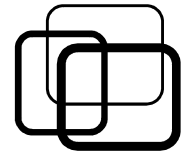


# Vấn đề con trỏ, Tính đóng gói

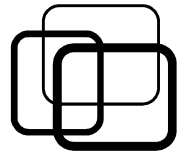
GV. Nguyễn Minh Huy



- Vấn đề con trỏ.
- Tính đóng gói.



- **Vấn đề con trỏ.**
- **Tính đóng gói.**



## ■ Các phương thức mặc định trong lớp:

- Được trình biên dịch cung cấp khi không khai báo.
- Ba phương thức mặc định:
  - Phương thức hủy mặc định.
  - Phương thức khởi tạo sao chép mặc định.
  - Toán tử gán mặc định.

```
class PhanSo
{
private:
    int m_tu;
    int m_mau;
public:
    PhanSo( int tu, int mau );
};
```

```
void main()
{
    PhanSo p1( 1, 3 );
    // Mặc định có phương thức
    // khởi tạo sao chép.
    PhanSo p2( p1 );

    // Mặc định có toán tử gán.
    p1 = p2;
}
```



## ■ Ví dụ 1:

```
class Array
{
private:
    int    m_size;
    int    *m_data;
public:
    Array(int size);
};
Array::Array(int size)
{
    m_size = size;
    m_data = new int[m_size];
}
```

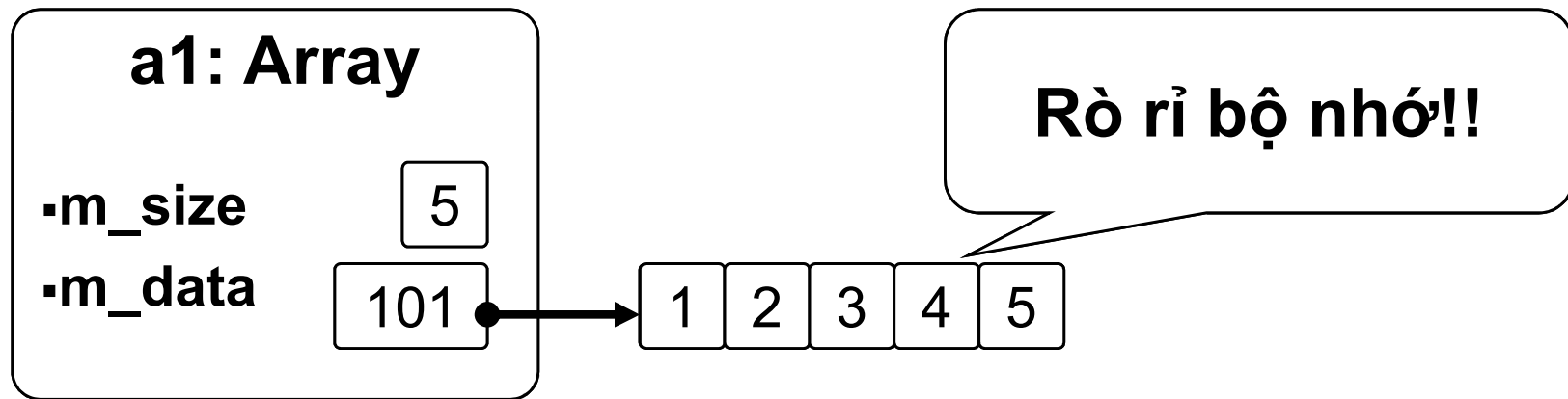
```
void main()
{
    Array a1(5);
    ...
}
```

# Vấn đề con trỏ



## ■ Vấn đề hủy mặc định:

- Lớp có thuộc tính con trỏ (có cấp phát bộ nhớ).
- Phương thức hủy mặc định không thu hồi bộ nhớ!!



**Phải cài đặt TƯỜNG MINH phương thức hủy để dọn dẹp!!**



## ■ Ví dụ 1:

```
class Array
{
private:
    int    m_size;
    int    *m_data;
public:
    Array(int size);
    ~Array();
};
Array::~~Array()
{
    delete []m_data;
}
```

```
void main()
{
    Array  a1(5);
    ...
}
```



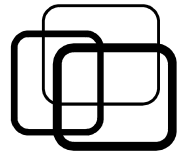
## ■ Ví dụ 2:

```
class Array
{
private:
    int      m_size;
    int      *m_data;
public:
    Array(int size);
    ~Array();
};
```

```
void main()
{
    Array  a1(5);
    Array  a2(a1);
    ...
}
```

