## 上机七

**实验一**

#include<iostream>

using namespace std;

class Complex {

public:

double real, img;

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

{

real = r;

img = i;

}

};

Complex operator\*(Complex a, Complex b)

{

Complex temp;

temp.real = a.real \* b.real - a.img \* b.img;

temp.img = a.real \* b.img + a.img \* b.real;

return temp;

}

int main()

{

double a, b, c, d;

cout << "请输入两个复数的实部和虚部" << endl;

cin >> a >> b >> c >> d;

Complex A(a, b);

Complex B(c, d);

Complex temp;

temp = A \* B;

cout << "乘积的实部是：" << temp.real << endl;

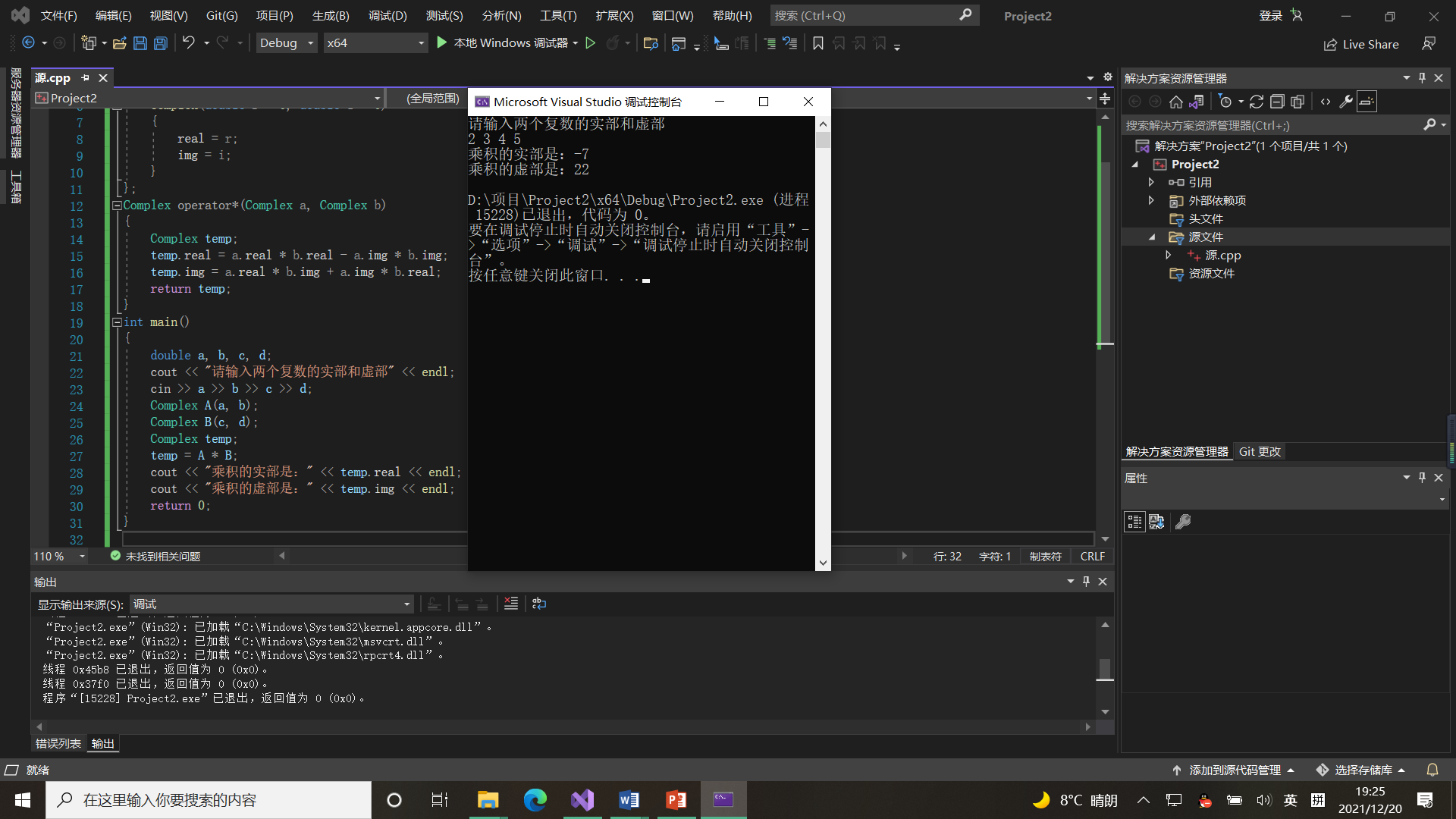
cout << "乘积的虚部是：" << temp.img << endl;

return 0;

}

**运行结果**

**实验一**



**感想&心得：**

运算符重载，就是对已有的运算符重新进行定义，赋予其另一种功能，以适应不同的数据类型。

运算符重载是通过创建运算符重载函数来实现的。运算符重载体现了C++的多态性。运算符重载是针对新类型数据的实际需要，对原有运算符进行适当的改造完成的。重载不能改变运算符的操作对象的个数。

运算符重载的函数格式：

函数类型 operator 运算符名称（形参表）

{

对运算符的重载处理

}

Copyright ©2021-2099 Shuyue. All rights reserved