

#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

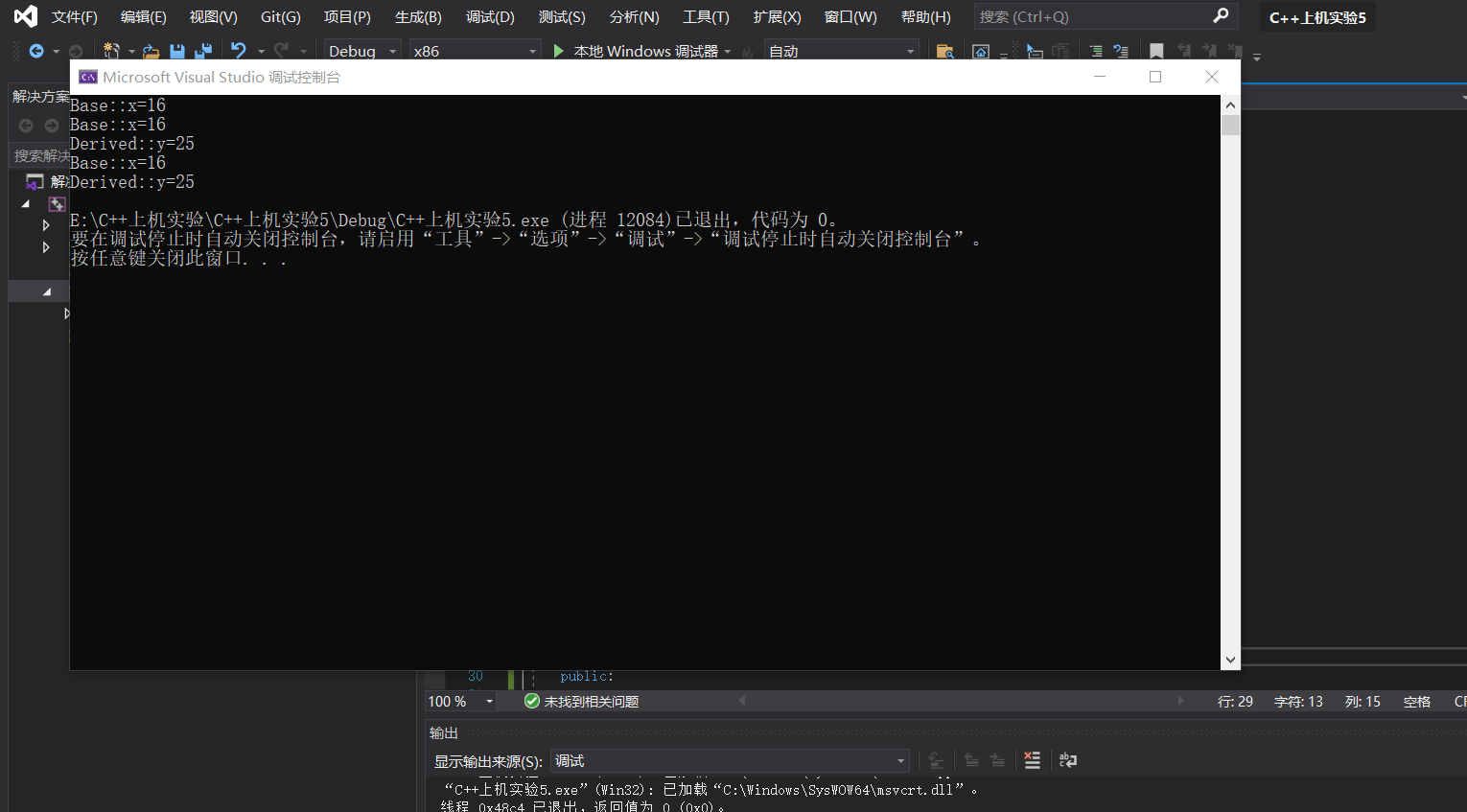
cout << "Derived::y=" << bb.gety() << endl; // 语句9

return 0;

