Nhóm truy xuất thông tin)

- Lấy bậc đa thức.
- Lấy hệ số tại bậc nào đó.





■ Bài tập 5.3:

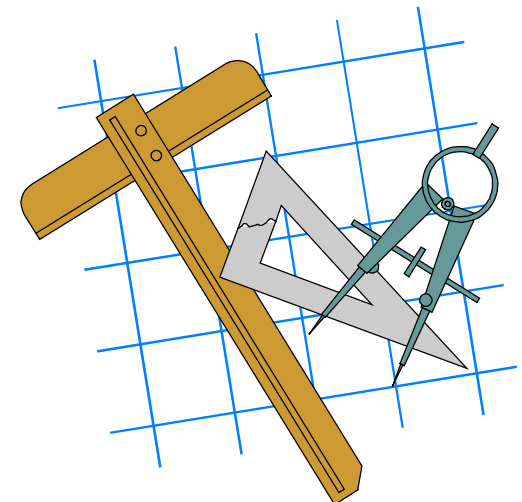
Xây dựng lớp **đa thức** (tiếp theo):

(Nhóm xử lý nghiệp vụ)

- So sánh với đa thức khác.
- Rút gọn đa thức.
- Tính giá trị, đạo hàm, nguyên hàm.

(Nhóm toán tử)

- Toán tử số học: $+$, $-$, $*$, $/$, $=$.
- Toán tử so sánh: $>$, $<$, $==$, $>=$, $<=$, $!=$.
- Toán tử đạo hàm (!), nguyên hàm (~).
- Toán tử nhập xuất: $>>$, $<<$.





■ Bài tập 5.4:

Xây dựng các lớp cần thiết để thực hiện các thao tác trên tam giác:

- Tính chu vi và diện tích.
- Tìm trọng tâm (giao 3 đường trung tuyến).
- Tìm đường trung trực của một cạnh (vuông góc tại trung điểm).
- Tìm tâm đường tròn ngoại tiếp (giao 3 đường trung trực).

