

THAO TÁC VỚI ĐỐI TƯỢNG

C++

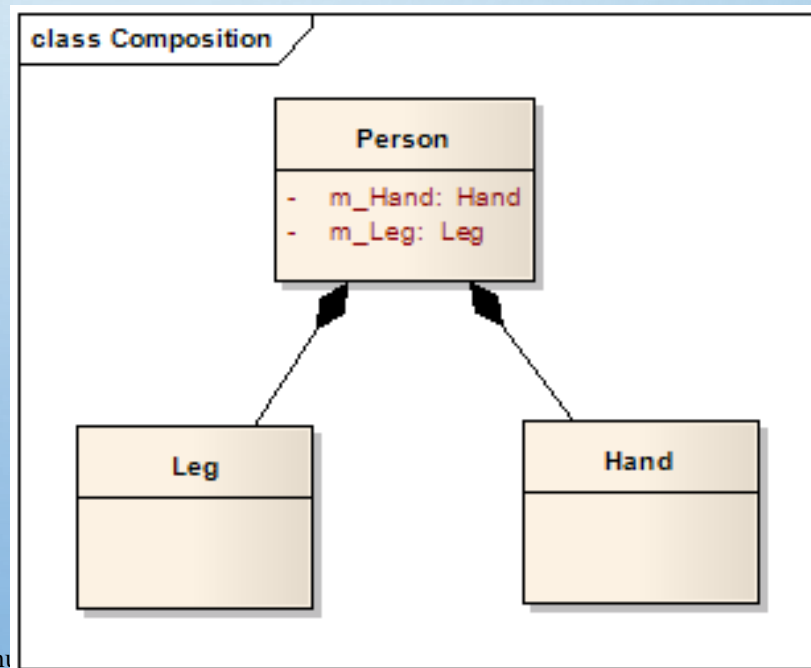


Microsoft®

Visual Studio®

ĐỐI TƯỢNG LÀ THÀNH PHẦN CỦA LỚP

❖ ĐỐI TƯỢNG CÓ THỂ LÀ THÀNH PHẦN CỦA ĐỐI TƯỢNG KHÁC, KHI MỘT ĐỐI TƯỢNG THUỘC LỚP “LỚN” ĐƯỢC TẠO RA, CÁC THÀNH PHẦN CỦA NÓ CŨNG ĐƯỢC TẠO RA.



ĐỐI TƯỢNG LÀ THÀNH PHẦN CỦA LỚP

- ❖ PHƯƠNG THỨC THIẾT LẬP (NẾU CÓ) SẼ ĐƯỢC TỰ ĐỘNG GỌI CHO CÁC ĐỐI TƯỢNG THÀNH PHẦN.
- ❖ KHI ĐỐI TƯỢNG KẾT HỢP BỊ HỦY → ĐỐI TƯỢNG THÀNH PHẦN CỦA NÓ CŨNG BỊ HỦY, NGHĨA LÀ PHƯƠNG THỨC HỦY BỎ SẼ ĐƯỢC GỌI CHO CÁC ĐỐI TƯỢNG THÀNH PHẦN, SAU KHI PHƯƠNG THỨC HỦY BỎ CỦA ĐỐI TƯỢNG KẾT HỢP ĐƯỢC GỌI.

ĐỐI TƯỢNG LÀ THÀNH PHẦN CỦA LỚP

- ❖ NẾU ĐỐI TƯỢNG THÀNH PHẦN PHẢI CUNG CẤP THAM SỐ KHI THIẾT LẬP THÌ ĐỐI TƯỢNG KẾT HỢP (ĐỐI TƯỢNG LỚN) **PHẢI CÓ PHƯƠNG THỨC THIẾT LẬP** ĐỂ CUNG CẤP THAM SỐ THIẾT LẬP CHO CÁC ĐỐI TƯỢNG THÀNH PHẦN.
- ❖ **CÚ PHÁP ĐỂ KHỞI ĐỘNG ĐỐI TƯỢNG THÀNH PHẦN LÀ DÙNG DẤU HAI CHẤM (:) THEO SAU BỞI TÊN THÀNH PHẦN VÀ THAM SỐ KHỞI ĐỘNG.**

VÍ DỤ

```
class TamGiac{  
    Diem A, B, C;  
public:  
    TamGiac(double xA, double yA, double xB, double yB, double  
    xC, double yC) : A(xA,yA) {  
    }  
    void Ve();  
    // ...  
};  
  
TamGiac t(100,100,200,400,300,300);
```

