**运算符重载**

**一、实验程序**

#include<iostream>

using namespace std;

class Complex {

double real;

double imag;

public:

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

{

real = r; imag = i;

}

void print();

friend Complex operator\*(Complex co1, Complex co2);

};

Complex operator\*(Complex co1, Complex co2)

{

Complex temp;

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

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

return temp;

}

void Complex::print()

{

cout << "total real=" << real << " " << " total imag=" << imag << endl;

}

int main()

{

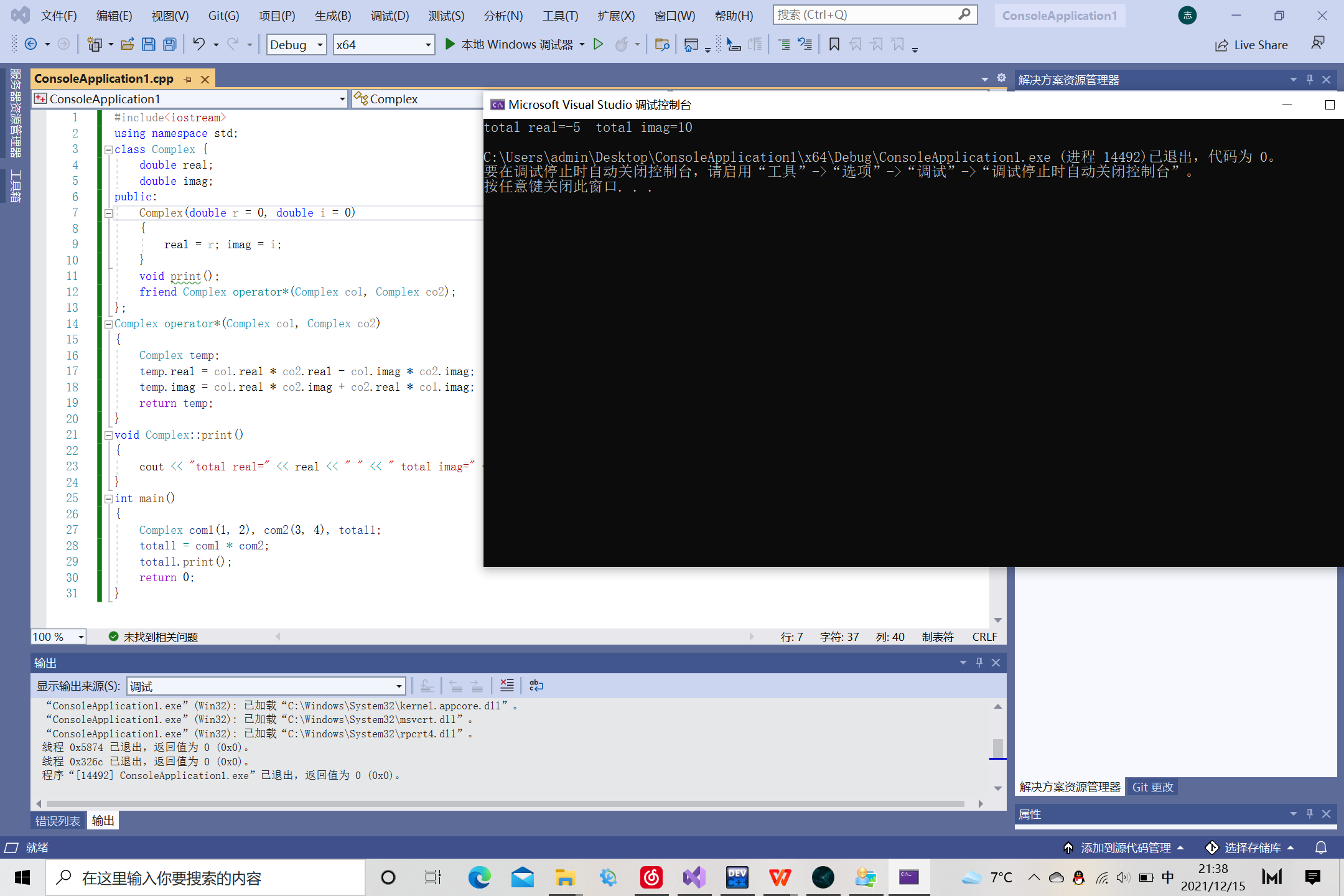
Complex com1(1, 2), com2(3, 4), total1;

total1 = com1 \* com2;

total1.print();

return 0;}

**二、实验结果**



**三、心得体会**

复数的乘法相比于加法而言仅仅是算法有所改变，但我们更应该进行实践，由此对运算符重载有更深入的理解。对双目运算符而言，成员运算符重载函数的形参表中仅有一个参数，它作为运算符的右操作数。另一个操作数是隐含的，是该类的当前对象，他是通过this指针隐含传递给函数。

Copyright ©2021-2099 电自2003班 陈阳202030310072. All rights reserved