实验报告五（2021.11.19）

1. **实验目的**

1.掌握派生类的声明方法和派生类构造函数的定义方法

2.掌握不同方式下，基类成员在派生类中的访问属性

1. **实验内容**

按照以下要求，对程序进行修改后再调试，指出调试中出错的原因。

1. 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？

**会出现编译报错，因为改为私有变量不能对象访问。**

1. 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？

**会出现编译报错，因为是保护类型的数据。**

1. 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？

**会出现编译报错，因为是以私有方式登录，派生类中所有成员不能访问。**

1. 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？

**会出现编译报错，因为是以保护方式继承。**

#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

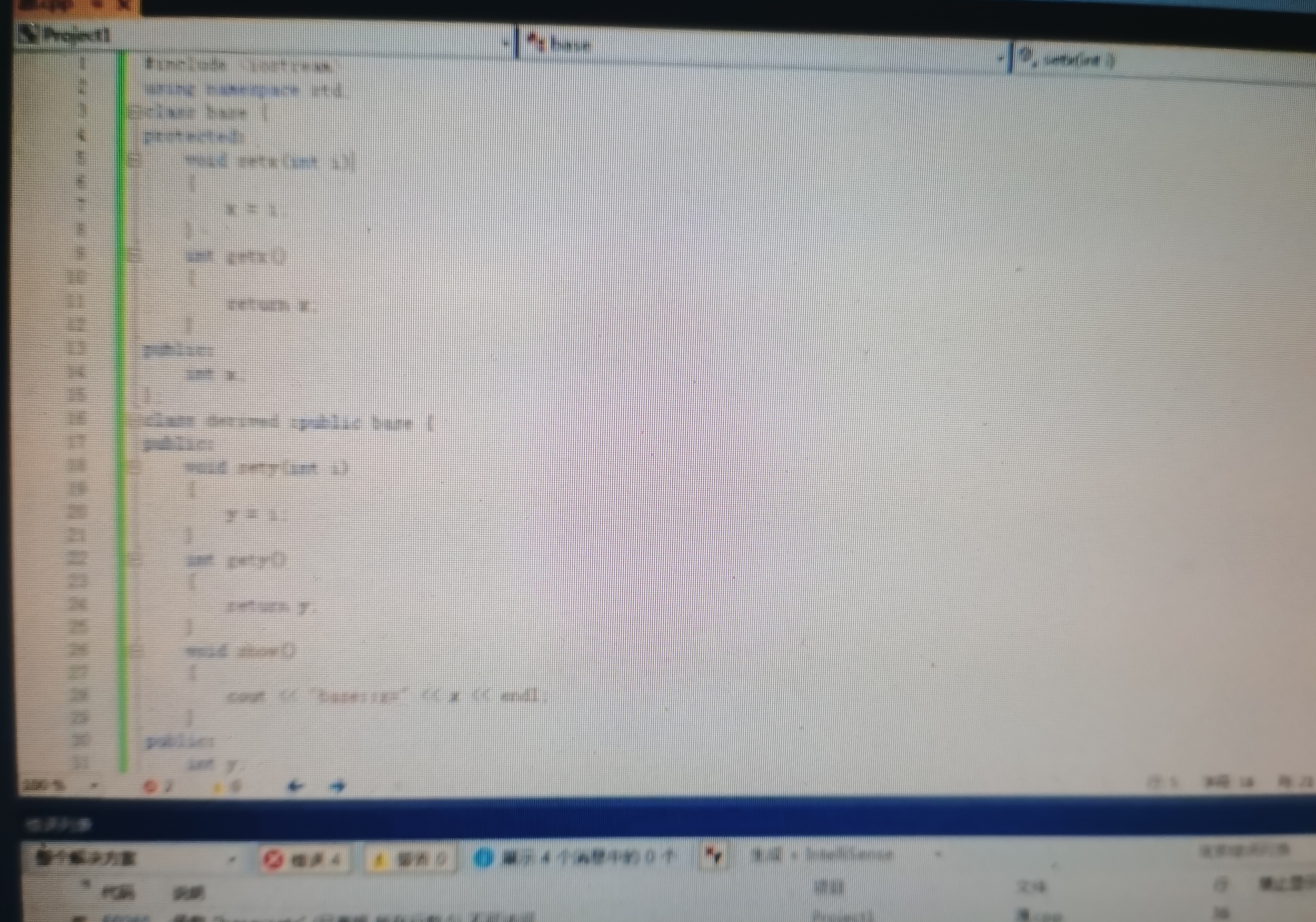
cout<<"Derived::y="<<bb.y<<endl; // 语句7

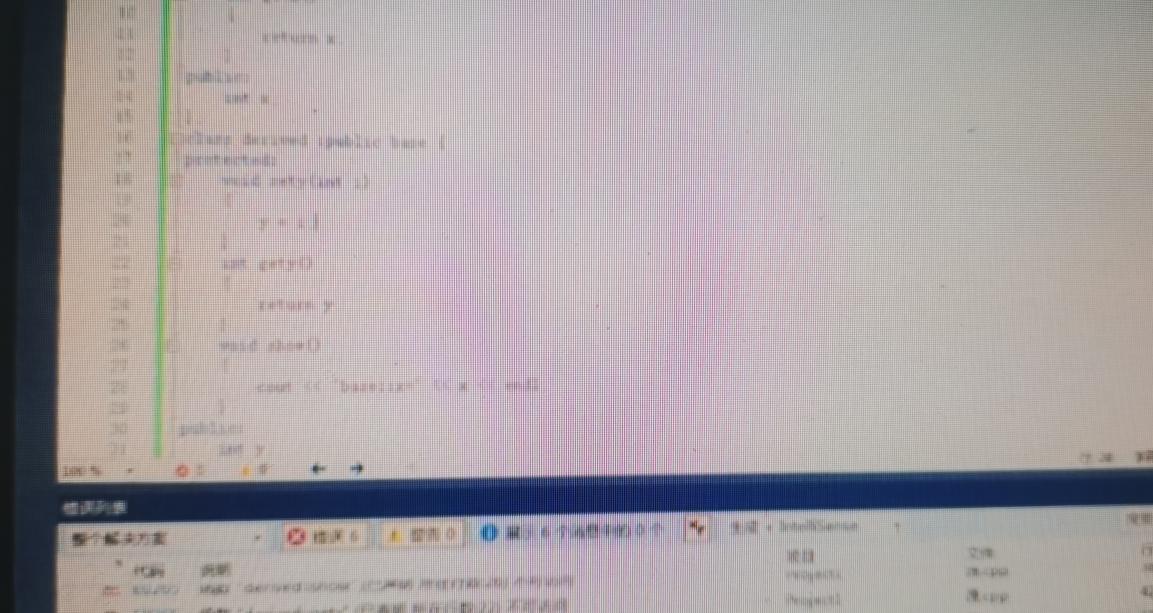
cout<<"Base::x="<<bb.getx()<<endl; // 语句8

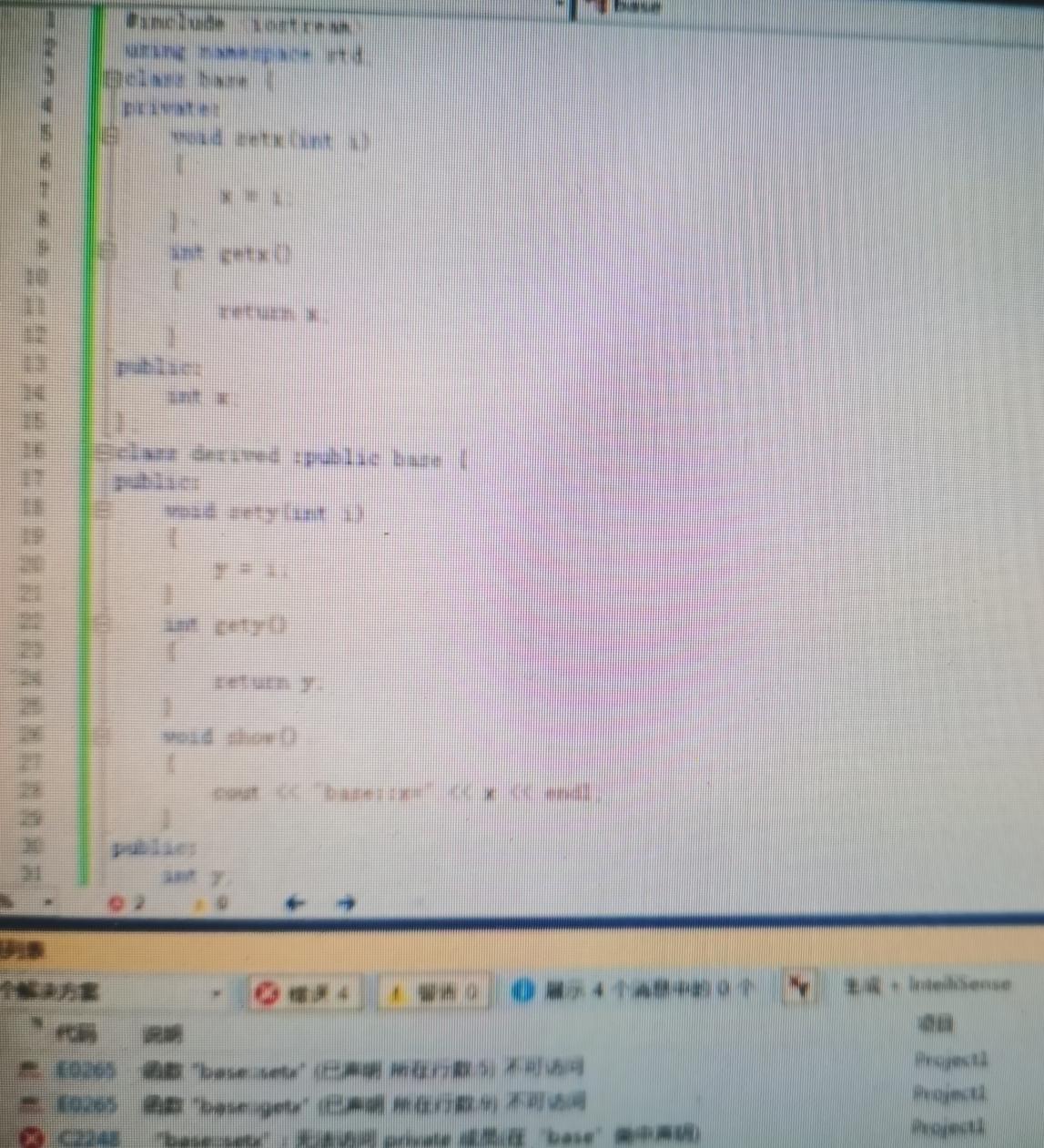
cout<<"Derived::y="<<bb.gety()<<endl; // 语句9

return 0;

}

****

****

****

**3.心得体会**

通过学习本节继承，学会了每种继承方式后的访问性，不同类型的变量经过不同的继承后其访问性质也会发生改变。

Copyright©2021-2099 ShiRuiFu.All rights reserved.