ĐỐI TƯỢNG LÀ THÀNH PHẦN CỦA MẢNG

- ❖ KHI MỘT MẢNG ĐƯỢC TẠO RA → CÁC PHẦN TỬ CỦA NÓ CŨNG ĐƯỢC TẠO RA → PHƯƠNG THỨC THIẾT LẬP SẼ ĐƯỢC GỌI CHO TỪNG PHẦN TỬ.
- ❖ VÌ KHÔNG THỂ CUNG CẤP THAM SỐ KHỞI ĐỘNG CHO TẤT CẢ CÁC PHẦN TỬ CỦA MẢNG → KHI KHAI BÁO MẢNG, MỖI ĐỐI TƯỢNG TRONG MẢNG PHẢI CÓ **KHẢ NĂNG TỰ KHỞI ĐỘNG**, NGHĨA LÀ CÓ THỂ THIẾT LẬP KHÔNG CẦN THAM SỐ.

ĐỐI TƯỢNG LÀ THÀNH PHẦN CỦA MẢNG

❖ ĐỐI TƯỢNG CÓ KHẢ NĂNG TỰ KHỞI ĐỘNG TRONG NHỮNG TRƯỜNG HỢP NÀO?

1. LỚP KHÔNG CÓ PHƯƠNG THỨC THIẾT LẬP
2. LỚP CÓ PHƯƠNG THỨC THIẾT LẬP KHÔNG THAM SỐ
3. LỚP CÓ PHƯƠNG THỨC THIẾT LẬP MÀ MỌI THAM SỐ ĐỀU CÓ GIÁ TRỊ MẶC NHIÊN

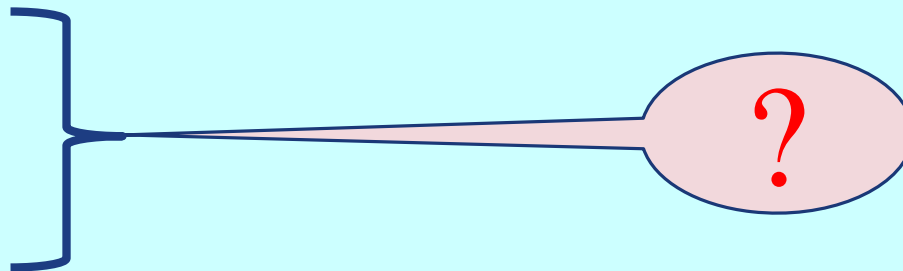
ĐỐI TƯỢNG LÀ THÀNH PHẦN CỦA MẢNG

```
class Diem
{
    double x,y;
    public:
        Diem(double xx, double yy) : x(xx), y(yy) {}
        void Set(double xx, double yy) {
            x = xx, y = yy;
        }
        // ...
};
```


ĐỐI TƯỢNG LÀ THÀNH PHẦN CỦA MẢNG

```
class SinhVien{  
    String MaSo;  
    String HoTen;  
    int NamSinh;  
  
    public:  
        SinhVien(char *ht, char *ms, int ns) : HoTen(ht), MaSo(ms),  
        NamSinh(ns){ }  
};
```

```
String arrs[3];  
Diem arrd[5];  
SinhVien arrsv[7];
```



DÙNG PHƯƠNG THỨC THIẾT LẬP VỚI THAM SỐ CÓ GIÁ TRỊ MẶC NHIÊN

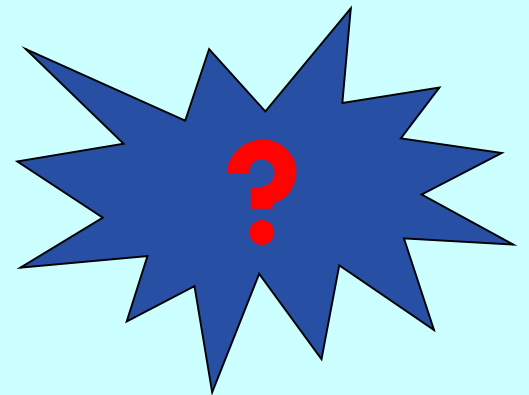
```
class Diem
{
    double x,y;
public:
    Diem(double xx = 0, double yy = 0) : x(xx), y(yy) { }
    void Set(double xx, double yy) {
        x = xx, y = yy;
    }
    // ...
};
```

