**心得：**

这次的实验也收获颇多，从理论和实践得来的只是很多，多重继承时参数的调用方法和上面一样，只要在函数名前加上范围，调用成员变量也是一样。基类中的所有 public 成员在派生类中均为 private 属性；基类中的所有 protected 成员在派生类中均为 private 属性；基类中的所有 private 成员在派生类中不能使用。

**代码：**#include<iostream>

#include<cmath>

#include<string>

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; // 语句2

bb.setx(16); // 语句3

bb.sety(25); // 语句4

bb.show(); // 语句5

cout << "Base::x=" << bb.x << endl; // 语句6

cout << "Derived::y=" << bb.y << endl; // 语句7

cout << "Base::x=" << bb.getx() << endl; // 语句8

cout << "Derived::y=" << bb.gety() << endl; // 语句9

}

**程序结果：**

