

LỚP VÀ ĐỐI TƯỢNG



LỚP TRONG C++

- ❖ MỘT LỚP BAO GỒM CÁC THÀNH PHẦN **DỮ LIỆU (THUỘC TÍNH)** VÀ CÁC **PHƯƠNG THỨC (HÀM THÀNH PHẦN)**.
- ❖ LỚP TRONG C++ THỰC CHẤT LÀ MỘT KIỂU DỮ LIỆU DO NGƯỜI SỬ DỤNG ĐỊNH NGHĨA.
- ❖ TRONG C++, DÙNG TỪ KHÓA **CLASS** ĐỂ CHỈ ĐIỂM BẮT ĐẦU CỦA MỘT LỚP SẼ ĐƯỢC CÀI ĐẶT.
- ❖ LỚP LÀ MỘT MÔ TẢ TRỪU TƯỢNG CỦA NHÓM CÁC ĐỐI TƯỢNG CÙNG BẢN CHẤT, NGƯỢC LẠI MỖI MỘT ĐỐI TƯỢNG LÀ MỘT **THỂ HIỆN** CỤ THỂ CHO NHỮNG MÔ TẢ TRỪU TƯỢNG ĐÓ.

CÚ PHÁP KHAI BÁO LỚP

Header

CLASS CLASS_NAME

{

ACCESS_CONTROL_LABEL:

MEMBERS;
(DATA & CODE)

ACCESS_CONTROL_LABEL:

MEMBERS;
(DATA & CODE)

};

class Rectangle

{

private:

int width;

int length;

public:

void set(int w, int l);

int area();

};

CÁC THÀNH PHẦN CỦA LỚP

- ❖ **THUỘC TÍNH:** CÁC THUỘC TÍNH ĐƯỢC KHAI BÁO GIỐNG NHƯ KHAI BÁO BIẾN TRONG C
- ❖ **PHƯƠNG THỨC:** CÁC PHƯƠNG THỨC ĐƯỢC KHAI BÁO GIỐNG NHƯ KHAI BÁO HÀM TRONG C. CÓ HAI CÁCH ĐỊNH NGHĨA THI HÀNH CỦA MỘT PHƯƠNG THỨC
 - ❖ **ĐỊNH NGHĨA THI HÀNH TRONG LỚP**
 - ❖ **ĐỊNH NGHĨA THI HÀNH NGOÀI LỚP**

CƠ CHẾ TẠO LẬP CÁC LỚP

- ❖ XÁC ĐỊNH CÁC THUỘC TÍNH (DỮ LIỆU)
 - NHỮNG GÌ MÀ TA BIẾT VỀ ĐỐI TƯỢNG – GIỐNG NHƯ MỘT STRUCT
- ❖ XÁC ĐỊNH CÁC PHƯƠNG THỨC (HÀNH VI)
 - NHỮNG GÌ MÀ ĐỐI TƯỢNG CÓ THỂ LÀM
- ❖ XÁC ĐỊNH CÁC QUYỀN TRUY XUẤT

ĐỊNH NGHĨA HÀM THÀNH PHẦN

❖ CÚ PHÁP ĐỊNH NGHĨA CÁC HÀM THÀNH PHẦN Ở BÊN NGOÀI KHAI BÁO LỚP:

<TÊN KIỂU GIÁ TRỊ TRẢ VỀ> <TÊN LỚP>::<TÊN HÀM>
(<DANH SÁCH THAM SỐ>)

{

<NỘI DUNG >

}

VÍ DỤ:

VOID POINT::DISPLAY() {

//.....

