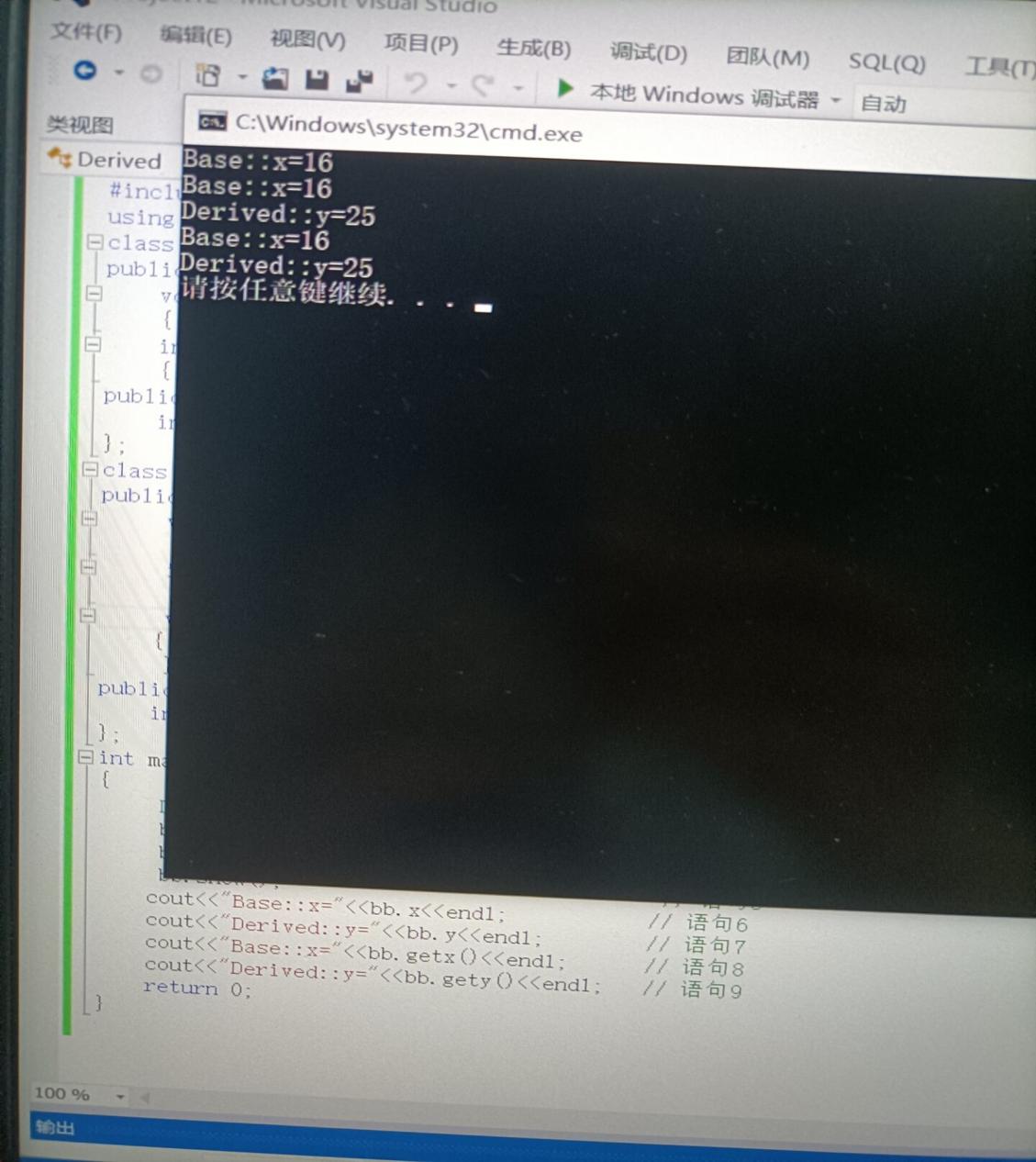
五2021.11.19周五上机

1.#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;  
}



2.实验感想

注意函数的派生，基类的公有成员经过私有派生后，在派生类中变为私有成员。  
类的“不可访问”成员指的是类的成员函数内部也不能访问的成员。例如，基类的私有成员在派生类的成员函数内就不能访问，但它依然是派生类的成员，因此说基类的私有成员在派生类中成为不可访问成员。  
使用公有派生，基类的公有成员将成为派生类的公有成员；基类的私有部分也将成为派生类的一部分，但属于“不可访问”成员，只能通过基类的公有和保护方法访问。  
一般情况下都应使用公有派生。