

LỚP VÀ ĐỐI TƯỢNG (TT)



PHƯƠNG THỨC THIẾT LẬP

- ❖ PHƯƠNG THỨC THIẾT LẬP HAY CÒN GỌI LÀ CONSTRUCTOR, LÀ MỘT LOẠI PHƯƠNG THỨC ĐẶC BIỆT DÙNG ĐỂ KHỞI TẠO THỂ HIỆN CỦA LỚP.
- ❖ BẤT KỲ MỘT ĐỐI TƯỢNG NÀO ĐƯỢC KHAI BÁO ĐỀU PHẢI SỬ DỤNG MỘT HÀM THIẾT LẬP ĐỂ KHỞI TẠO CÁC GIÁ TRỊ THÀNH PHẦN CỦA ĐỐI TƯỢNG.
- ❖ HÀM THIẾT LẬP ĐƯỢC KHAI BÁO GIỐNG NHƯ MỘT PHƯƠNG THỨC VỚI TÊN PHƯƠNG THỨC TRÙNG VỚI TÊN LỚP VÀ KHÔNG CÓ GIÁ TRỊ TRẢ VỀ (KỂ CẢ VOID).
- ❖ CONSTRUCTOR PHẢI CÓ THUỘC TÍNH PUBLIC

PHƯƠNG THỨC THIẾT LẬP MẶC ĐỊNH

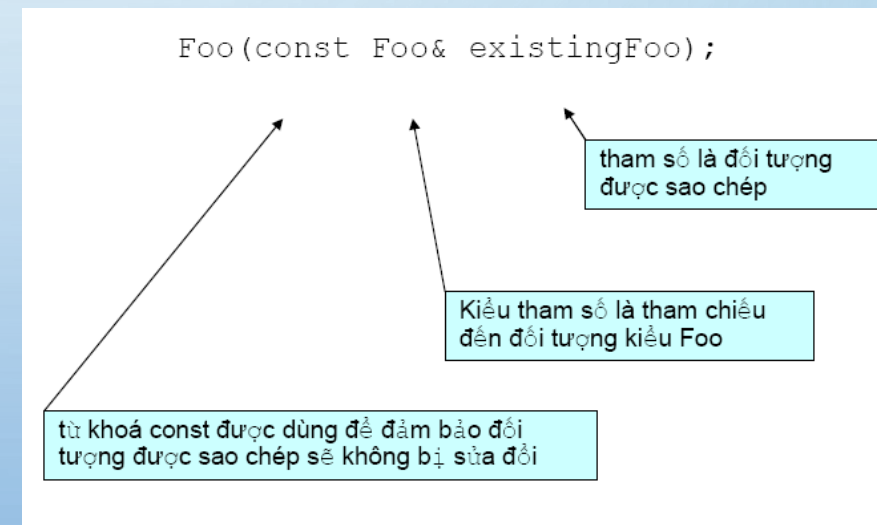
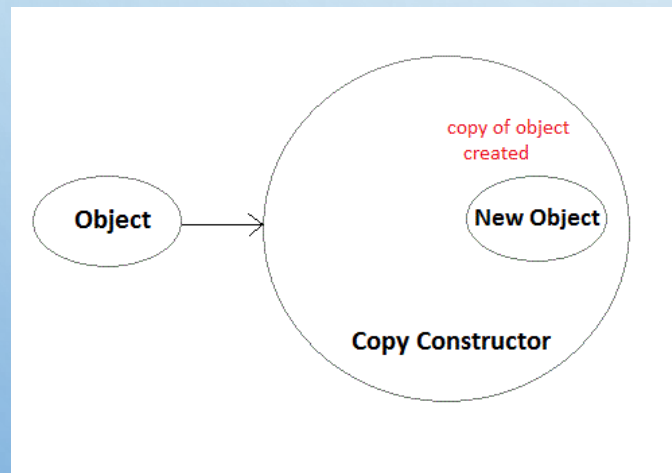
- ❖ ĐỐI VỚI CONSTRUCTOR MẶC ĐỊNH, NẾU TA KHÔNG CUNG CẤP BẤT KỲ CONSTRUCTOR NÀO, C++ SẼ TỰ SINH CONSTRUCTOR MẶC ĐỊNH LÀ MỘT PHƯƠNG THỨC RỒI.
- ❖ TUY NHIÊN, NẾU TA KHÔNG ĐỊNH NGHĨA CONSTRUCTOR MẶC ĐỊNH NHƯNG LẠI CÓ CÁC CONSTRUCTOR KHÁC, TRÌNH BIÊN DỊCH SẼ BÁO LỖI KHÔNG TÌM THẤY CONSTRUCTOR MẶC ĐỊNH NẾU TA KHÔNG CUNG CẤP THAM SỐ KHI TẠO THỂ HIỆN.

