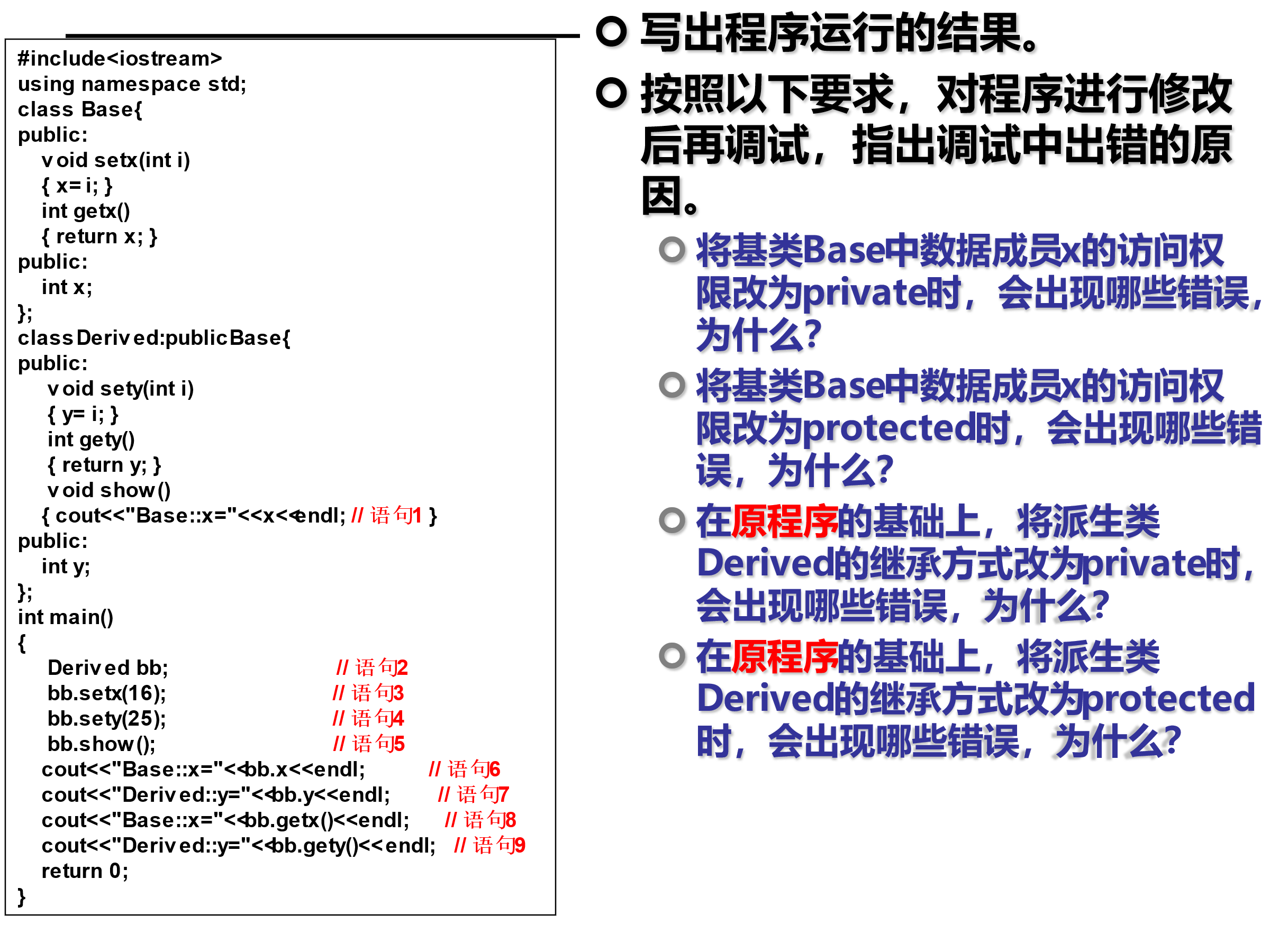