DÙNG PHƯƠNG THỨC THIẾT LẬP KHÔNG THAM SỐ

```
class Diem
{
    double x,y;
public:
    Diem(double xx, double yy) : x(xx), y(yy)
    {}
    Diem() : x(0), y(0)
    {}
    // ...
};
```

DÙNG PHƯƠNG THỨC THIẾT LẬP KHÔNG THAM SỐ

```
class SinhVien {  
    String MaSo, HoTen;  
    int NamSinh;  
public:  
    SinhVien(char *ht, char *ms, int ns) : HoTen(ht), MaSo(ms),  
        NamSinh(ns) { }  
    SinhVien() : HoTen("Nguyen Van A"), MaSo("19920014"),  
        NamSinh(1982) { }  
};  
String as[3];  
Diem ad[5];  
SinhVien asv[7];
```



ĐỐI TƯỢNG ĐƯỢC CẤP PHÁT ĐỘNG

- ❖ ĐỐI TƯỢNG ĐƯỢC CẤP PHÁT ĐỘNG LÀ CÁC ĐỐI TƯỢNG ĐƯỢC TẠO RA BẰNG PHÉP TOÁN **NEW** VÀ BỊ HỦY ĐI BẰNG PHÉP TOÁN **DELETE**
- ❖ PHÉP TOÁN **NEW** CẤP ĐỐI TƯỢNG TRONG VÙNG HEAP VÀ GỌI PHƯƠNG THỨC THIẾT LẬP CHO ĐỐI TƯỢNG ĐƯỢC CẤP.

CẤP PHÁT VÀ HỦY NHIỀU ĐỐI TƯỢNG

```
INT *PAI = NEW INT[10];  
DIEM *PAD = NEW DIEM[5];  
STRING *PAS = NEW STRING[5];
```



- ❖ Trong trường hợp cấp phát nhiều đối tượng, ta không thể cung cấp tham số cho từng phần tử được cấp phát.

CẤP VÀ HỦY NHIỀU ĐỐI TƯỢNG

❖ THÔNG BÁO LỖI CHO ĐOẠN CHƯƠNG TRÌNH TRÊN NHƯ SAU:

- *CANNOT FIND DEFAULT CONSTRUCTOR TO INITIALIZE ARRAY ELEMENT OF TYPE 'DIEM'*
- *CANNOT FIND DEFAULT CONSTRUCTOR TO INITIALIZE ARRAY ELEMENT OF TYPE STRING'*

❖ KHẮC PHỤC LỖI?

Lỗi trên được khắc phục bằng cách **cung cấp phương thức thiết lập để đối tượng có khả năng tự khởi động.**

CẤP VÀ HỦY NHIỀU ĐỐI TƯỢNG

```
class String{
    char *p;
public:
    String (char *s = "Alibaba") { p = strdup(s); }
    String (const String &s) { p = strdup(s.p); }
    ~String () {delete [] p;}
    //...
};

class Diem {
    double x,y;
public:
    Diem (double xx, double yy) : x(xx),y(yy){};
    Diem () : x(0),y(0){};
};
```

CẤP VÀ HỦY NHIỀU ĐỐI TƯỢNG

❖ KHI ĐÓ MỌI PHẦN TỬ ĐƯỢC CẤP ĐỀU ĐƯỢC KHỞI ĐỘNG VỚI CÙNG GIÁ TRỊ.

```
INT *PAI = NEW INT[10];
```

```
DIEM *PAD = NEW DIEM[5];
```

```
//CA 5 DIEM CO CUNG TOA DO (0,0)
```

```
STRING *PAS = NEW STRING[5];
```

```
//CA 5 CHUOI CUNG DUOC KHOI DONG LA "ALIBABA"
```

CẤP VÀ HỦY NHIỀU ĐỐI TƯỢNG

❖ VIỆC HỦY NHIỀU ĐỐI TƯỢNG ĐƯỢC THỰC HIỆN BẰNG CÁCH DÙNG **DELETE** VÀ CÓ THÊM DẤU **[]** Ở TRƯỚC.

DELETE [] PAS;

DELETE [] PAD;

DELETE [] PAI;

❖ CÓ THỂ THAY BA PHÁT BIỂU TRÊN BẰNG MỘT PHÁT BIỂU DUY NHẤT SAU HAY KHÔNG?

DELETE PAS,PAD,PAI;