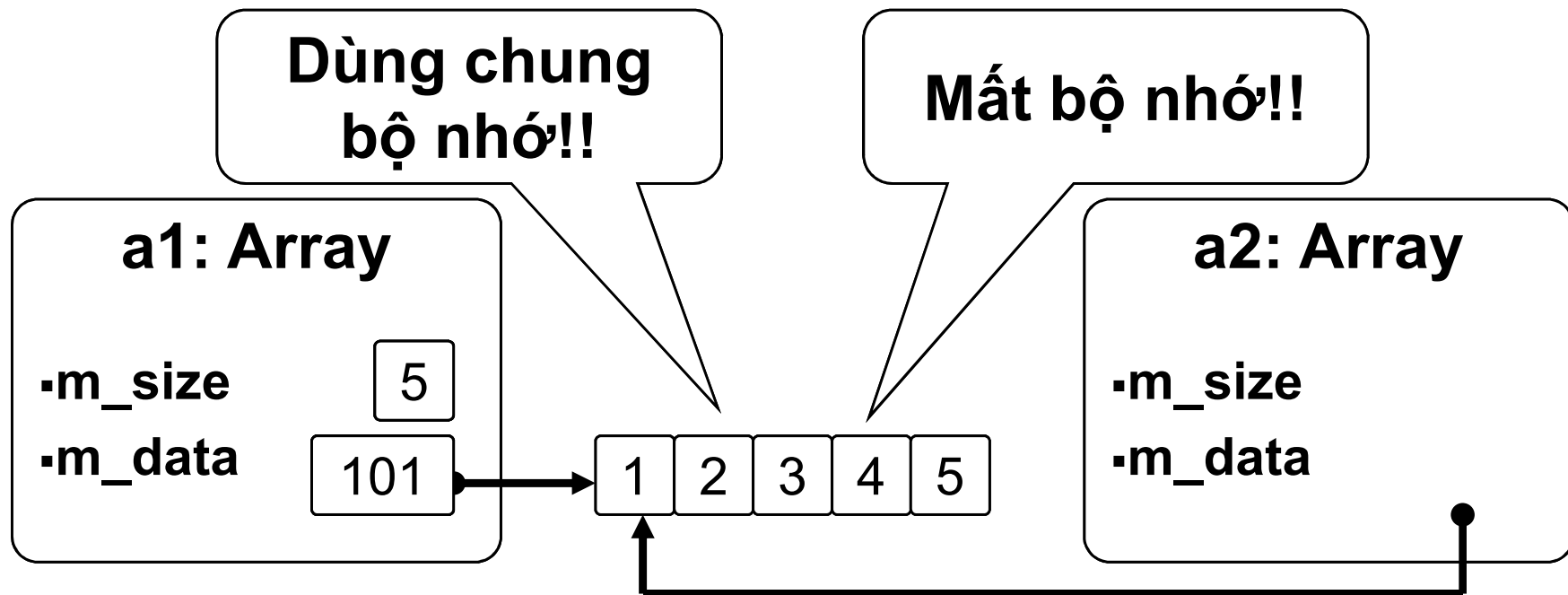


# Vấn đề con trỏ

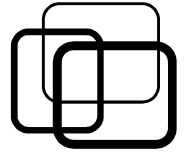


## ■ Vấn đề sao chép mặc định:

- Phương thức khởi tạo sao chép mặc định sao chép giá trị từng thuộc tính!!



**Phải cài đặt TƯỜNG MINH phương thức khởi tạo sao chép!!**



## ■ Ví dụ 2:

```
class Array
{
private:
    int      m_size;
    int      *m_data;
public:
    Array(int size);
    Array(const Array &a);
    ~Array();
};
```

```
Array::Array(const Array &a)
{
    m_size = a.m_size;
    m_data = new int[m_size];
    for (int i = 0; i < m_size; i++)
        m_data[ i ] = a.m_data[ i ];
}

void main()
{
    Array  a1(5);
    Array  a2(a1);
    ...
}
```

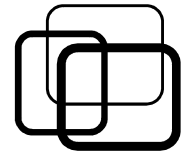


## ■ Ví dụ 3:

```
class Array
{
private:
    int    m_size;
    int    *m_data;
public:
    Array(int size);
    Array(const Array &a);
    ~Array();
};
```

