#include<iostream>using namespace std;class Base{public: void setx(int i) { x=i; } int getx() { return x; }public: int x;};class Derived:public Base{public: void sety(int i) { y=i; } int gety() { return y; } void show() { cout<<"Base::x="<<x<<endl; }public: int y;};int main(){ Derived bb; bb.setx(16);

bb.sety(25);

bb.show();

cout<<"Base::x="<<bb.x<<endl;

cout<<"Derived::y="<<bb.y<<endl;

cout<<"Base::x="<<bb.getx()<<endl;

cout<<"Derived::y="<<bb.gety()<<endl;

return 0;}



**实验总结**

C++中派生类对基类成员的访问形式主要有以下两种：  
1、内部访问：由派生类中新增成员对基类继承来的成员的访问。  
2、对象访问：在派生类外部，通过派生类的对象对从基类继承来的成员的访问。

1、私有继承的访问规则

当类的继承方式为私有继承时，基类的public成员和protected成员被继承后成为派生类的private成员，派生类的其它成员可以直接访问它们，但是在类的外部通过派生类的对象无法访问。基类的private成员在私有派生类中是不可直接访问的，所以无论是派生类的成员还是通过派生类的对象，都无法直接访问从基类继承来的private成员，但是可以通过基类提供的public成员函数间接访问。

2、公有继承的访问规则

当类的继承方式为公有继承时，基类的public成员和protected成员被继承到派生类中仍作为派生类的public成员和protected成员，派生类的其它成员可以直接访问它们。但是，类的外部使用者只能通过派生类的对象访问继承来的public成员。基类的private成员在私有派生类中是不可直接访问的，所以无论是派生类成员还是派生类的对象，都无法直接访问从基类继承来的private成员，但是可以通过基类提供的public成员函数直接访问它们。

3、保护继承的访问规则

当类的继承方式为保护继承时，基类的public成员和protected成员被继承到派生类中都作为派生类的protected成员，派生类的其它成员可以直接访问它们，但是类的外部使用者不能通过派生类的对象访问它们。基类的private成员在私有派生类中是不可直接访问的，所以无论是派生类成员还是通过派生类的对象，都无法直接访问基类中的private成员。