程序代码：

#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;

}// 语句1

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;

}

实验结果：

Base::X=16

Base::X=16

Derived::y=25

Base::X=16

Derived::y=25

心得体会：  
尽量去熟练掌握每一种继承方式，在编写程序中做到灵活使用，来提高编程的效率，更好的熟练使用和创新。

Copyright ©2021-2099 YuenZhu. All rights reserved