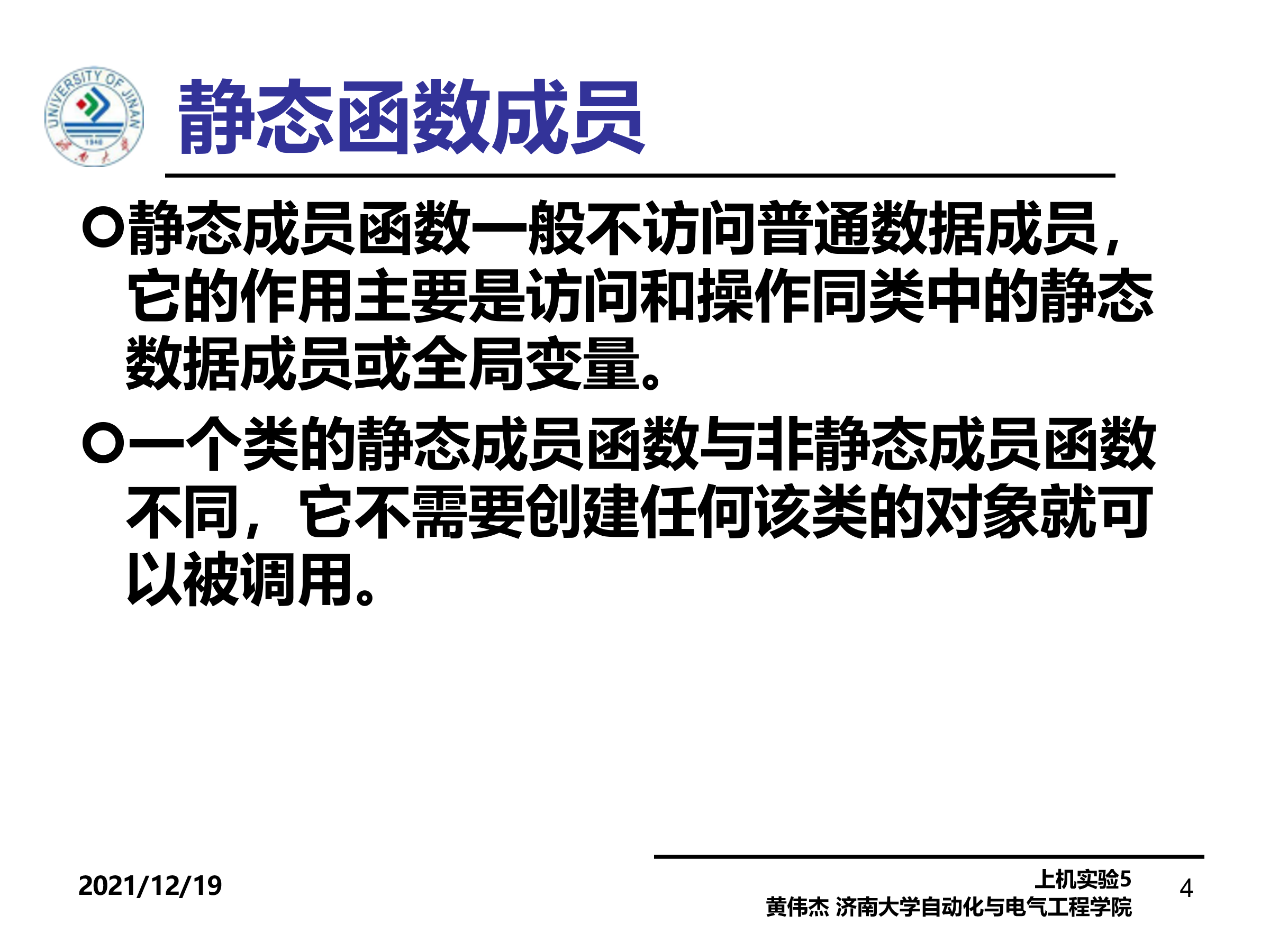
**实验内容：**