PHƯƠNG THỨC THIẾT LẬP SAO CHÉP

- ❖ CHÚNG TA CÓ THỂ TẠO ĐỐI TƯỢNG MỚI GIỐNG ĐỐI TƯỢNG CŨ MỘT SỐ ĐẶC ĐIỂM, KHÔNG PHẢI HOÀN TOÀN NHƯ PHÉP GÁN BÌNH THƯỜNG, HÌNH THỨC “GIỐNG NHAU” ĐƯỢC ĐỊNH NGHĨA THEO QUAN NIỆM CỦA NGƯỜI LẬP TRÌNH. ĐỂ LÀM ĐƯỢC VẤN ĐỀ NÀY, TRONG CÁC NGÔN NGỮ OOP CHO PHÉP TA XÂY DỰNG PHƯƠNG THỨC THIẾT LẬP SAO CHÉP.
- ❖ ĐÂY LÀ PHƯƠNG THỨC THIẾT LẬP CÓ THAM SỐ LÀ THAM CHIẾU ĐẾN ĐỐI TƯỢNG THUỘC CHÍNH LỚP NÀY.

PHƯƠNG THỨC THIẾT LẬP SAO CHÉP

❖ TRONG PHƯƠNG THỨC THIẾT LẬP SAO CHÉP CÓ THỂ TA CHỈ SỬ DỤNG MỘT SỐ THÀNH PHẦN NÀO ĐÓ CỦA ĐỐI TƯỢNG TA THAM CHIẾU → “GẦN GIỐNG NHAU”



PHƯƠNG THỨC HỦY BỎ

- ❖ PHƯƠNG THỨC HỦY BỎ HAY CÒN GỌI LÀ DESTRUCTOR, ĐƯỢC GỌI NGAY TRƯỚC KHI MỘT ĐỐI TƯỢNG BỊ THU HỒI.
- ❖ DESTRUCTOR THƯỜNG ĐƯỢC DÙNG ĐỂ THỰC HIỆN VIỆC DỌN DẸP CẦN THIẾT TRƯỚC KHI MỘT ĐỐI TƯỢNG BỊ HỦY.

