



实验UNIT 07 类的继承

《程序设计》课程组



第7讲上机实验

实验目的：

1. 学习声明和使用类的继承关系，声明派生类；
2. 熟悉不同继承方式下对基类成员的访问控制；
3. 学习利用虚基类解决二义性问题。



第7讲上机实验

实验任务：

1. 课堂练习：派生类的构造函数，多继承的构造函数
2. 编程练习：类的继承关系



第7讲上机实验

◆ 实验步骤提示：

1. 为每个题目建立一个新的控制台项目文件；
2. 向其中提交程序所需的头文件、源程序文件；
3. 选择菜单“生成解决方案”编译源程序；
4. 执行程序，观察输出结果是否正确观察输出结果是否正确，如果有错误，可以执行第5步；
5. 使用debug功能：跟踪观察数组的数组元素值、指针及其指向对象的值变化是否正确？



现在开始课堂练习！

练习内容：派生类构造函数、

多继承的构造函数



练习1：派生类构造函数的构造规则

```
#include <iostream>
using namespace std;
class Base
{ public:
    Base() { a=0; }
    Base( int i) { a=i; }
protected:
    int a;
};
class Derived:public Base
{
```



练习1：派生类构造函数的构造规则

```
public:
    Derived() { b=0; }
    Derived( int i) { b=i; }
    void Print() { cout<<"a="<<a<<",b="<<b<<endl; }
private:
    int b;
};
int main( )
{   Derived d1;   Derived d2(12);
    d1.Print(); d2.Print();
    cin.ignore(); return 0;
}
```

请分析此程序的运行结果，并上机验证！

练习2：派生类构造函数的构造规则

// 问题1：如果基类不定义任何构造函数，会怎么样？

```
#include <iostream>
using namespace std;
class Base
{ public:
    //      Base() { a=0; }
    //      Base( int i) { a=i;
}
    protected:
        int a;
};
class Derived:public Base
{
```


练习2：派生类构造函数的构造规则

```
public:
    Derived() { b=0; }
    Derived( int i) { b=i; }
    void Print() { cout<<"a="<<a<<",b="<<b<<endl; }
private:
    int b;
};

int main( )
{   Derived d1;   Derived d2(12);
    d1.Print(); d2.Print();
    cin.ignore(); return 0;
}
```

如果基类不定义任何构造函数，请分析此程序运行会出现什么现象？为什么？

练习3：派生类构造函数的构造规则

// 问题2：如果基类只定义带参数的构造函数，会怎么样？

```
#include <iostream>
using namespace std;
class Base
{ public:
//      Base() { a=0; }
      Base( int i) { a=i; }
protected:
      int a;
};
class Derived:public Base
{
```

练习3：派生类构造函数的构造规则

```
public:
    Derived() { b=0; }
    Derived( int i) { b=i; }
    void Print() { cout<<"a="<<a<<",b="<<b<<endl; }
private:
    int b;
};

int main()
{   Derived d1;   Derived d2(12);
    d1.Print(); d2.Print();
    cin.ignore(); return 0;
}
```

如果基类只定义带参数的构造函数，请分析此程序运行会出现什么现象？为什么？

练习4：多继承的构造函数

// 多继承方式下构造函数和析构函数的调用顺序。

```
#include <iostream>
```

```
using namespace std;
```

```
class A          //定义基类A
```

```
{
```

```
public:
```

```
    A(int i){a=i;cout<<"A Constructor"<<endl;}
```

```
    void disp(){cout<<"a="<<a<<endl;}
```

```
    ~A(){cout<<"A Destructor"<<endl;}
```

```
private:
```

```
    int a;
```

```
};
```

练习4：多继承的构造函数

```
class B           //定义基类B
{
public:
    B(int j){b=j;cout<<"B Constructor"<<endl;}
    void disp(){cout<<"b="<<b<<endl;}
    ~B(){cout<<"B Destructor"<<endl;}
private:
    int b;
};
```



练习4：多继承的构造函数

```
class C: public B, public A { //定义A和B的派生类C。B在前，A在后
public:
    C(int k):A(k+2),B(k-2) { //包含基类成员初始化列表
        c=k;
        cout<<"C Constructor"<<endl;
    }
    void disp() { //用类名加作用域运算符限定调用某个基类的同名成员
        A::disp();
        B::disp();
        cout<<"c="<<c<<endl;
    }
    ~C(){cout<<"C Destructor"<<endl;}
private:  int c;
};
```


练习4：多继承的构造函数

```
int main()
{
    C obj(10);
    //调用类C的成员函数disp
    obj.disp();
    cin.ignore();
    return 0;
}
```

请分析此程序的运行结果，并上机验证！



练习5：多继承的构造函数 (构造函数的调用次序)

```
// 派生类构造函数举例（多继承，含有内嵌对象）。  
#include <iostream>  
using namespace std;  
class B1{                //基类B1，构造函数有参数  
public: B1(int i) {cout<<"constructing B1 "<<i<<endl;}  
};  
class B2{                //基类B2，构造函数有参数  
{ public:      B2(int j) {cout<<"constructing B2 "<<j<<endl;}  
};  
class B3 {                //基类B3，构造函数无参数  
public: B3 ( ) {cout<<"constructing B3 *"<<endl;}  
};
```

练习5：多继承的构造函数 (构造函数的调用次序)

```
class C: public B2, public B1, public B3
{
public:           //派生类的公有成员
                //注意基类名的个数与顺序
                //注意成员对象名的个数与顺序
    C(int a, int b, int c, int d) :
        B1(a),memberB2(d),memberB1(c),B2(b){ }
private:        //派生类的私有对象成员
    B1 memberB1;
    B2 memberB2;
    B3 memberB3;
};
```

练习5：多继承的构造函数 (构造函数的调用次序)

```
int main ( )  
{  
    C obj(1,2,3,4);  
    cin.ignore();  
    return 0;  
}
```

请分析此程序的运行结果，并上机验证！



本次课堂练习结束！



上机编程练习任务

练习内容：类的继承和派生



第7讲上机任务

编程练习：

学生用书 实验7中实验任务 (1) (2) (3) (4) 。

7-5、定义一个基类Shape，在此基础上派生出Rectangle和Circle，二者带有getArea()函数计算对象的面积。使用Rectangle类创建一个派生类Square()。请自行完善各类的数据成员和成员函数以及主函数测试语句。

7-11、定义一个基类BaseClass，从它派生出DerivedClass，BaseClass有成员函数fn1()、fn2()，DerivedClass也有成员函数fn1()、fn2()，在主函数中声明一个DerivedClass的对象，分别用DerivedClass的对象以及BaseClass和DerivedClass的指针来调用fn1()、fn2()，观察运行结果。

请自行完善各成员函数函数体的功能以及主函数测试语句。



第7讲上机任务

编程练习：

7-15、下面的程序能得到预期的结果吗？如何避免类似问题的发生？

```
#include<iostream>
using namespace std;
struct Base1{int x;};
struct Base2{float y;};
struct Derived:Base1,Base2{};
int main(){
    Derived *pd=new Derived;
    pd->x=1; pd->y=2.0f;
    void *pv=pd;
    Base2 *pb=static_cast<Base2 *>(pv);
    cout<<pd->y<<" "<<pb->y<<endl;
    delete pb;
    return 0;
}
```



本讲结束

