第七次实验报告：运算符重载

1.程序代码

#include<iostream>

using namespace std;

class fs

{

private:

double real;

double vir;

public:

fs();

friend fs operator\*(fs a1, fs a2);

void input();//输入函数

void display();//输出函数

};

fs::fs()

{

real = 0;

vir = 0;

}

fs operator\*(fs a1, fs a2)

{

fs a3;

a3.real = a1.real\*a2.real - a1.vir\*a2.vir;

a3.vir = a1.vir\*a2.real + a1.real\*a2.vir;

return a3;

}

void fs::input()//输入函数

{

cin >> real >> vir;

}

void fs::display()//输出函数

{

if (real != 0)

{

if (vir > 0)cout << real << "+" << vir << "i" << endl;

if (vir == 0)cout << real << endl;

if (vir < 0)cout << real << vir << "i" << endl;

}

if (real == 0)

{

if (vir > 0)cout << vir << "i" << endl;

if (vir == 0)cout << "0" << endl;

if (vir < 0)cout << vir << "i" << endl;

}

}

int main()

{

fs c1, c2, c3, c4, c5;

cout << "please input two complex numbers:" << endl;

c1.input();

c2.input();

cout << "c1 is ";

c1.display();

cout << "c2 is ";

c2.display();

c4.display();

c5 = c1 \* c2;

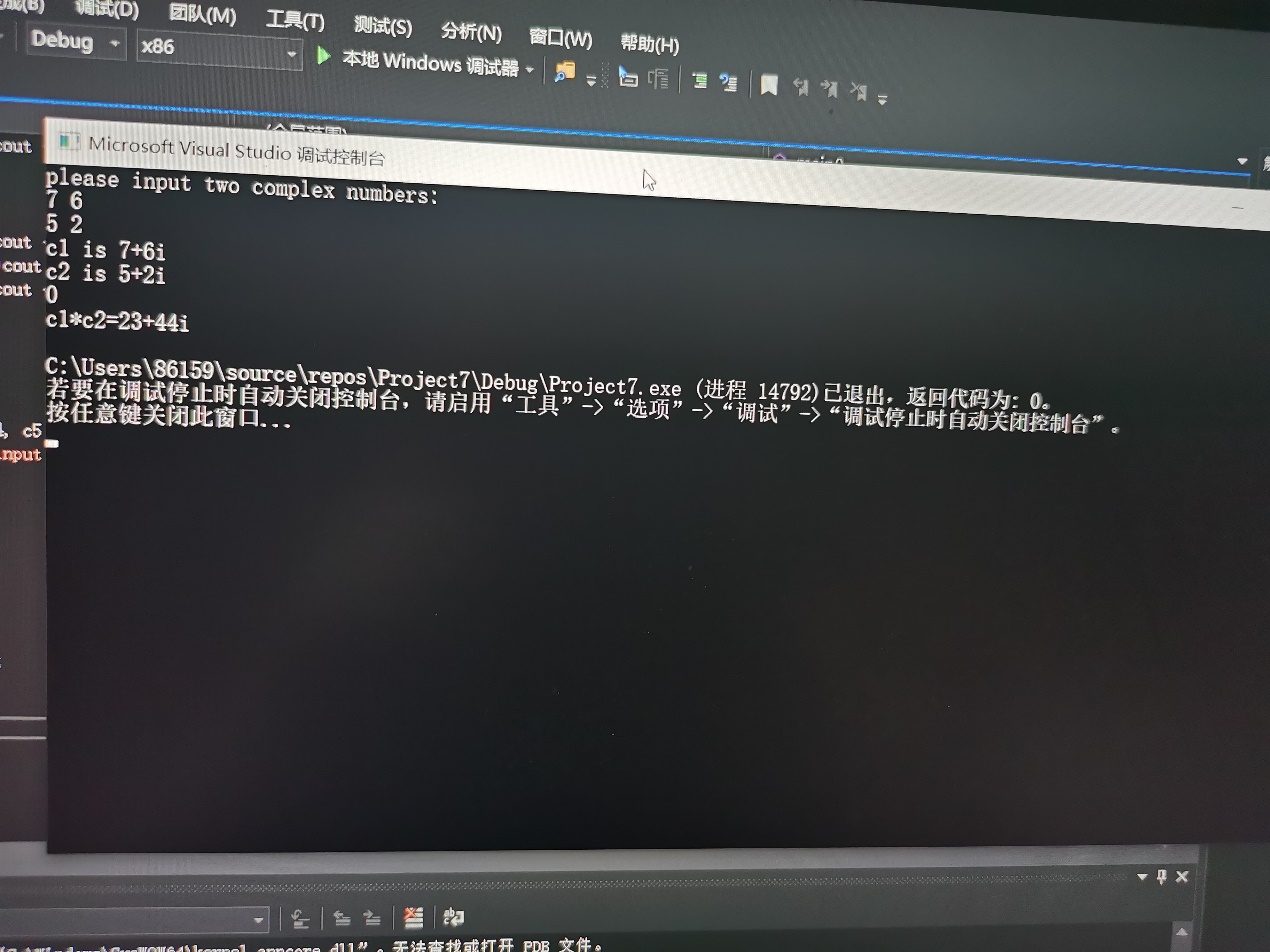
cout << "c1\*c2=";

c5.display();

return 0;

}

2.程序结果



3.心得体会

在运算符重载中，要弄清楚前置和后置运算符的区别，前置运算符是先运算后返回，而后置运算符是先返回后运算，这点十分重要。由于之前没用注意到这一点，所以做出来的值结果不对，后来发现问题改正之后才正确，通过此次实验，掌握了基本的运算符的重载，学会了几种基本的运算符重载方法。