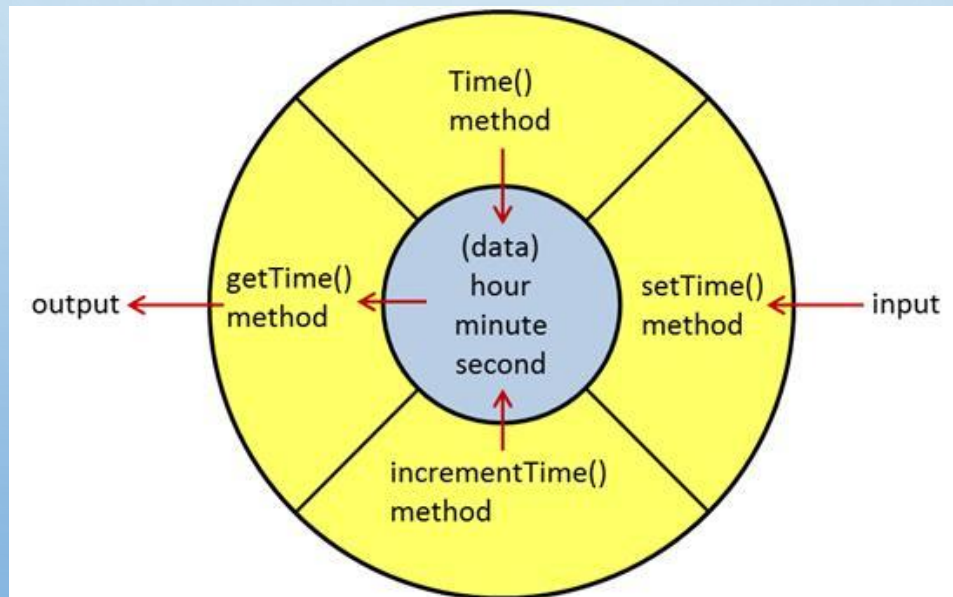


PHƯƠNG THỨC HỦY BỎ

- ❖ MỘT LỚP CHỈ CÓ DUY NHẤT MỘT **DESTRUCTOR**
- ❖ CÚ PHÁP:
 - ❖ PHƯƠNG THỨC **DESTRUCTOR** CÓ TÊN TRÙNG TÊN VỚI TÊN LỚP VÀ CÓ DẤU **~** ĐẶT TRƯỚC
 - ❖ KHÔNG CÓ GIÁ TRỊ TRẢ VỀ
- ❖ ĐƯỢC **TỰ ĐỘNG GỌI** THỰC HIỆN KHI ĐỐI TƯỢNG HẾT PHẠM VI SỬ DỤNG.
- ❖ DESTRUCTOR PHẢI CÓ THUỘC TÍNH **PUBLIC**

THAO TÁC VỚI DỮ LIỆU PRIVATE

- ❖ KHI MUỐN **TRUY XUẤT DỮ LIỆU PRIVATE** TỪ CÁC ĐỐI TƯỢNG THÌ PHẢI LÀM THẾ NÀO?
- ❖ KHI MUỐN **CẬP NHẬT DỮ LIỆU PRIVATE** TỪ CÁC ĐỐI TƯỢNG THÌ PHẢI LÀM THẾ NÀO?



PHƯƠNG THỨC TRUY VẤN

- ❖ ĐỐI VỚI CÁC TRUY VẤN ĐƠN GIẢN, QUY ƯỚC ĐẶT TÊN PHƯƠNG THỨC NHƯ SAU: TIỀN TỐ “GET”, TIẾP THEO LÀ TÊN CỦA THÀNH VIÊN CẦN TRUY VẤN
 - `INT GETX();`
 - `INT GETSIZE();`
- ❖ CÁC LOẠI TRUY VẤN KHÁC NÊN CÓ TÊN CÓ TÍNH MÔ TẢ
- ❖ TRUY VẤN ĐIỀU KIỆN NÊN CÓ TIỀN TỐ “IS”

PHƯƠNG THỨC CẬP NHẬT

- ❖ THƯỜNG ĐỂ THAY ĐỔI TRẠNG THÁI CỦA ĐỐI TƯỢNG BẰNG CÁCH SỬA ĐỔI MỘT HOẶC NHIỀU THÀNH VIÊN DỮ LIỆU CỦA ĐỐI TƯỢNG ĐÓ.
- ❖ DẠNG ĐƠN GIẢN NHẤT LÀ GÁN MỘT GIÁ TRỊ NÀO ĐÓ CHO MỘT THÀNH VIÊN DỮ LIỆU.
- ❖ ĐỐI VỚI DẠNG CẬP NHẬT ĐƠN GIẢN, QUY ƯỚC ĐẶT TÊN NHƯ SAU: DÙNG TIỀN TỐ “SET” KÈM THEO TÊN THÀNH VIÊN CẦN SỬA
 - `INT SETX(INT);`

TRUY VẤN VÀ CẬP NHẬT

- ❖ NẾU PHƯƠNG THỨC GET/SET CHỈ CÓ NHIỆM VỤ CHO TA ĐỌC/GHI GIÁ TRỊ CHO CÁC THÀNH VIÊN DỮ LIỆU → QUY ĐỊNH CÁC THÀNH VIÊN PRIVATE ĐỂ ĐƯỢC ÍCH LỢI GÌ?
 - NGOÀI VIỆC BẢO VỆ CÁC NGUYÊN TẮC ĐÓNG GÓI, TA CẦN KIỂM TRA XEM GIÁ TRỊ MỚI CHO THÀNH VIÊN DỮ LIỆU CÓ HỢP LỆ HAY KHÔNG.
 - SỬ DỤNG PHƯƠNG THỨC TRUY VẤN CHO PHÉP TA THỰC HIỆN VIỆC KIỂM TRA TRƯỚC KHI THỰC SỰ THAY ĐỔI GIÁ TRỊ CỦA THÀNH VIÊN.
 - CHỈ CHO PHÉP CÁC DỮ LIỆU CÓ THỂ TRUY VẤN HAY THAY ĐỔI MỚI ĐƯỢC TRUY CẬP ĐẾN.

