Copyright ©2021-2099 FengJiaHe. All rights reserved

**例题/实验：**

#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;

bb.setx(16);

bb.sety(25);

bb.show();

cout << "Base::x=" << bb.x << endl;

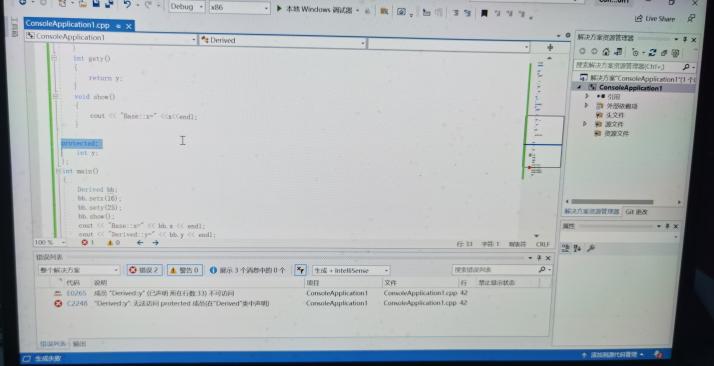
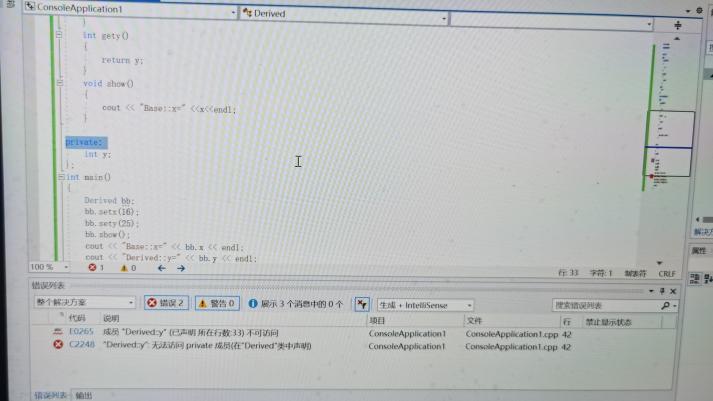
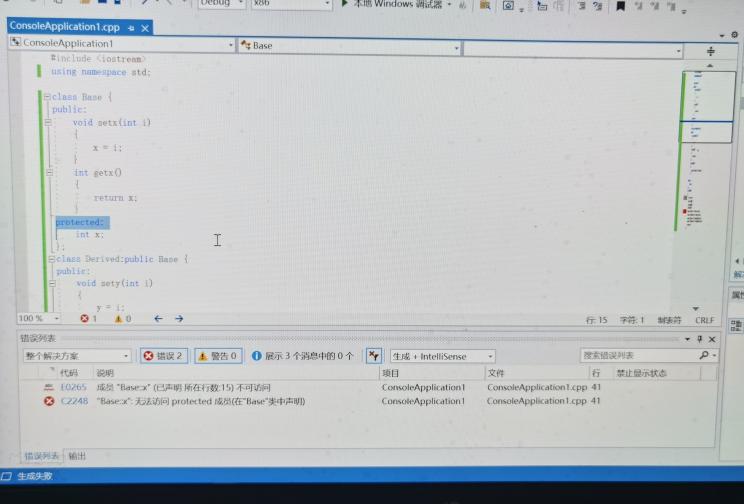
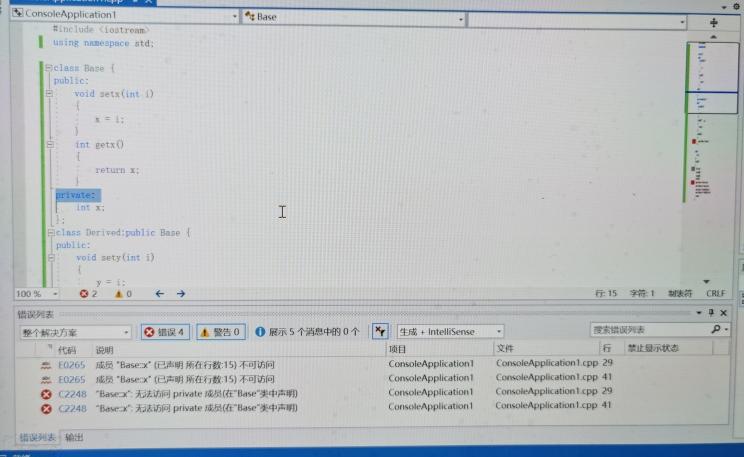
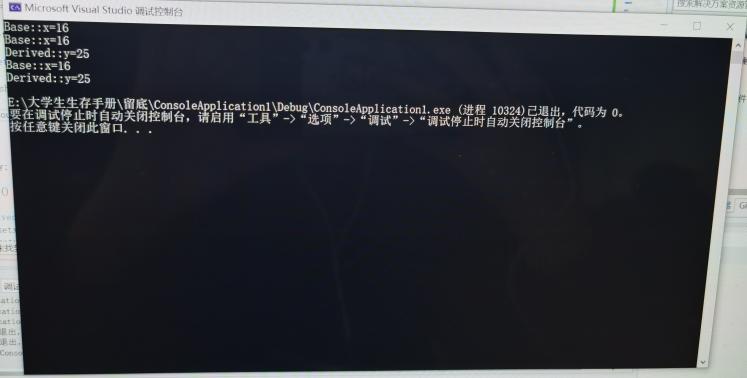
cout << "Derived::y=" << bb.y << endl;

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

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

return 0;

}



**感想心得：**

本次实验较为简单，通过实验，我掌握了派生类的声明方法和派生类构造函数的定义方法，了解不同方式下基类成员在派生类中的访问属性。

通过实操，我修改原程序基类Base中数据成员的访问权限和派生类Derived的继承方式，并观察调试中出现的错误及错误原因，加深了对派生类与继承的理解。