}

ĐỊNH NGHĨA HÀM THÀNH PHẦN

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

inline

class name

member function name

```
void Rectangle :: set (int w, int l)  
{  
    width = w;  
    length = l;  
}
```

scope operator

KHAI BÁO VÀ TẠO LẬP ĐỐI TƯỢNG

❖ KHAI BÁO VÀ TẠO ĐỐI TƯỢNG:

<TÊN LỚP> <TÊN ĐỐI TƯỢNG>;

❖ GỌI HÀM THÀNH PHẦN CỦA LỚP

<TÊN ĐỐI TƯỢNG>.<TÊN HÀM THÀNH PHẦN> (<DANH SÁCH CÁC THAM SỐ NẾU CÓ>);

<TÊN CON TRỞ ĐỐI TƯỢNG>→<TÊN HÀM THÀNH PHẦN> (<DANH SÁCH CÁC THAM SỐ NẾU CÓ>);

KHAI BÁO VÀ TẠO LẬP ĐỐI TƯỢNG

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

r1 is statically allocated

```
void main()
{
    Rectangle r1;
    ➡ r1.set(5, 8);
}
```

r1

**width = 5
length = 8**

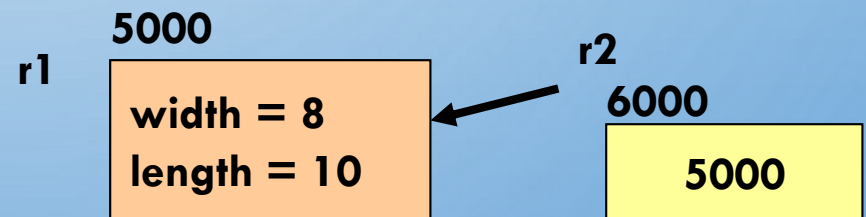
KHAI BÁO VÀ TẠO LẬP ĐỐI TƯỢNG

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

r2 is a pointer to a Rectangle object

```
main()
{
    Rectangle r1;
    r1.set(5, 8);           //dot notation

    Rectangle *r2;
    r2 = &r1;
    r2->set(8,10);         //arrow notation
}
```



KHAI BÁO VÀ TẠO LẬP ĐỐI TƯỢNG

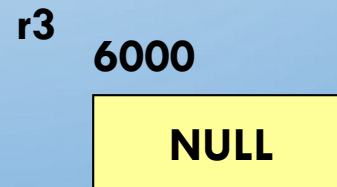
```
class Rectangle
{
    private:
        int width;
        int length;
    public:
        void set(int w, int l);
        int area();
};
```

r3 is dynamically allocated

```
main()
{
    Rectangle *r3;
    r3 = new Rectangle();

    r3->set(80,100); //arrow notation

    delete r3;
    ➡ r3 = NULL;
}
```

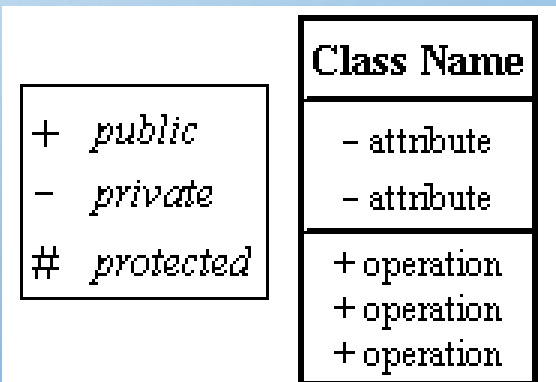


PHẠM VI TRUY XUẤT

- ❖ TRONG ĐỊNH NGHĨA CỦA LỚP TA CÓ THỂ XÁC ĐỊNH **KHẢ NĂNG TRUY XUẤT THÀNH PHẦN** CỦA MỘT LỚP NÀO ĐÓ TỪ BÊN NGOÀI PHẠM VI LỚP.
- ❖ **PRIVATE**, **PROTECTED** VÀ **PUBLIC** LÀ CÁC TỪ **KHÓA** XÁC ĐỊNH PHẠM VI TRUY XUẤT
- ❖ MỌI THÀNH PHẦN ĐƯỢC LIỆT KÊ TRONG PHẦN **PUBLIC** ĐỀU CÓ THỂ TRUY XUẤT TRONG **BẤT KỲ** HÀM NÀO.
- ❖ NHỮNG THÀNH PHẦN ĐƯỢC LIỆT KÊ TRONG PHẦN **PRIVATE** CHỈ ĐƯỢC TRUY XUẤT **BÊN TRONG PHẠM VI LỚP**.