**程序：**

#include <iostream>

using namespace std;

class Base {

public:

void setx(int i)

{

x = i;

}

int getx()

{

return x;

}

private:

int x;

};

class Derived :public Base {

public:

void sety(int i)

{

y = i;

}

int gety()

{

return y;

}

void show()

{

cout << "Base::x=" << x << endl;

}

public:

int y;

};

int main()

{

Derived bb;

bb.setx(16);

bb.sety(25);

bb.show();

cout << "Base::x=" << bb.x << endl;

cout << "Derived::y=" << bb.y << endl;

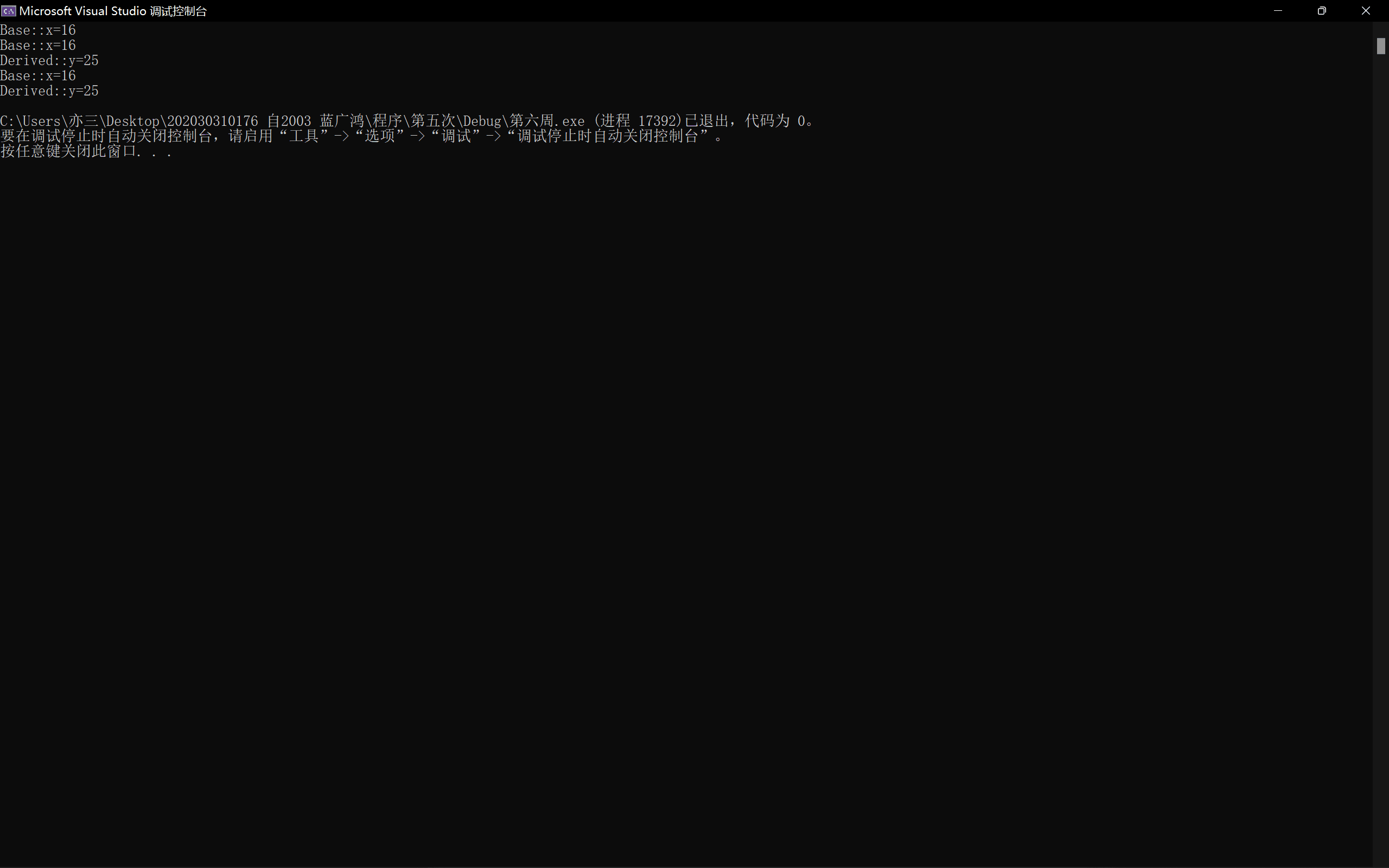
cout << "Base::x=" << bb.getx() << endl;

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

return 0;

}

**运行：**

****

**1、“base：：x”无法访问private成员**

**2、“base：：x”无法访问protected成员**

**3、出现错误为：bb.x , bb.getx()无法访问**

**4、出现错误为：bb.x , bb.getx()无法访问**

**总结：**

**在一开始只是按照课件打出程序，并按照程序进行一系列的操作，课后根据实验要求,并查阅资料。得出了一些结论：**

**声明一个派生类的一般格式：class 派生类名[继承方式] 基类名{派生类新增的数据成员和成员函数}；**

**派生类构造函数的一般格式：派生类名（参数总表）：基类名（参数表）{**

**派生类新增数据成员的初始化语句}**