THÀNH VIÊN TĨNH – STATIC MEMBER

- ❖ TRONG C, **STATIC** XUẤT HIỆN TRƯỚC DỮ LIỆU ĐƯỢC KHAI BÁO TRONG MỘT HÀM NÀO ĐÓ THÌ GIÁ TRỊ CỦA DỮ LIỆU ĐÓ VẪN ĐƯỢC LƯU LẠI NHƯ MỘT BIẾN TOÀN CỤC.
- ❖ TRONG C++, NẾU **STATIC** XUẤT HIỆN TRƯỚC MỘT DỮ LIỆU HOẶC MỘT PHƯƠNG THỨC **CỦA LỚP** THÌ GIÁ TRỊ CỦA NÓ VẪN ĐƯỢC LƯU LẠI VÀ **CÓ Ý NGHĨA CHO ĐỐI TƯỢNG KHÁC CỦA CÙNG LỚP NÀY.**
- ❖ CÁC THÀNH VIÊN **STATIC** CÓ THỂ LÀ **PUBLIC, PRIVATE** HOẶC **PROTECTED.**

THÀNH VIÊN TĨNH – STATIC MEMBER

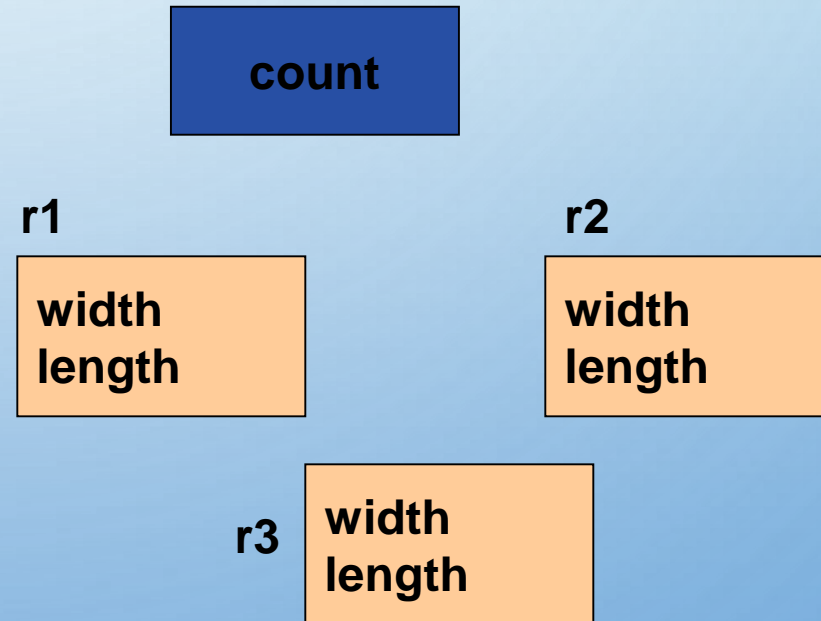
❖ ĐỐI VỚI CLASS, **STATIC** DÙNG ĐỂ **KHAI BÁO THÀNH VIÊN DỮ LIỆU DÙNG CHUNG CHO MỌI THỂ HIỆN CỦA LỚP**:

- MỘT BẢN DUY NHẤT TỒN TẠI TRONG SUỐT QUÁ TRÌNH CHẠY CỦA CHƯƠNG TRÌNH.
- DÙNG CHUNG CHO TẤT CẢ CÁC THỂ HIỆN CỦA LỚP.
- BẤT KỂ LỚP ĐÓ CÓ BAO NHIÊU THỂ HIỆN.

