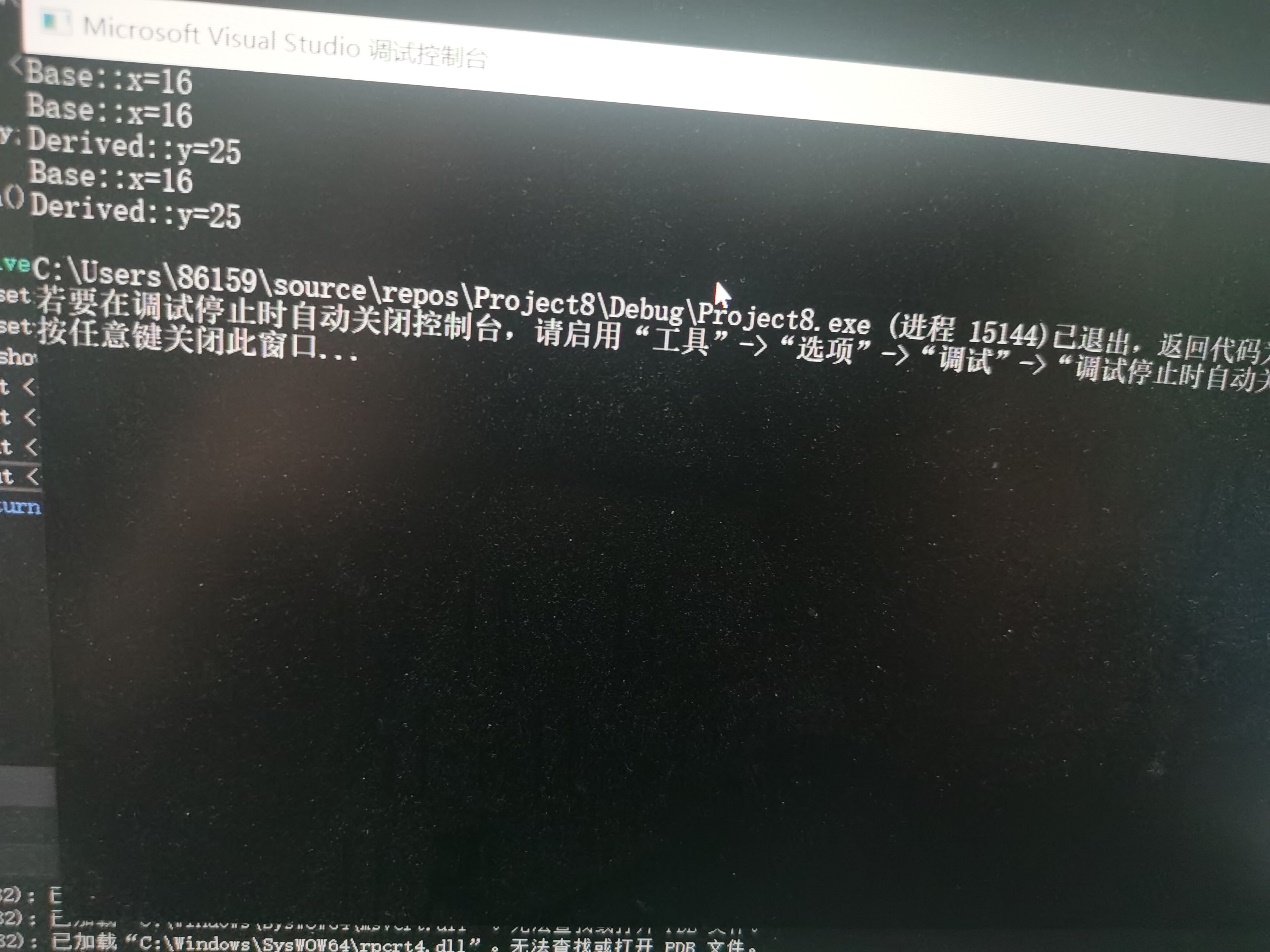
第五次实验报告：派生类与继承

1. 程序代码
2. #include<iostream>
3. using namespace std;
4. class Base{
5. public:
6. void setx(int i)
7. { x=i; }
8. int getx()
9. { return x; }
10. public:
11. int x;
12. };
13. class Derived:public Base{
14. public:
15. void sety(int i)
16. { y=i; }
17. int gety()
18. { return y; }
19. void show()
20. { cout<<"Base::x="<<x<<endl; // 语句1 }
21. public:
22. int y;
23. };
24. int main()
25. {
26. Derived bb; // 语句2
27. bb.setx(16); // 语句3
28. bb.sety(25); // 语句4
29. bb.show(); // 语句5
30. cout<<"Base::x="<<bb.x<<endl; // 语句6
31. cout<<"Derived::y="<<bb.y<<endl; // 语句7
32. cout<<"Base::x="<<bb.getx()<<endl; // 语句8
33. cout<<"Derived::y="<<bb.gety()<<endl; // 语句9
34. return 0;
35. }

2.程序结果



1：基类中的x的访问权改为private时，main函数以及派生类中的x都不能访问，因为基类中的访问权为private。

2：将基类中x的访问权改为protected时，main函数中的x无法访问，但派生类中的可以。

因为x的访问权为protected。

3：将派生类的继承方式改为private时，main函数中的x与setx函数不能访问。因为派生类的继承方式为private。

4：将派生类的继承方式改为protected时，main函数中的x不能访问，因为派生类的继承方式为protected。

3.心得体会

本章的学习最主要的就是学会基类与派生类的继承。要明白当基类中的成员以及成员函数为private、protected、public时对应的派生类的访问权，更要明白当派生类的继承方式为private、protected、public时不同的访问，对于我而言，最容易混淆的就是protected的继承方式时对应的派生类的访问权限，这是我的一大弱点。通过本次实验报告，我很好的明白了梳理了protected继承时的访问权限，解决了自己的一大缺点。