#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;                             // 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

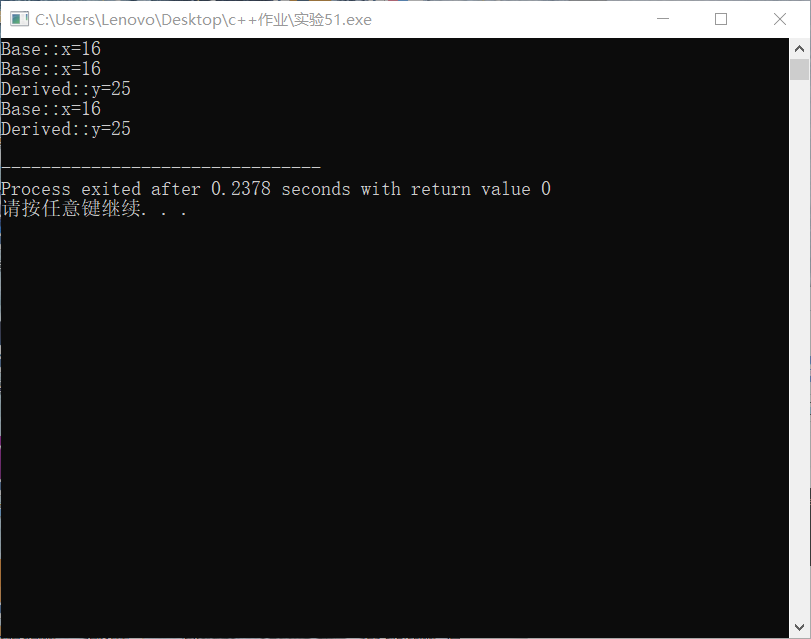
    cout << "Derived::y=" << bb.y << endl;  // 语句7

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

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

    return 0;

}



将基类Base中数据成员x的访问权限改为private时，会出现哪些错误，为什么？

会出现[错误] 'int Base::x' is private within this context，私有成员不能直接访问，只能通过访问含有该私有成员的公有成员函数间接访问。

将基类Base中数据成员x的访问权限改为protected时，会出现哪些错误，为什么？

会出现[错误] 'int Base::x' is protected within this context，保护成员不能直接访问，只能通过访问含有该成员的公有成员函数间接访问或在派生类内访问。

在原程序的基础上，将派生类Derived的继承方式改为private时，会出现哪些错误，为什么？

会出现[错误] 'void Base::setx(int)' is inaccessible within this context，int Base::x' is protected within this context。等错误，继承方式改成私有时，基类的所有成员都以私有成员的身份出现在派生类中，不能直接访问。

在原程序的基础上，将派生类Derived的继承方式改为protected时，会出现哪些错误，为什么

[错误] 'void Base::setx(int)' is inaccessible within this context，[错误] 'int Base::x' is protected within this context等，继承方式改成保护时，基类中私有成员还是以私有成员的身份出现在派生类中，保护成员和公有成员以保护成员的身份出现在派生类中，只能在类内访问，类外不能访问。