## 程序代码

#include<iostream>

using namespace std;

class Complex {

double real;

double imag;

public:

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

{

real = r; imag = i;

}

void print();

friend Complex operator\*(Complex co1, Complex co2);

};

Complex operator\*(Complex co1, Complex co2)

{

Complex temp;

temp.real = co1.real \* co2.real - co1.imag \* co2.imag;

temp.imag = co1.real \* co2.imag + co1.imag \* co2.real;

return temp;

}

void Complex::print()

{

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

}

int main()

{

float a, b, c, d;

cout << "please input real and imag:" << endl;

cin >> a >> b;

cout << "please input real and imag:" << endl;

cin >> c >> d;

Complex com1(a, b), com2(c, d), total1;

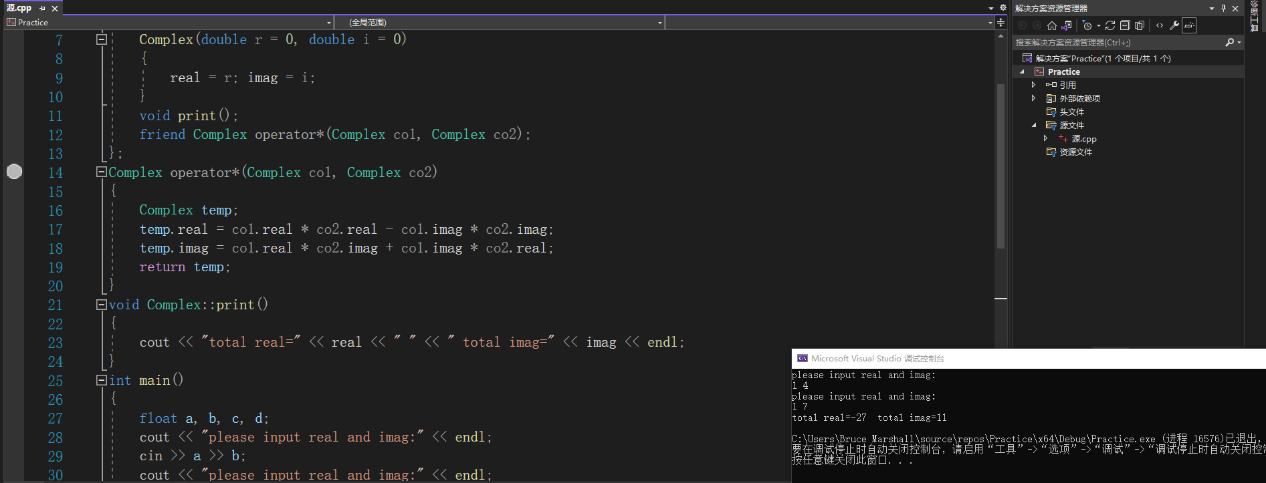
total1 = com1 \* com2;

total1.print();

return 0;

}

## 程序结果



## 感想心得

1、成员函数不能直接调用类的私有成员，而是要通过引用或者指针来调用类的私有成员，友元函数不是类的成员函数，不能用作用域解析运算符操作，友元函数因为不是成员函数，不能被派生类继承

2、通过利用友元函数，在类外进行了运算符的重载，直接使用了重载后的\*；同时将数据的输入与输出放在主函数中，虽然降低了类的重用性，没能很好的体现封装性，但也大大简化了类内的函数，使得程序变得更加简洁易读。

Copyright ©2021-2099 BinLu. All rights reserved