C++ 函数参数默认值

Void fun(int x,int y=10,int k=18)

Void fun(int x,int y=10,int k ) 错误，有默认参数值的参数必须在参数列表的最右端

传参数的时候，无实参，则用默认的参数，有实参的话，就用实参覆盖默认的实参

1.const 修饰类的成员变量，表示成员常量，不能被修改。

2.const修饰函数承诺在本函数内部不会修改类内的数据成员，不会调用其它非 const 成员函数。

3.如果 const 构成函数重载，const 对象只能调用 const 函数，非 const 对象优先调用非 const 函数。

4.const 函数只能调用 const 函数。非 const 函数可以调用 const 函数。

5.类体外定义的 const 成员函数，在定义和声明处都需要 const 修饰符。

cosnt 成员函数主要目的是防止成员函数修改对象的内容。即 const 成员函数不能修改成员变量的值，但可以访问成员变量。

static 成员函数主要目的是作为类作用域的全局函数。不能访问类的非静态数据成员。类的静态成员函数没有 this 指针，这导致：

1、不能直接存取类的非静态成员变量，调用非静态成员函数。

2、不能被声明为 virtual。

**1、类里的const成员初始化：**

在一个类里建立一个 const 时，不能给他初值。

class foo{ public:

foo():i(100){}

private:

const int i=100;//error!!! };

//或者通过这样的方式来进行初始化 foo::foo():i(100){}

**2、类里的 static 成员初始化：**

类中的 static 变量是属于类的，不属于某个对象，它在整个程序的运行过程中只有一个副本，因此不能在定义对象时 对变量进行初始化，就是不能用构造函数进行初始化，其正确的初始化方法是：

数据类型 类名::静态数据成员名=值；

class foo{ public: foo(); private: staticint i; }; intfoo::i=20;

这表明：

1、初始化在类体外进行，而前面不加static，以免与一般静态变量或对象相混淆

2、初始化时不加该成员的访问权限控制符private、public等

3、初始化时使用作用域运算符来表明它所属的类，因此，静态数据成员是类的成员而不是对象的成员

**3、类里的 static cosnt 和 const static 成员初始化**

这两种写法的作用一样，为了便于记忆，在此值说明一种通用的初始化方法：

class Test{ public:

static const int mask1;

const static int mask2; };

const int Test::mask1=0xffff; const int Test::mask2=0xffff; //它们的初始化没有区别，虽然一个是静态常量一个是常量静态。静态都将存储在全局变量区域，其实最后结果都一样。可能在不同编译器内，不同处理，但最后结果都一样。