第七次上机：运算符重载

#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;

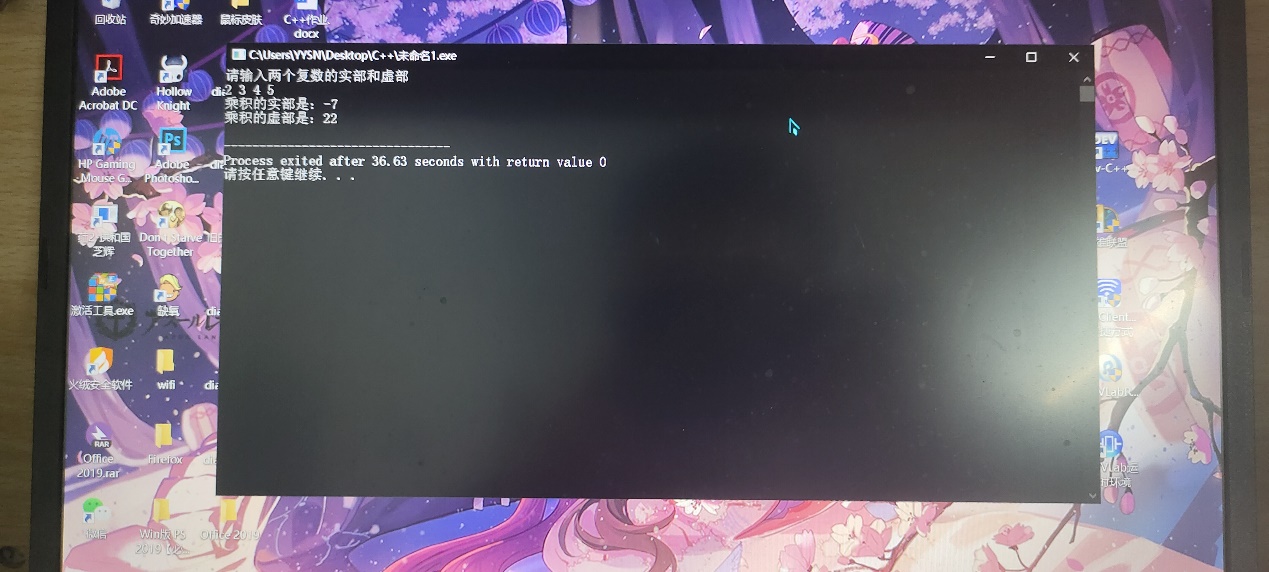
temp = A \* B;

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

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

return 0;

}



在C++中，除了可以对函数重载外,还可以对大多数运算符实施重载。运算符重载与函数重载相比,稍微复杂一些。运算符重载是通过创建运算符重载函数来实现的。运算符重载函数可以是在类外定义的普通函数,也可以是类的成员函数或友元函数。