代码：

#include<iostream>

#include<cmath>

#include<string>

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(17); // 语句3

bb.sety(28); // 语句4

bb.show(); // 语句5

cout << "Base::x=" << bb.x << endl; // 语句6

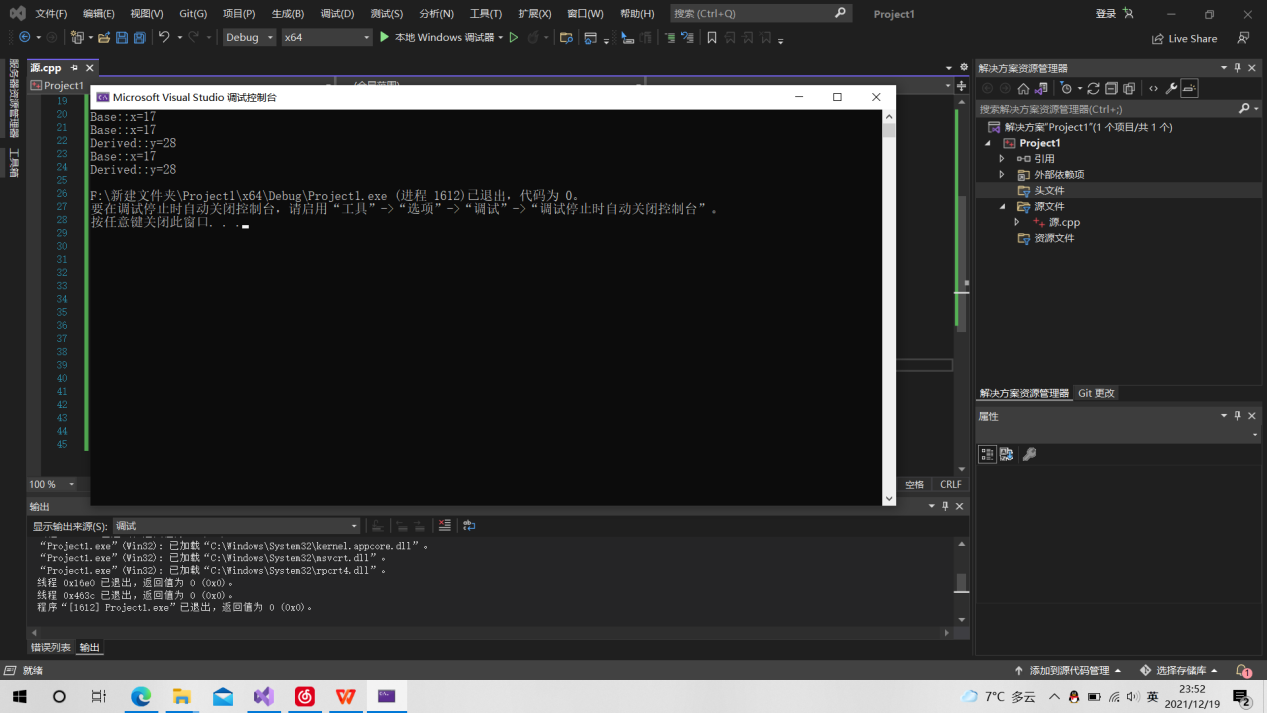
cout << "Derived::y=" << bb.y << endl; // 语句7

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

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

}

结果：



总结：掌握派生类的声明方法和派生类构造函数的定义方法，掌握不同方式下，基类成员在派生类中的访问属性，分别通过public，protected，private三个关键字来控制成员变量和成员函数的访问权限，分别表示：公有的、受保护的、私有的。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。