22 Static for Classes and Structs in C++

效果见：类内static

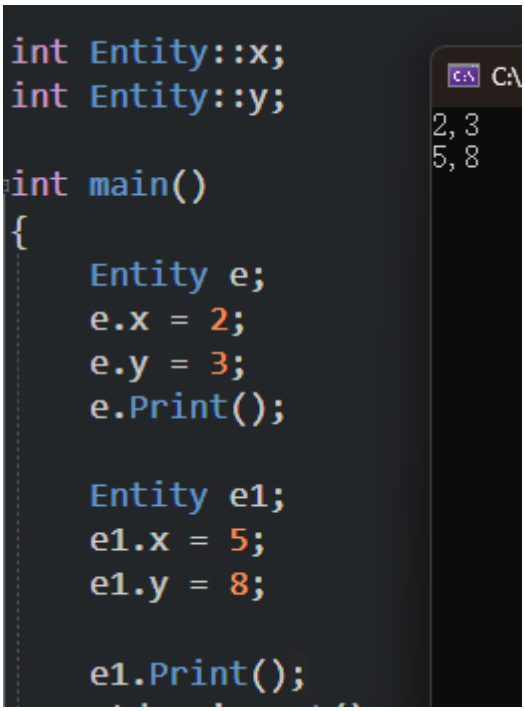
如果你在类中创建了一个static变量，则这个类的所有实例中，这个变量只有一个实例。
同样，如果一个实例修改了这个变量，则这个改变会体现在所有的类实例中。
就像时这个类的*global*实例

C++

```cpp
struct Entity  // 用struct是想默认public
{
    int x, y;

    void Print()
    {
        std::cout << x << "," << y << std::endl;
    }
};

int main()
{
    Entity e;
    e.x = 2;
    e.y = 3;

    Entity e1 = { 5,8 };  // Initializer

    e.Print();   // 2, 3
    e1.Print();  // 5, 8
    std::cin.get();
}
```

C++

```cpp
struct Entity  // 用struct是想默认public
{
    static int x, y;

    void Print()
    {
        std::cout << x << "," << y << std::endl;
    }
};

int Entity::x;
int Entity::y;

int main()
{
    Entity e;
    e.x = 2;  // Entity::x;
    e.y = 3;  // Entity::y;
    e.Print();

    Entity e1;
    Entity.x = 5;
    Entity.y = 8;

    e1.Print();
    std::cin.get();
}

    Entity e1;
    Entity::x = 5;
    Entity::y = 8;

    e1.Print();
    std::cin.get();
}
```



*static method*不能访问非静态变量。你在类里写的每个非静态方法都会获得当前的类实例作为参数（this指针）

```C++
static void Print(Entity e)  //给一个引用
{
    std::cout << e.x << "," << e.y << std::endl;
}
```