实验报告（第一次上机）

自2001 杨洺宇

实验一：

代码：

#include<iostream>

//#include<cstring>

#include<cmath>

using namespace std;

class complex {

private:

double real;

double image;

public:

complex(double real1=0 ,double image1=0):real(real1),image(image1){}

friend complex operator\*(complex& x, complex& y);

void show();

};

void complex::show()

{

cout << "The multiplication of the complexes is : " << endl;

cout << real;

if (image > 0)

cout << "+";

if(image!=0)

cout << image << "i" << endl;

}

complex operator\*(complex& x, complex& y)

{

complex temp;

temp.real = x.real \* y.real;

temp.image = x.image \* y.image;

return temp;

}

int main()

{

double x1, x2, y1, y2;

cout << "Please input the first complex: " << endl;

cin >> x1 >> y1;

cout << "Pease input the second complex: " << endl;

cin >> x2 >> y2;

complex a(x1 ,y1);

complex b(x2, y2);

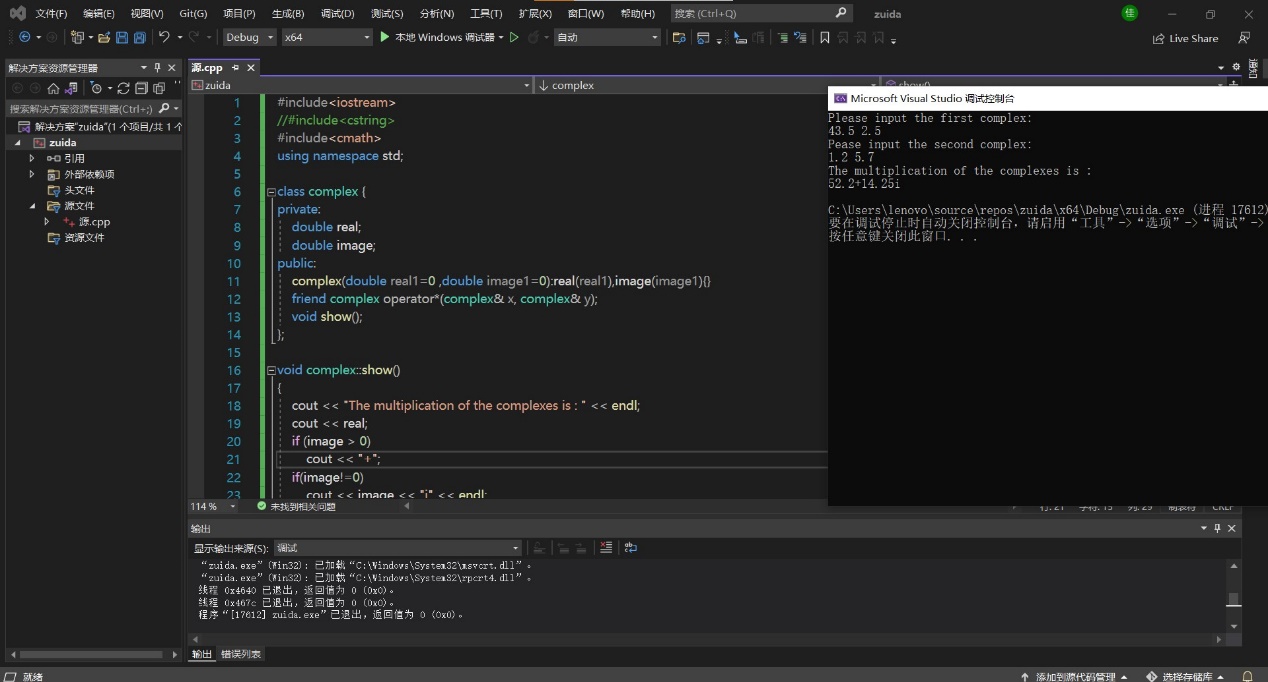
complex c;

c = a \* b;

c.show();

return 0;

}

图片：  


实验二：

代码：

#include<iostream>

//#include<cstring>

#include<cmath>

using namespace std;

class complex {

private:

double real;

double image;

public:

complex(double real1=0 ,double image1=0):real(real1),image(image1){}

friend complex operator\*(complex& x, complex& y);

void show();