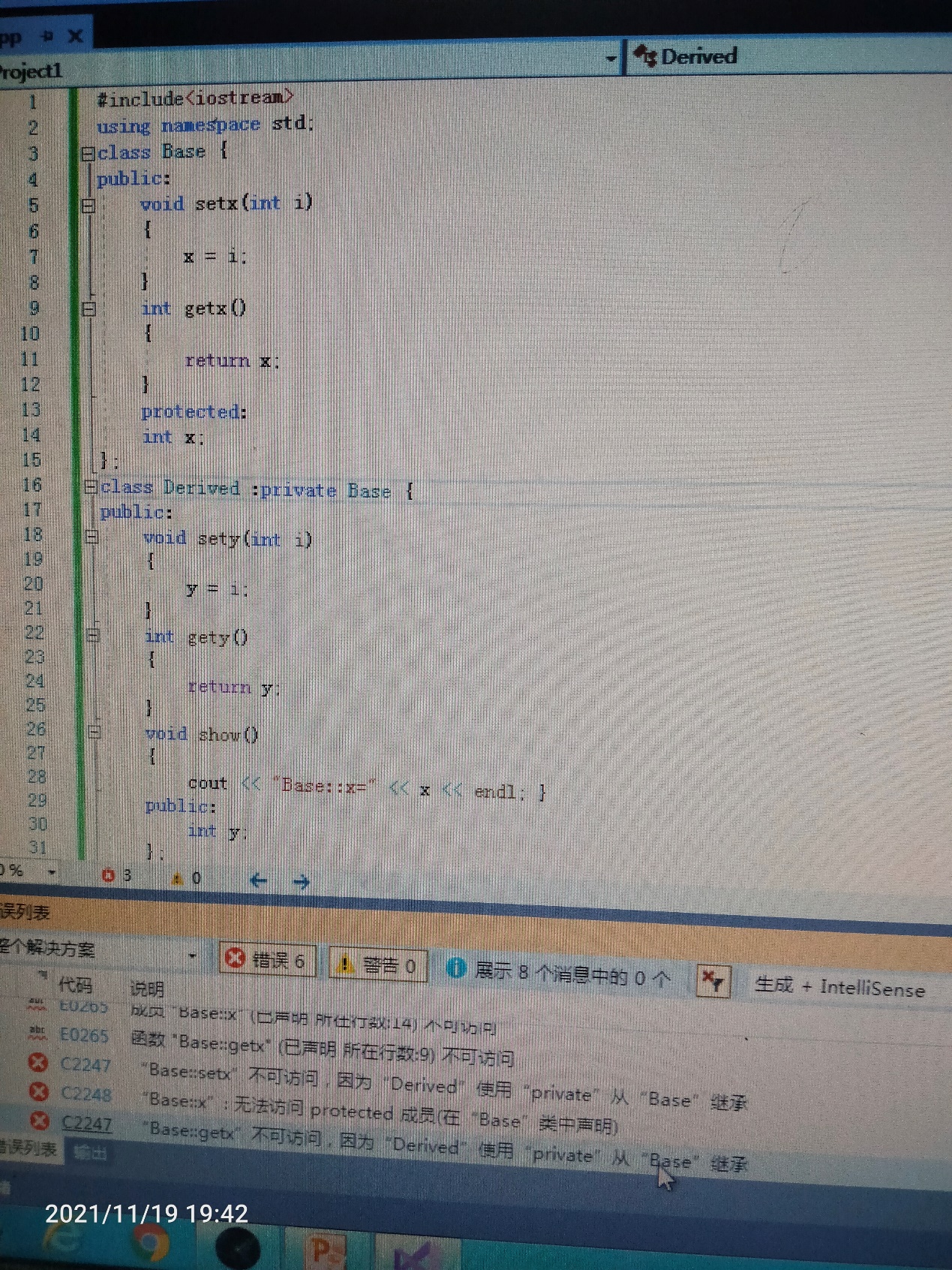
}

**实验内容**

1. **写出程序运行的结果。**

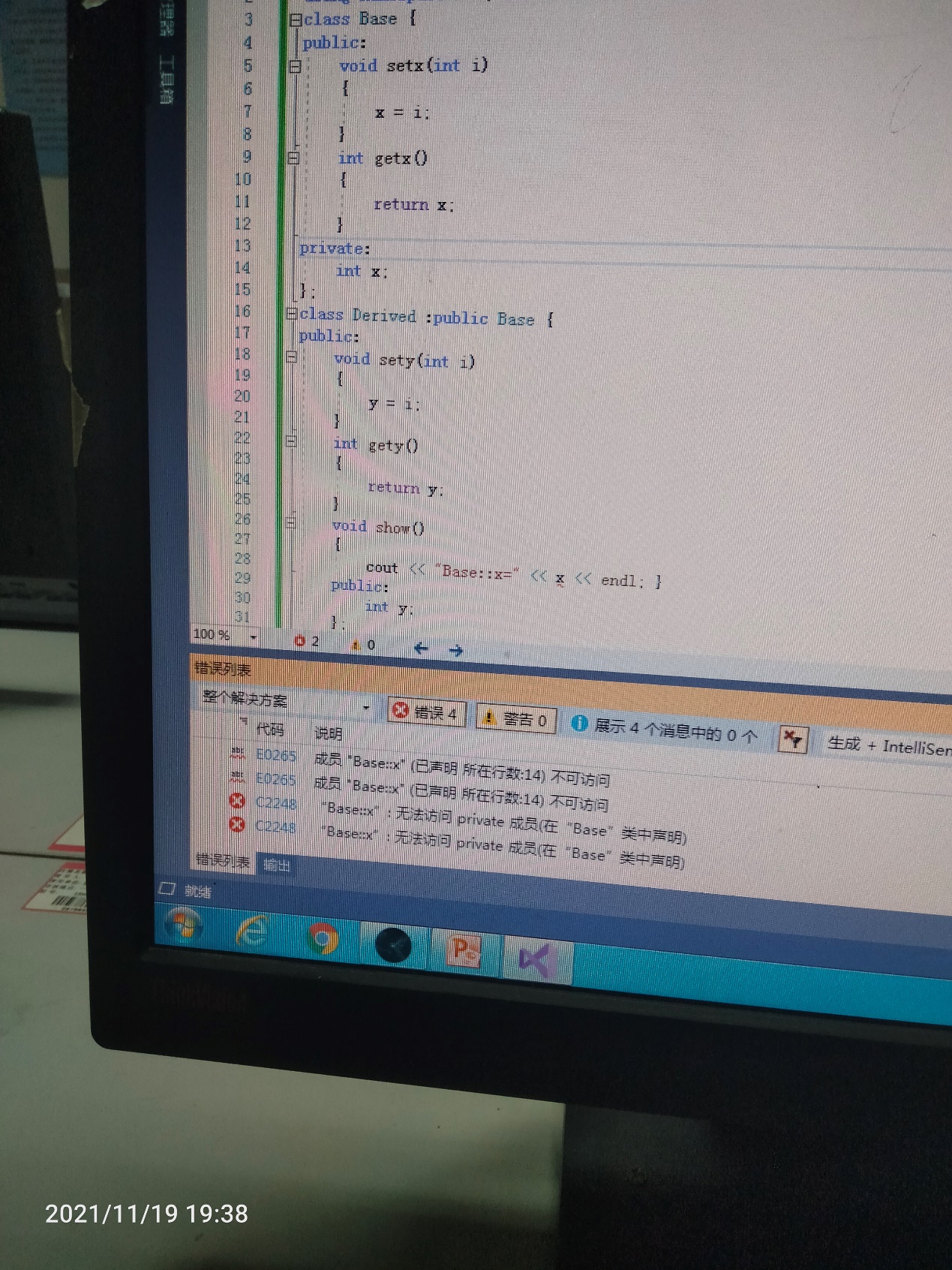


1. 按照以下要求，对程序进行修改后再调试，指出调试中出错的原因。