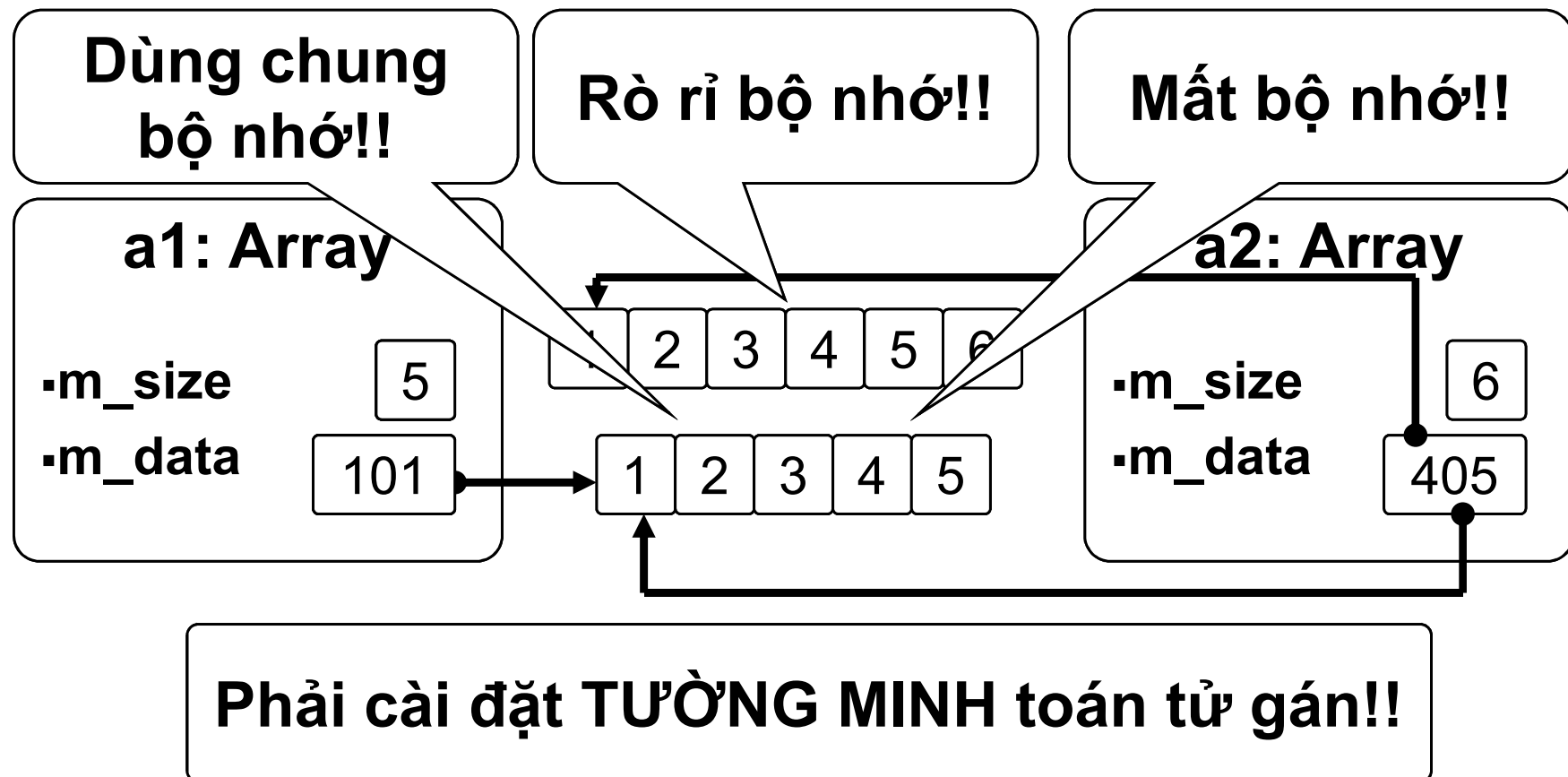
```
void main()
{
    Array a1(5);
    Array a2(6);
    ...
    a2 = a1;
    ...
}
```

# Vấn đề con trỏ



## ■ Vấn đề gán mặc định:

- Toán tử gán mặc định sao chép giá trị từng thuộc tính!!



# Vấn đề con trỏ



## ■ Ví dụ 2:

```
class Array
{
private:
    int      m_size;
    int      *m_data;
public:
    Array(int size);
    Array(const Array &a);
    ~Array();
    Array & operator =(const Array &a);
};
```

```
Array & Array::operator =(const Array &a)
{
    delete []m_data;
    m_size = a.m_size;
    m_data = new int[m_size];
    for (int i = 0; i < m_size; i++)
        m_data[ i ] = a.m_data[ i ];
    return *this;
}

void main()
{
    Array  a1(5);
    Array  a2(6);
    ...
    a2 = a1;
    ...
}
```



## ■ Dr. Guru khuyên: luật “ba ông lớn”

- Lớp có thuộc tính con trỏ (có cấp phát bộ nhớ),  
→ Phải cài đặt TƯỞNG MINH:
  - Phương thức hủy: thu hồi bộ nhớ.
  - Phương thức khởi tạo sao chép: sao chép bộ nhớ.
  - Toán tử gán: hủy bộ nhớ cũ, sao chép bộ nhớ mới.

```
class HocSinh
{
private:
    char    *m_hoTen;
public:
    HocSinh(const HocSinh &h);
    ~HocSinh();
    HocSinh & operator =(const HocSinh &h);
};
```





