# C++的类的继承与多态

1. 是不是一个父类写了一个virtual 函数，如果子类覆盖它的函数不加virtual ,也能实现多态?
2. 构造、拷贝构造和赋值运算符的概念

3、struct C

{

C()

{

cout << "construct: C()" << endl;

i = 0;

}

C(const C &c) {

cout << "construct: C(const C &c)" << endl;

}

C(int i) {

cout << "construct: C(int i)" << endl;

this->i = i;

}

~C() {

}

C& operator = (const C &c) {

cout << "operator = (const C &c)" << endl;

i = c.i;

return \*this;

}

int i;

};

void test()

{

C \*c;

c = new C(100);

C c2 = \*c;

C c3;

c3 = c2;

}

4、四行输出的结果依次顺序为：

a. construct: C() b. construct: C(const C &c)

c. construct: C(int i) d. operator = (const C &c)

则输出的正确顺序为\_\_\_\_\_\_\_\_

1. c b a d; B. a c b d C. c a b d D. d b a c

5、请选择下列程序的输出结果\_\_\_\_\_\_\_\_

class A

{

public:

void PrintOne(){cout<<” A”};

virtual void PrintSecond(){cout<<” A”};

};

class B: public A

{

public:

virtual void PrintOne(){cout<<” B”};

virtual void PrintSecond(){cout<<”B”};

};

void main()

{

A\* a= new B;//

B\* b=new B;

1. PrintOne(); a.PrintSecond();

b.PrintOne(); b.PrintSecond();

}

A. A A B B B. B B B B

C. B B A B D. A B B B

1. 函数的各析构函数可以继承吗？派生类构造函数中各部分的执行次序是怎样的？

7、下列运算符函数中，肯定不属于类Value的成员函数的是（）

1. Value operator/(Value)
2. Value operator-(Value,Value)
3. Value operator+(Value)
4. Value operator\*(Value)
5. 重载方法和重析构方法的区别
6. static在C和c++中表示什么？

10、 c++的模板类有什么用？

# C++的类的基础语法(修饰符)

1. C中已经有malloc/free ,为什么在c++中还需要new/delete，并指出
2. malloc/free new/deletede 区别。
3. 什么是“引用”？声明和使用“引用”要注意哪些问题？
4. 宏定义、内联函数、和普通函数的区别

宏和内联函数的区别:

带参数的宏定义并不对参数进行运算，而是直接替换;

内联函数先把参数表达式进行运算求值，然后把表达式的值传递给形式参数

内联函数的参数类型和返回值类型在声明中都有明确的指定；而带参数

宏定义的参数没有类型的概念

宏和普通函数的区别

<1>宏做的是简单的字符串替换; 函数的参数的传递,参数是有数据类型的,可以是各种各样的类型.

<2>宏的参数替换是不经计算而直接处理的,而函数调用是将实参的值传递给形参,既然说是值,自然是计算得来的.

<3> ,宏占用的是编译的时间,而函数占用的是执行时的时间.

<4>宏的参数是不占内存空间的, 函数调用时的参数传递则是具体变量之间的信息传递,形参作为函数的局部变量,显然是占用内存的.

<5>函数的调用是需要付出一定的时空开销的,因为系统在调用函数时,要保留现场,然后转入被调用函数去执行,调用完, 再返回主调函数,此时再恢复现场,这些操作,显然在宏中是没有的.

所谓"内联函数"就是将很简单的函数"内嵌"到调用他的程序代码中,只样做的目的是为了避免上面说到的第5点,目的旨在节约下原本函数调用时的时空开销.但必须注意的是:作为内联函数,函数体必须十分简单,不能含有循环、条件、选择等复杂的结构，否则就不能做为内联函数了。

1. C ，C++中struct有什么区别？

5、使用纯虚拷贝构造函数时可能会出现什么异常

6、传常数、传指针、传引用是什么

7、static在C和c++中表示什么？

8、哪一个不是C++11中新增的语法（）

1. 右键引用
2. Lamdba表达式
3. 智能指针
4. Rang-base for
5. 静态成员函数没有（返回值/ this指针/指针参数/返回类型）？

10、以下代码中的输出语句能输出吗?

struct CLS

{

int m\_i;

CLS(int i):m\_i(i){};

CLS()

{

CLS(0);

}

};

CLS obj;

cout << obj.m\_i <<endl;

**11、重载和重写，以及隐藏的区别：**

Overload(重载)：在C++程序中，可以将语义、功能相似的几个函数用同一个名字表示，但参数或返回值不同（包括类型、顺序不同），即函数重载。

（1）相同的范围（在同一个类中）；

（2）函数名字相同；

（3）参数不同；

（4）virtual 关键字可有可无。Override(覆盖)：是指派生类函数覆盖基类函数，特征是：

（1）不同的范围（分别位于派生类与基类）；

（2）函数名字相同；

（3）参数相同；

（4）基类函数必须有virtual 关键字。  
**注：重写基类虚函数的时候，会自动转换这个函数为virtual函数，不管有没有加virtual，因此重写的时候不加virtual也是可以的，不过为了易读性，还是加上比较好。**Overwrite(重写)：隐藏，是指派生类的函数屏蔽了与其同名的基类函数，规则如下：

（1）如果派生类的函数与基类的函数同名，但是参数不同。此时，不论有无virtual关键字，基类的函数将被隐藏（注意别与重载混淆）。

（2）如果派生类的函数与基类的函数同名，并且参数也相同，但是基类函数没有virtual关键字。此时，基类的函数被隐藏（注意别与覆盖混淆）。