第五次上机实验

**1、程序代码**

**实验一**

#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

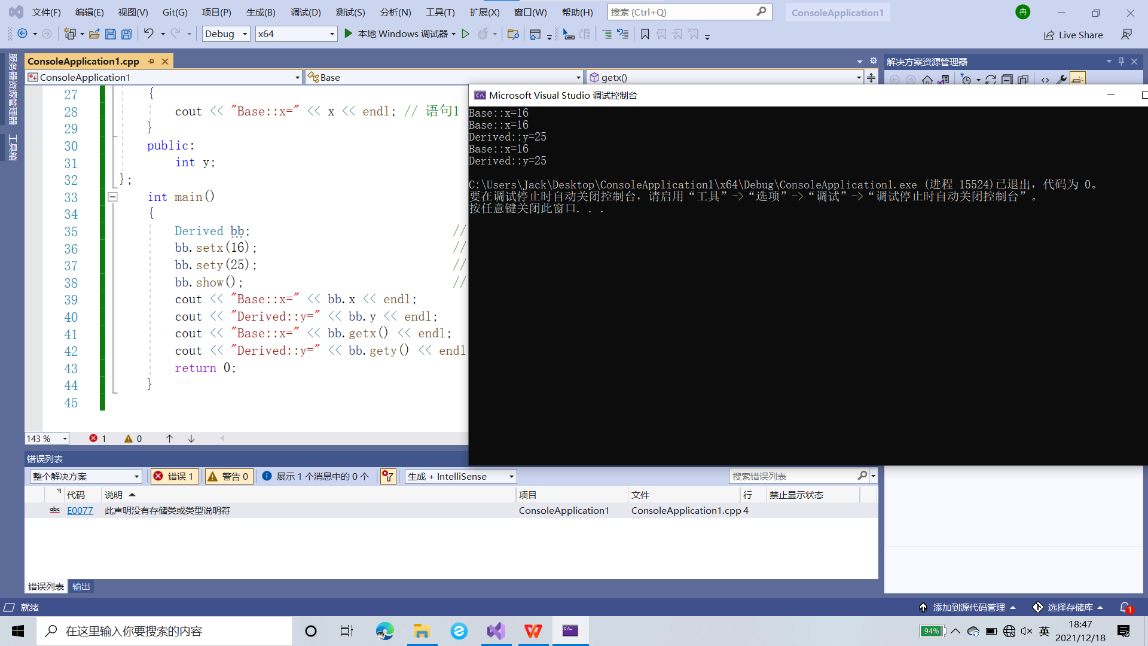
cout << "Base::x=" << bb.getx() << endl; // 语句8

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

return 0;

}

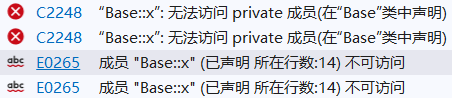
**2、程序结果**

**实验一**

**3、感想心得**

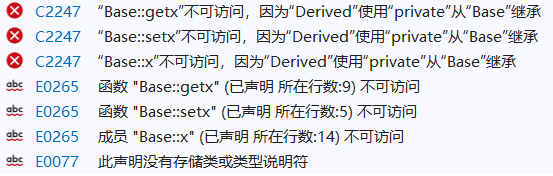
本次实验主要是解决几个调试错误的问题。

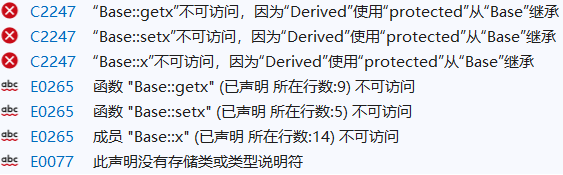
一、将基类Base中数据成员x的访问权限改为private时，会出现哪些错误，为什么？

派生类不可访问基类中的私有成员

二、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？

派生类无法通过对象访问基类中的保护成员。

三、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？因为将继承方式改为私有继承后，从派生类调用基类中的函数时，无法调用，因为派生类对象不能访问基类中的公有成员。

四、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？因为将继承方式改为保护继承后，从派生类调用基类中的函数时，无法调用，因派生类对象不能访问基类中的公有成员。

本次实验让我明白了派生类当中的三种继承的访问关系，在做题时，能够很熟练的解决程序出错误的问题，得出对应部分的正确访问关系。