**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; // 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

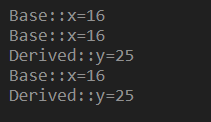
cout<<"Derived::y="<<bb.gety()<<endl; // 语句9

return 0;

}

**程序结果：**

更改前：



将x改为私有后：

语句1和6报错。原因为基类的私有成员x不可被派生类Derived和派生类对象bb访问，仅能被基类自己的成员函数getx访问，因此语句8没有出错。

将x改为保护后：

语句6报错。原因为基类的保护成员x不可被派生类对象bb访问，仅能被派生类Derived和基类自己的成员函数getx访问，因此语句1和8没有出错。

将派生类改为私有继承后：

语句3,6,8报错。基类中的所有公有成员均变为私有访问属性，即仅可内部访问（派生类对其访问），而对象bb则无法访问。

将派生类改为保护继承后：

语句3,6,8报错。基类中的所有公有成员均变为保护访问属性，即仅可内部访问（派生类对其访问），而对象bb则无法访问。

**感想心得：**

本次上机实践之后，我对于继承方式和类中成员的声明属性有了更加深刻清晰的认识