2. 将基类Base中数据成员x的访问权限改为private时，会出现哪些错误，为什么？



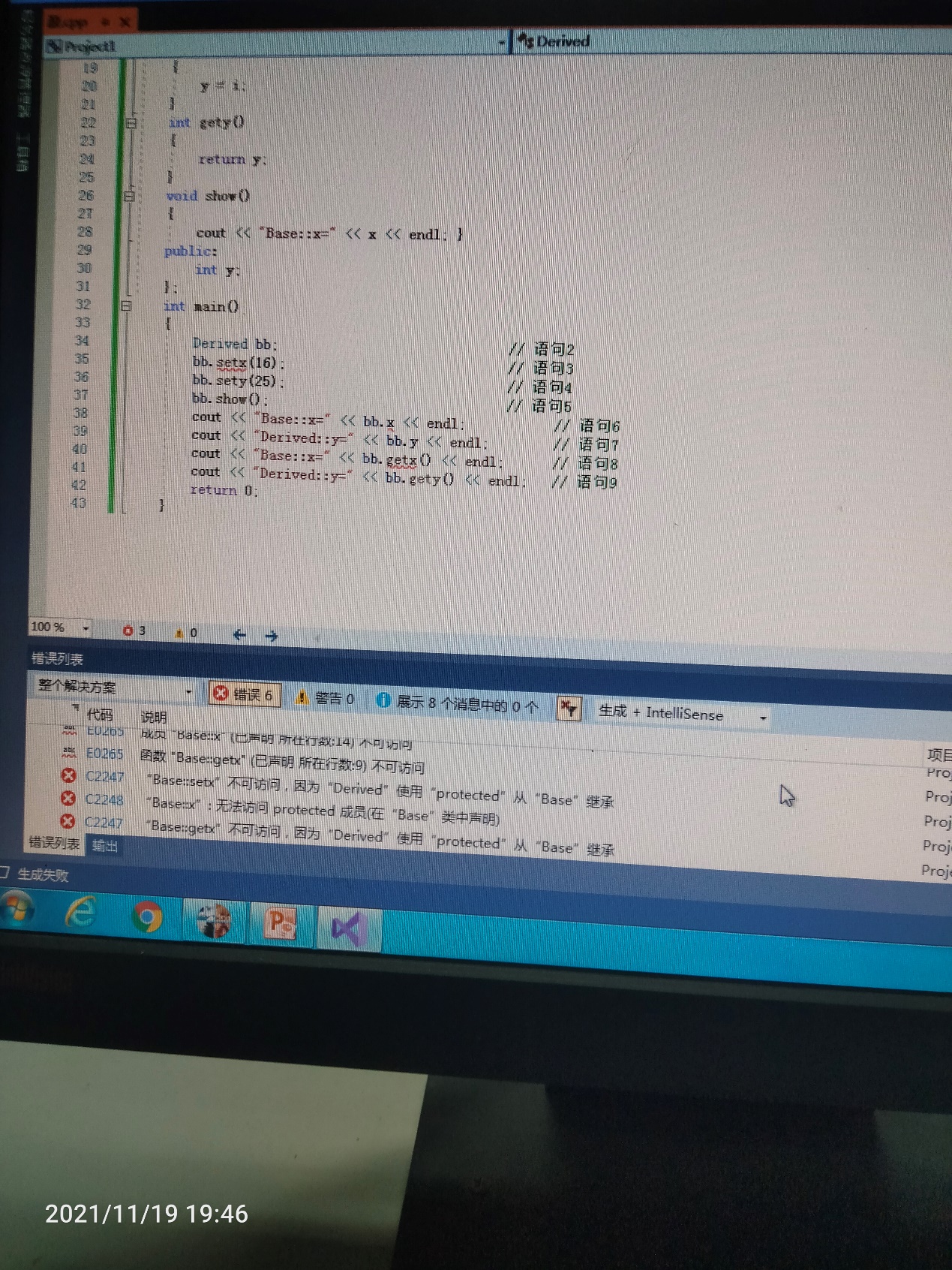
会导致派生类和对象均无法访问基类成员x，进而出现错误。

