实验内容:

#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

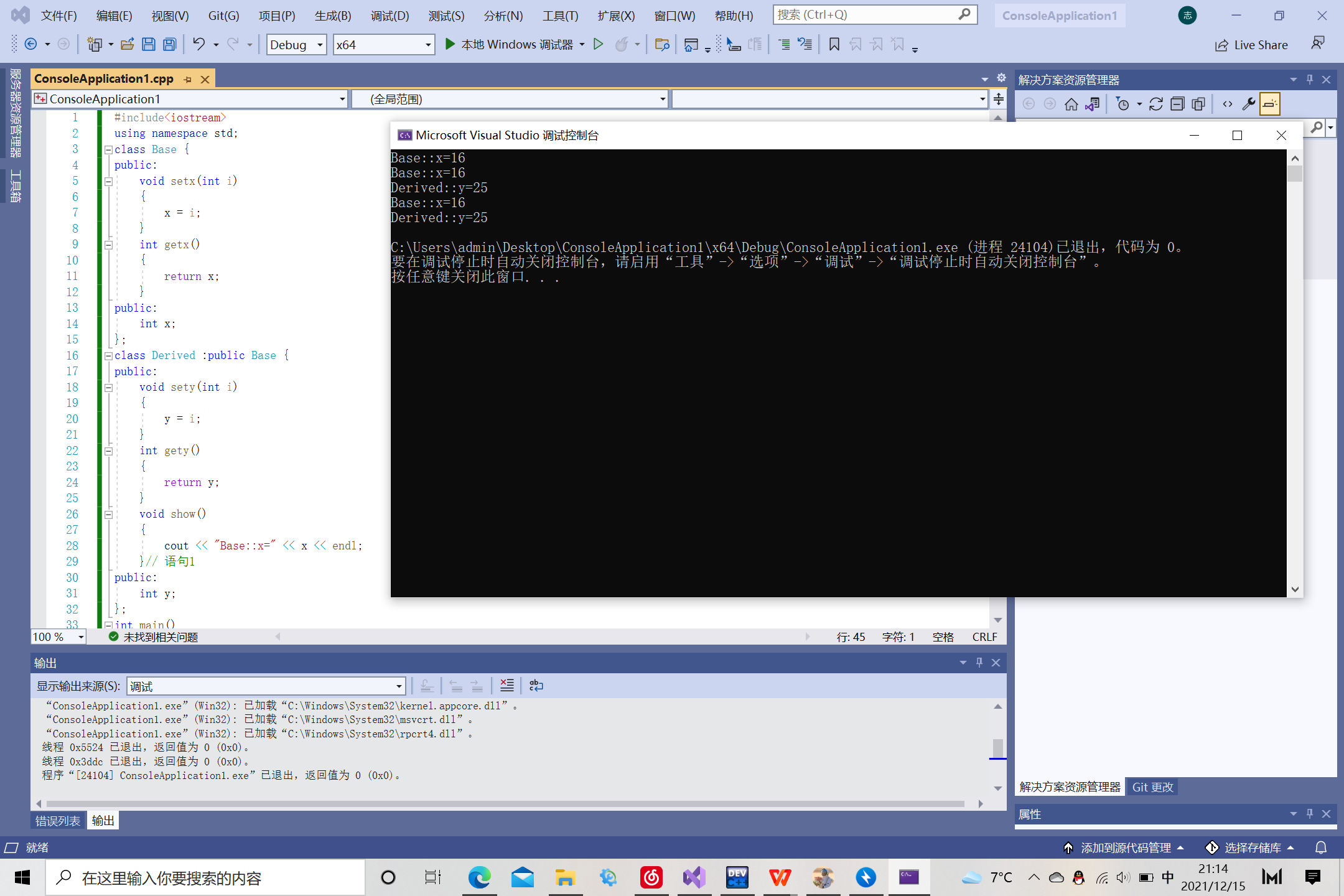
cout << "Base::x=" << bb.getx() << endl; // 语句8

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

return 0;

}

运行结果：



开始能顺利输出结果，每次修改程序后，均会出现无法访问的情况，可见修改后降低了权限。权限感想心得：

一般地，在类的内部，无论成员被声明为哪种，都是可以互相访问的；但在类的外部，如通过类的对象，则只能访问 public 属性的成员，不能访问protected、private属性的成员掌握了派生类的声明方法和构造函数的定义方法等。

Copyright ©2021-2099 xiaoxuandong. All rights reserved