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：

输入：

3

6

4

5

结果：

Total rela=-18 total imag=39

心得体会：

友元函数具有自己的的访问规则：友元函数属于当前类的成员函数，是独立于当前类的另一个外部函数，但它可以访问该类所有的成员，包括私有成员、保护成员和工有成员。

Copyright ©2021-2099 YuenZhu. All rights reserved