第七次上机实验报告

1. **实验目的**

掌握C++语言多态性的基本概念

掌握运算符重载函数的声明和定义方法

1. **实验内容**

尝试编写一个程序，实现两个复数的乘法

**三、实验程序结果**

#include<iostream>

using namespace std;

class Complex {

double real;

double imag;

public:

Complex(double r = 0.0, double i = 0.0);

void print();

Complex operator+(Complex c);

};

Complex::Complex(double r, double i)

{

real = r; imag = i;

}

Complex Complex::operator+(Complex c)

{

Complex temp;

temp.real = real\*c.real-imag\*c.imag;

temp.imag = real\*c.imag+c.real\*imag;

return temp;

}

void Complex::print()

{

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

}

int main()

{

double x1, x2, y1, y2;

cin >> x1 >> y1 >> x2 >> y2;

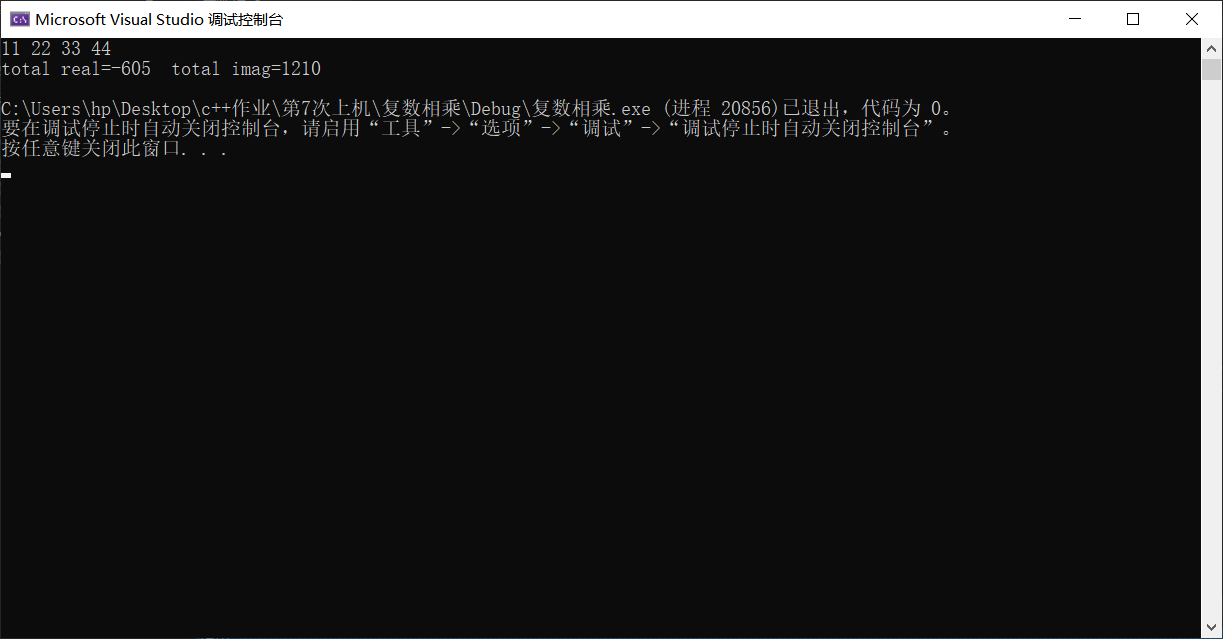
Complex com1(x1, y1), com2(x2, y2), total1;

total1 = com1 + com2;

total1.print();

return 0;

}



**四、实验心得**

C++可以对一些已有的运算符进行“重载”操作，重新改造使之进行之前所不能完成的运算

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（左操作数）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。