VÍ DỤ

```
class Rectangle
{
    private:
        int width;
        ➔ int length;
        static int count;
    public:
        void set(int w, int l);
        int area();
}
```

```
Rectangle r1;
Rectangle r2;
Rectangle r3;
```



THÀNH VIÊN TĨNH – STATIC MEMBER

❖ PHƯƠNG THỨC STATIC?

- ĐỐI VỚI CÁC PHƯƠNG THỨC STATIC, NGOÀI Ý NGHĨA TƯƠNG TỰ VỚI DỮ LIỆU, CÒN CÓ SỰ KHÁC BIỆT CƠ BẢN ĐÓ LÀ VIỆC CHO PHÉP TRUY CẬP ĐẾN CÁC PHƯƠNG THỨC STATIC KHI **CHƯA KHAI BÁO ĐỐI TƯỢNG** (THÔNG QUA TÊN LỚP)

THÀNH VIÊN TĨNH – STATIC MEMBER

- ❖ CÁC THÀNH VIÊN LỚP TĨNH PUBLIC CÓ THỂ ĐƯỢC TRUY CẬP THÔNG QUA BẤT KỲ ĐỐI TƯỢNG NÀO CỦA LỚP ĐÓ, HOẶC CHÚNG CÓ THỂ ĐƯỢC TRUY CẬP THÔNG QUA TÊN LỚP SỬ DỤNG TOÁN TỬ ĐỊNH PHẠM VI.
- ❖ CÁC THÀNH VIÊN LỚP TĨNH PRIVATE VÀ PROTECTED PHẢI ĐƯỢC TRUY CẬP THÔNG QUA CÁC HÀM THÀNH VIÊN PUBLIC CỦA LỚP HOẶC THÔNG QUA CÁC FRIEND CỦA LỚP.
- ❖ CÁC THÀNH VIÊN LỚP TĨNH TỒN TẠI NGAY CẢ KHI ĐỐI TƯỢNG CỦA LỚP ĐÓ KHÔNG TỒN TẠI.

THÀNH VIÊN TĨNH – STATIC MEMBER

- ❖ ĐỂ TRUY CẬP MỘT THÀNH VIÊN LỚP TĨNH PUBLIC KHI CÁC ĐỐI TƯỢNG CỦA LỚP KHÔNG TỒN TẠI, ĐƠN GIẢN THÊM VÀO ĐẦU TÊN LỚP VÀ TOÁN TỬ ĐỊNH PHẠM VI CHO THÀNH VIÊN DỮ LIỆU.
- ❖ ĐỂ TRUY CẬP MỘT THÀNH VIÊN LỚP TĨNH PRIVATE HOẶC PROTECTED KHI CÁC ĐỐI TƯỢNG CỦA LỚP KHÔNG TỒN TẠI, MỘT HÀM THÀNH VIÊN PUBLIC PHẢI ĐƯỢC CUNG CẤP VÀ HÀM PHẢI ĐƯỢC GỌI BỞI THÊM VÀO ĐẦU TÊN CỦA NÓ VỚI TÊN LỚP VÀ TOÁN TỬ ĐỊNH PHẠM VI.

VÍ DỤ VỀ ĐỐI TƯỢNG TOÀN CỤC

❖ XÉT ĐOẠN CHƯƠNG TRÌNH SAU:

```
#INCLUDE <Iostream.H>

VOID MAIN(){

    COUT << "HELLO, WORLD.\N";

}
```

❖ HÃY SỬA LẠI ĐOẠN CHƯƠNG TRÌNH TRÊN ĐỂ CÓ KẾT XUẤT:

```
ENTERING A C++ PROGRAM SAYING...

HELLO, WORLD.

AND THEN EXITTING...
```

❖ YÊU CẦU KHÔNG THAY ĐỔI HÀM MAIN() DƯỚI BẤT KỲ HÌNH THỨC NÀO.

VÍ DỤ VỀ ĐỐI TƯỢNG TOÀN CỤC

```
#include <iostream.h>

class Dummy{
public:
    Dummy(){cout << "Entering a C++ program saying...\n";}
    ~Dummy(){cout << "And then exiting...";}
};

Dummy A;

void main(){
    cout << "Hello, world.\n";
}
```