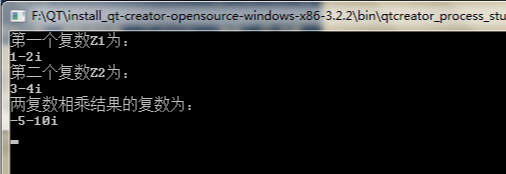
实验内容：#include <iostream> using namespace std; class matrixmul{private: double real; double imag;public: void output();//打印输出 void initSet(double re,double im);//初始化 matrixmul matrixMul(matrixmul Z1,matrixmul Z2);//函数返回值是multirxmul，所以此函数类型为matrixmal}; matrixmul matrixmul::matrixMul(matrixmul Z1,matrixmul Z2){ double temp1,temp2,temp3; matrixmul result; temp1=Z1.real\*Z2.imag; temp2=Z1.imag\*Z2.real; temp3=(Z1.imag+Z1.real)\*(Z2.real-Z2.imag); result.real=temp1+temp3-temp2; result.imag=temp1+temp2; return result;} void matrixmul::initSet(double re,double im){ real=re; imag=im;} void matrixmul::output(){ if(imag>0) { cout<<real<<"+"<<imag<<"i"<<endl; } else if(imag==0) { cout<<real<<endl; } else if(imag<0) { cout<<real<<imag<<"i"<<endl; }} int main(){ matrixmul Z1,Z2,Z3,result; Z1.initSet(1,-2); Z2.initSet(3,-4); cout<<"第一个复数Z1为："<<endl; Z1.output(); cout<<"第二个复数Z2为："<<endl; Z2.output(); result=Z3.matrixMul(Z1,Z2); cout<<"两复数相乘结果的复数为："<<endl; result.output(); return 0;}运行结果：



感想

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。然后再尝试编写一个程序，实现两个复数的乘法。

经过这几次的上机，我对C++有着更加深刻的理解。