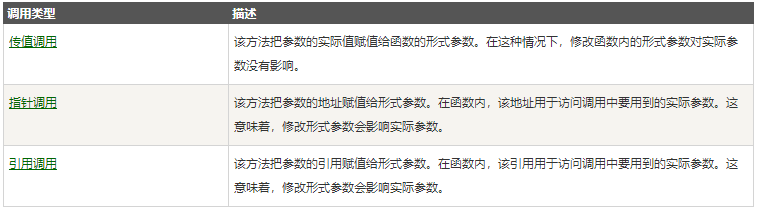


在程序中，局部变量和全局变量的名称可以相同，但是在函数内，局部变量的值会覆盖全局变量的值。

后缀是 U 和 L 的组合，U 表示无符号整数（unsigned），L 表示长整数（long）,F浮点数。





<https://www.runoob.com/cplusplus/cpp-functions.html>



**<cmath>: 平方根、**绝对值、**三**角函数等**内置函数。**



**字符串：**



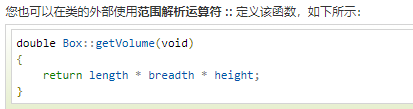
NULL 指针是一个定义在标准库中的值为零的常量。

类定义后必须跟着一个分号。

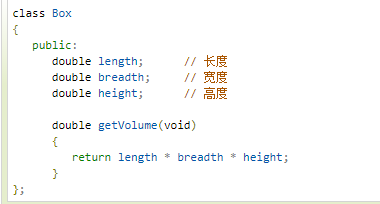
私有的成员和受保护的成员不能使用直接成员访问运算符 (.) 来直接访问。

类的函数定义：

在类的外部使用类的域限定符：



内部直接定义：



**构造/析构函数：**

类的**构造函数**是类的一种特殊的成员函数，它会在每次创建类的新对象时执行。

构造函数的名称与类的名称是完全相同的，构造函数可用于为某些成员变量设置初始值。

类的**析构函数**是类的一种特殊的成员函数，它会在每次删除所创建的对象时执行。

只是在前面加了个波浪号（~）作为前缀，它不会返回任何值，也不能带有任何参数

？？：如果这个类中存在在堆内存中申请的动态内存，且希望在对象销毁时堆内存也能得到释放，则必须写析构函数，在函数中用语句显式释放堆内存。如果不存在在堆内存中的动态分配，析构函数就不需要编写。Person为定义的类

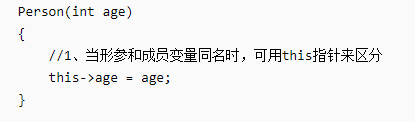
Person p1(10);括号法

Person p2 = Person(10);显示

Person p3 = 10;隐式

* 拷贝构造：使用一个已经创建完毕的对象来初始化一个新对象
* Person(const Person& p) {
* age = p.age;
* cout << "拷贝构造函数!" << endl;
* }
* Person p2(p3);//p3是已经初始化完成的对象
* 或者Person p2 = p3；//此方式必须在定义的同时赋予已经实例化的对象只有非静态成员变量才属于类的对象上，占用对象空间

this指针是隐含每一个非静态成员函数内的一种指针：

****

**初始化列表：**

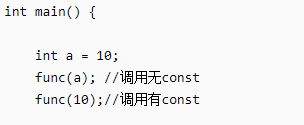
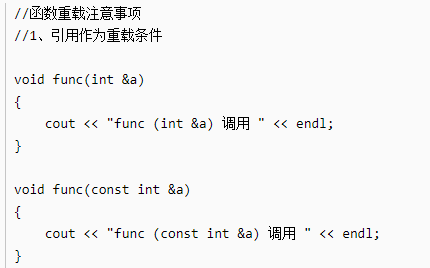


我们可以使用 **static** 关键字来把类成员定义为静态的。静态成员都只有一个副本。静态成员在类的所有对象中是共享的。如果不存在其他的初始化语句，在创建第一个对象时，所有的静态数据都会被初始化为零。我们不能把静态成员的初始化放置在类的定义中，但是可以在类的外部通过使用范围解析运算符 **::** 来重新声明静态变量从而对它进行初始化。

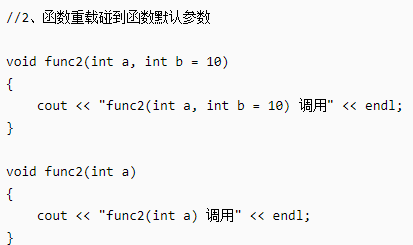
**C++ 中的函数重载：**

在同一个作用域内，可以声明几个功能类似的同名函数，但是这些同名函数的形式参数（指参数的个数、类型或者顺序）必须不同。您不能仅通过返回类型的不同来重载函数。

* 函数返回值不可以作为函数重载条件,即函数参数**类型不同** 或者 **个数不同** 或者 **顺序不同，只是返回值类型不同会报错，如int fun()与void fun()**



int a = 10；int &b = a;（ok）、、、int &a = 10；（x）、、、const int &a = 10（ok）





int \*p = new int(10); delete p;

int \*arr = new int[10]; delete[] arr;