- Vấn đề con trỏ.
- **Tính đóng gói.**



- Quy tắc hộp đen (rule of blackbox):
  - 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;
```

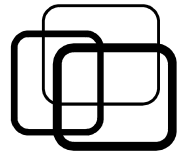
```
    float    m_diemToan;
```

```
};
```

Dữ liệu  
(Thuộc tính)

Tính năng  
(Phương thức)





## ■ Quy tắc hộp đen vs. Nhu cầu truy xuất:

### ■ Có nhu cầu truy xuất thuộc tính đối tượng?

- Phương án 1: private → public.
  - Phương án 2: phương thức truy xuất.
- Cả 2 đều 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 tuân thủ quy tắc hộp đen?
  - Cần xử lý gì thì đưa cho đối tượng thực hiện.

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

```
void main()
{
    HocSinh hs;

    // Cần tính điểm trung bình??
    // Trang bị cho học sinh tính.
    float dtb = hs.tinhDTB();

    // Cần xếp loại??
    // Trang bị cho cho học sinh xếp.
    int loai = hs.xepLoai();
}
```

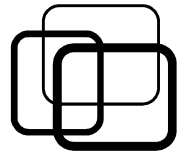
# Tính đóng gói



- 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”
    - ➔ “Hãy nhờ tôi làm giùm!!”
  - Đã giao thông tin
    - ➔ Giao luôn xử lý thông tin.



# Tính đóng gói



## ■ 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 giỏi??  
// (ĐTB >= 8.0)*

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

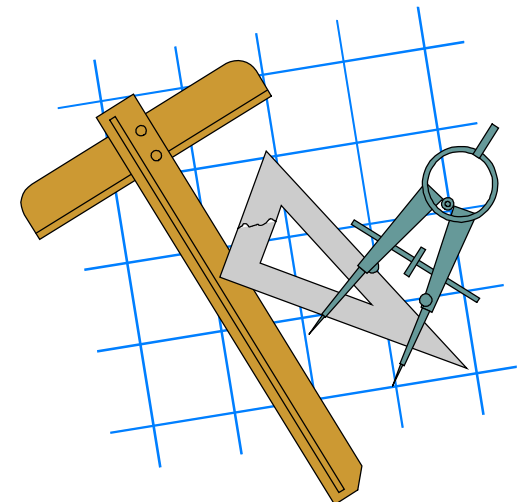
class BangDiem
{
private:
    vector<HocSinh>    m_danhSach;
};
```



## ■ Bài tập 5.1:

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.





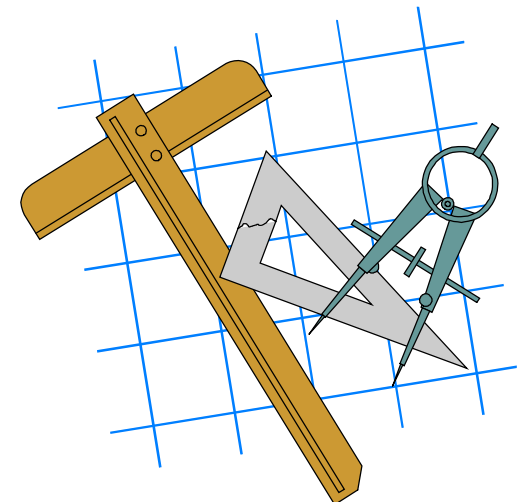
## ■ Bài tập 5.2:

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 5.2:

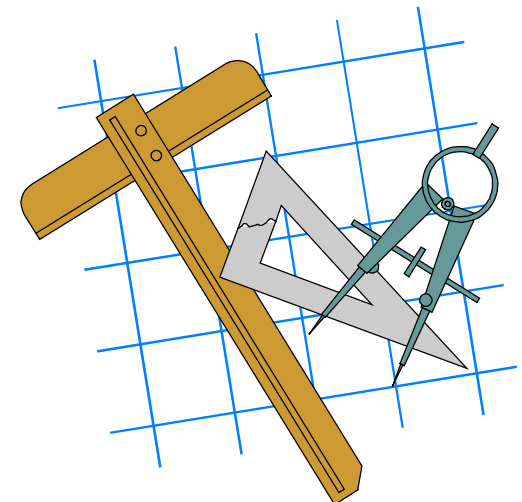
Xây dựng lớp **đa thức** (tiếp theo):

*(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 5.3:

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

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