PHẠM VI TRUY XUẤT

- ❖ TRONG LỚP CÓ THỂ CÓ NHIỀU NHÃN PRIVATE VÀ PUBLIC
- ❖ MỖI NHÃN NÀY CÓ PHẠM VI ẢNH HƯỞNG CHO ĐẾN KHI GẶP MỘT NHÃN KẾ TIẾP HOẶC HẾT KHAI BÁO LỚP.
- ❖ NHÃN PRIVATE ĐẦU TIÊN CÓ THỂ BỎ QUA VÌ C++ NGẦM HIỂU RẰNG CÁC THÀNH PHẦN TRƯỚC NHÃN PUBLIC ĐẦU TIÊN LÀ PRIVATE.



PHẠM VI TRUY XUẤT – VÍ DỤ

```
class TamGiac{  
    private:  
        float a,b,c;/*độ dài ba cạnh*/  
    public:  
        void Nhap();/*nhập vào độ dài ba cạnh*/  
        void In();/*in ra các thông tin liên quan đến tam giác*/  
    private:  
        int Loaitg();//cho biết kiểu của tam giác: 1-d,2-v,3-c,4-v,5-t  
        float DienTich();/*tính diện tích của tam giác*/  
};
```


THAM SỐ HÀM THÀNH PHẦN

```
void point::init (int xx, int yy){  
    x = xx;  
    y = yy; //x, y là thành phần của lớp point  
}
```



❖ **HÀM THÀNH PHẦN** CÓ QUYỀN TRUY NHẬP ĐẾN CÁC THÀNH PHẦN **PRIVATE** CỦA ĐỐI TƯỢNG GỌI NÓ

THAM SỐ HÀM THÀNH PHẦN

```
int Trung (point pt){  
    return (x==pt.x && y==pt.y);  
}  
  
int Trung (point *pt){  
    return (x==pt→x && y==pt→y);  
}  
  
int Trung (point &pt) {  
    return (x==pt.x && y==pt.y);  
}
```

Hàm thành phần
có quyền truy cập
đến tất cả các
thành phần
private của các
đối tượng, tham
chiếu đối tượng
hay con trỏ đối
tượng có cùng
kiểu lớp khi được
dùng là tham số
hình thức của nó.

CON TRỎ THIS

- ❖ TỪ KHÓA **THIS** TRONG ĐỊNH NGHĨA CỦA CÁC HÀM THÀNH PHẦN LỚP DÙNG ĐỂ XÁC ĐỊNH ĐỊA CHỈ CỦA ĐỐI TƯỢNG DÙNG LÀM **THAM SỐ NGẦM ĐỊNH** CHO HÀM THÀNH PHẦN.
- ❖ **CON TRỎ THIS** THAM CHIẾU ĐẾN ĐỐI TƯỢNG ĐANG GỌI HÀM THÀNH PHẦN.
- ❖ VÍ DỤ:

```
INT POINT::TRUNG(POINT PT){  
    RETURN (THIS → X == PT.X && THIS → Y == PT.Y);  
}
```

PHÉP GÁN ĐỐI TƯỢNG

❖ LÀ VIỆC SAO CHÉP GIÁ TRỊ CÁC THÀNH PHẦN DỮ LIỆU TỪ ĐỐI TƯỢNG A SANG ĐỐI TƯỢNG B TƯƠNG ỨNG TỪNG ĐÔI MỘT

❖ VÍ DỤ:

POINT A, B;

A.INIT(5,2);

B = A;