};

void complex::show()

{

cout << "The multiplication of the complexes is : " << endl;

cout << real;

if (image > 0)

cout << "+";

if(image!=0)

cout << image << "i" << endl;

}

complex operator\*(complex& x, complex& y)

{

complex temp;

temp.real = x.real \* y.real-x.image\*y.image;

temp.image = x.real \* y.image+x.image\*y.real;

return temp;

}

int main()

{

double x1, x2, y1, y2;

cout << "Please input the first complex: " << endl;

cin >> x1 >> y1;

cout << "Pease input the second complex: " << endl;

cin >> x2 >> y2;

complex a(x1 ,y1);

complex b(x2, y2);

complex c;

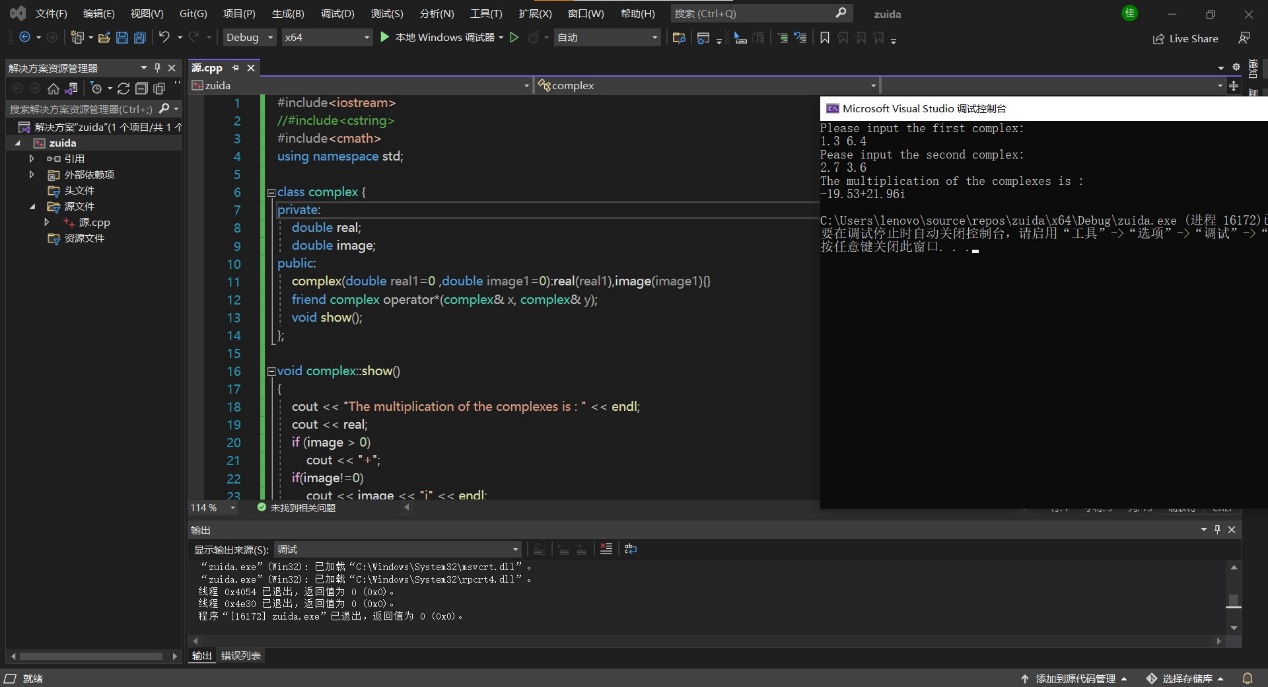
c = a \* b;

c.show();

return 0;

}

图片：



总结：

本次上机实验运算符重载的实质就是函数重载或函数多态。运算符重载是一种形式的C++多态。目的在于让人能够用同名的函数来完成不同的基本操作。要重载运算符，需要使用被称为运算符函数的特殊函数形式，运算符函数形式：operatorp（argument-list）//operator 后面的'p'为要重载的运算符符号。