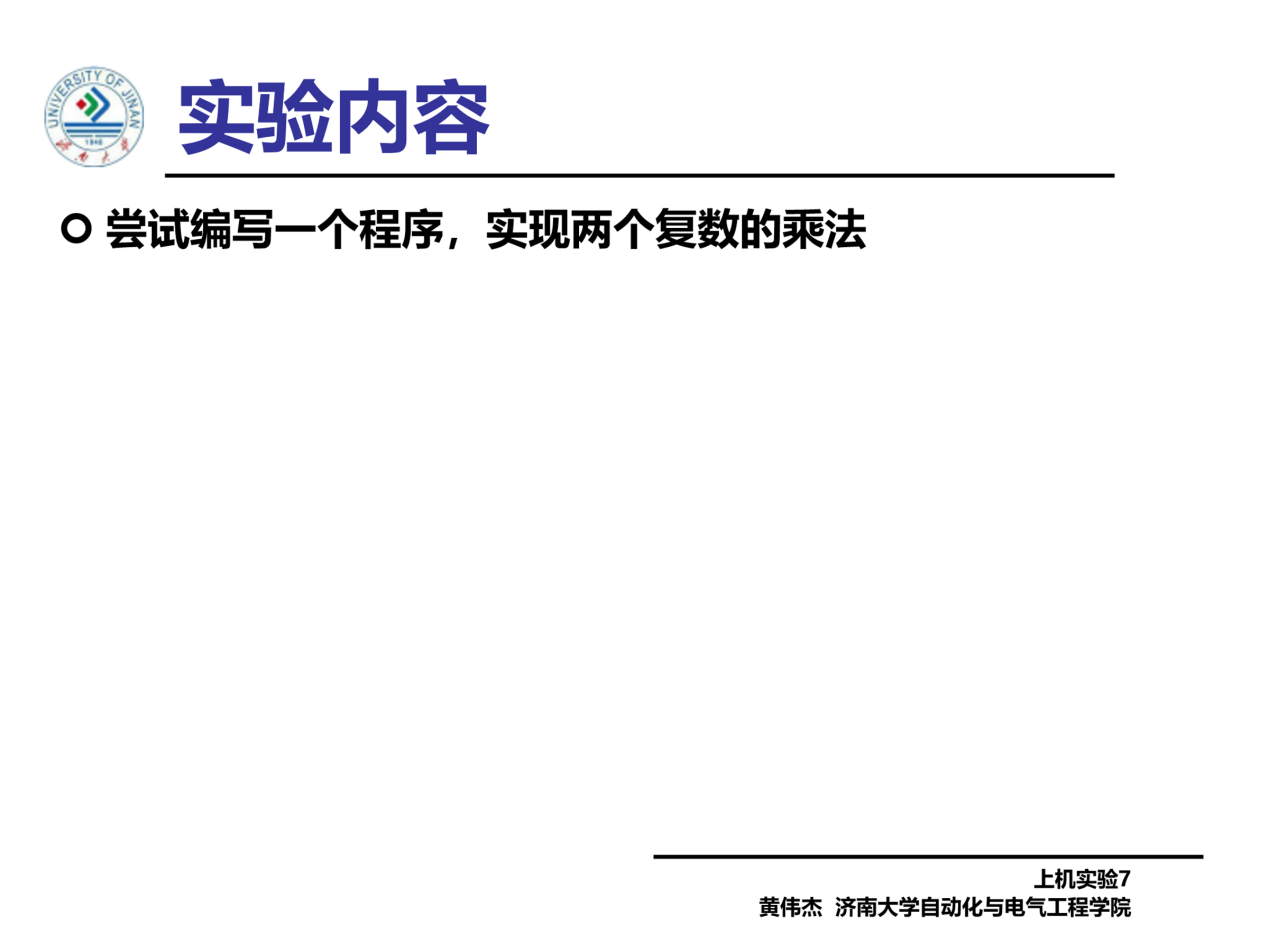
Copyright ©2021-2099 LuMingkai. All rights reserved