（2）.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？



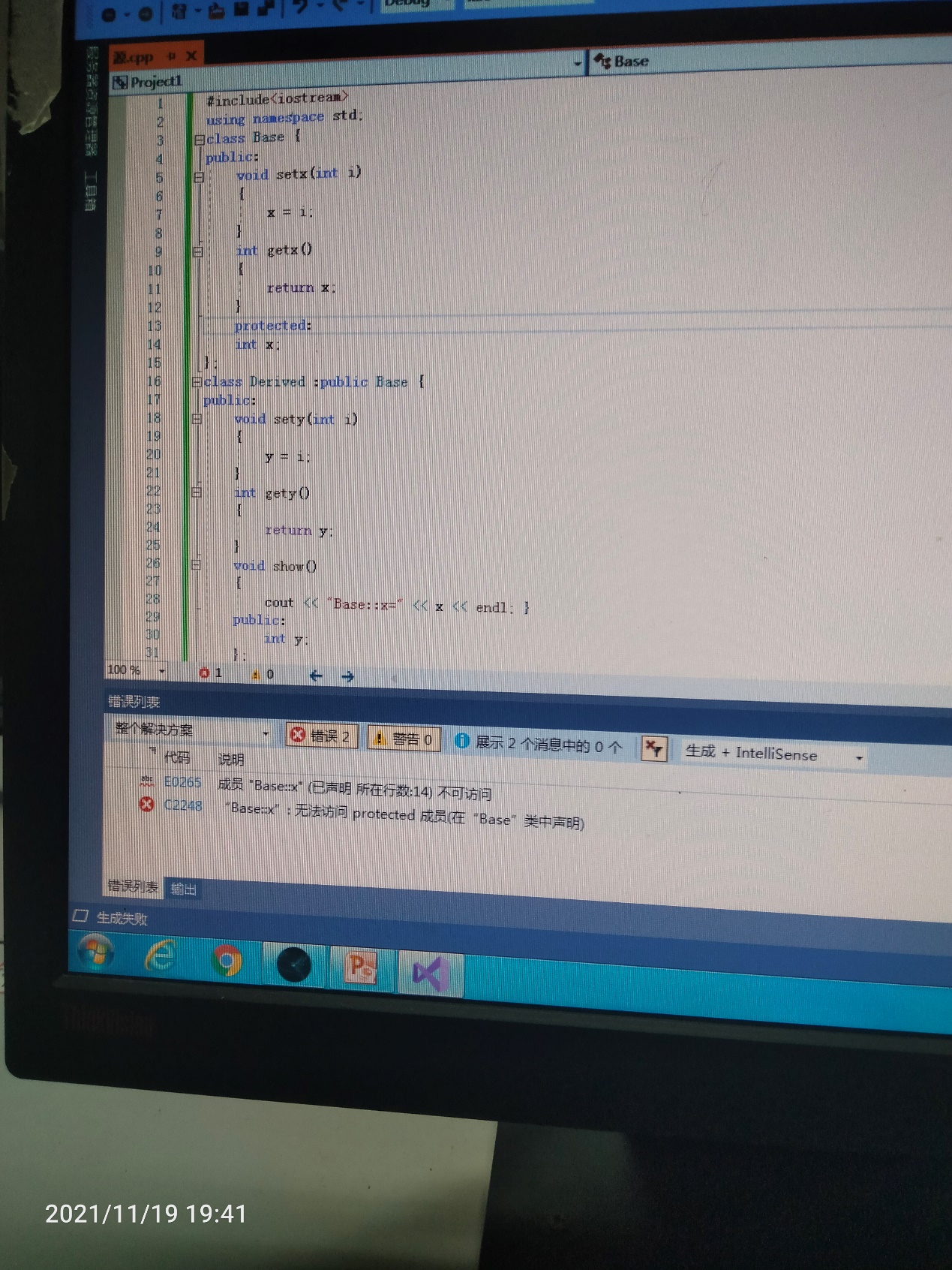
会导致对象无法访问基类成员x。

（3）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？



会导致派生类和对象均无法访问基类的成员函数和成员变量。

（4）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？



会导致对象无法访问基类的成员函数及变量。