**实验要求：**



**程序：**

**#include <iostream>**

**using namespace std;**

**class Complex {**

**public:**

**double real;**

**double imag;**

**Complex(double r = 0, double i = 0)**

**{**

**real = r;**

**imag = i;**

**}**

**};**

**Complex operator \*(Complex co1, Complex co2) {**

**Complex temp;**

**temp.real = co1.real \* co2.real - co1.imag \* co2.imag;**

**temp.imag = co1.real \* co2.imag - co1.imag \* co2.real;**

**return temp;**

**}**

**int main()**

**{**

**Complex com1(1.1, 2.2), com2(3.3, 4.4), total1, total2;**

**total1 = operator \*(com1, com2);**

**cout << "real=" << total1.real << " " << "imag1=" << total1.imag << endl;**

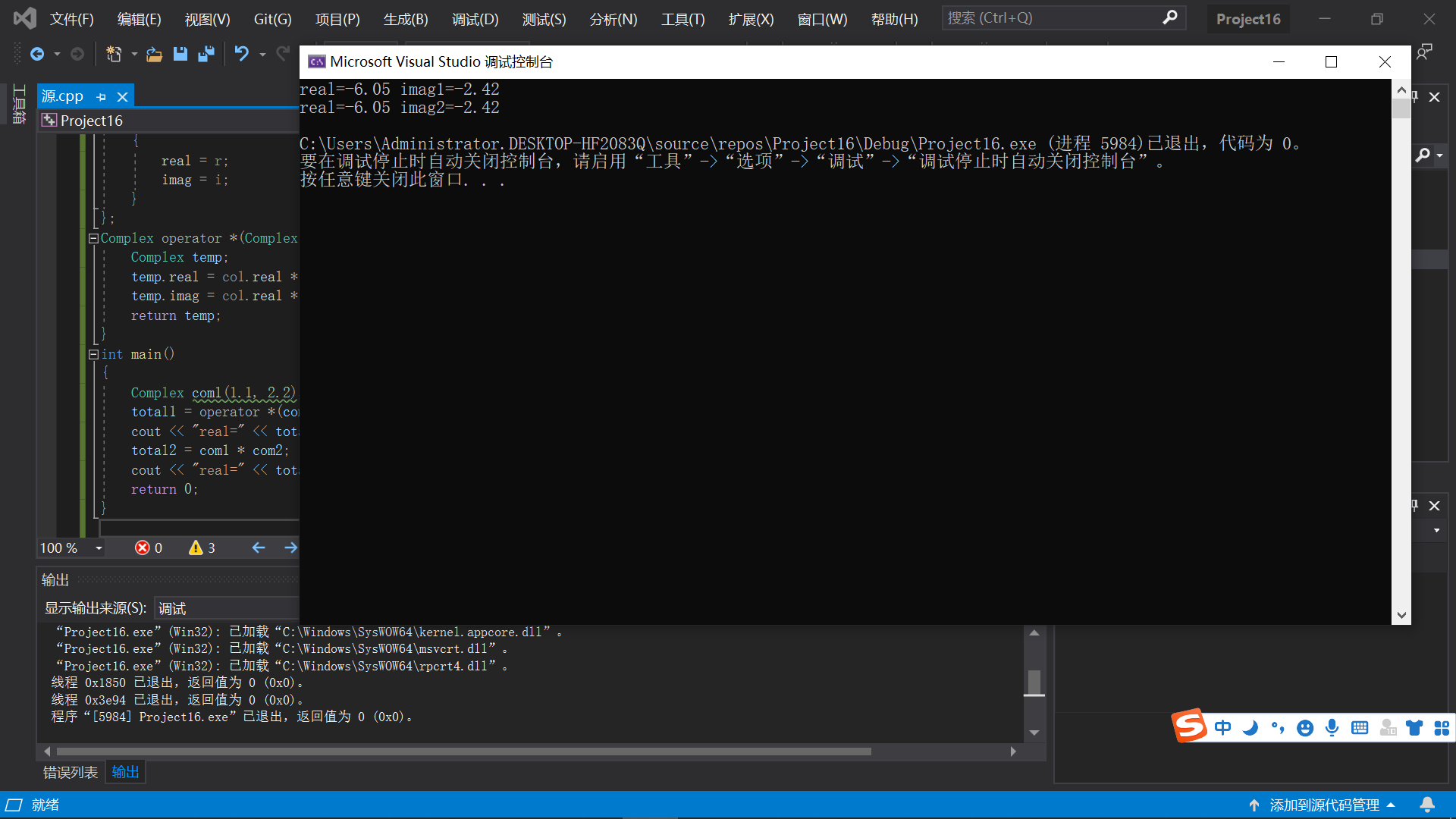
**total2 = com1 \* com2;**

**cout << "real=" << total2.real << " " << "imag2=" << total2.imag << endl;**

**return 0;**

**}**

**实验程序输出结果：**



**实验总结：**

多次编译均出现错误：输出复数相乘结果不对。查阅资料后知道有两处错误，其一复数乘法格式写错了，其二输出结果时把实部与虚部写反了。

**认识到了运算符重载的好处：**

那些运算符本身功能是有限的，比如不能进行复数的加减，但是我们重载以后，就可以让它实数与实数相加，虚数与虚数相加，最后结果也是复数
还有字符串string里也有很多重载的地方，像两个字符串也可以相加，等等。