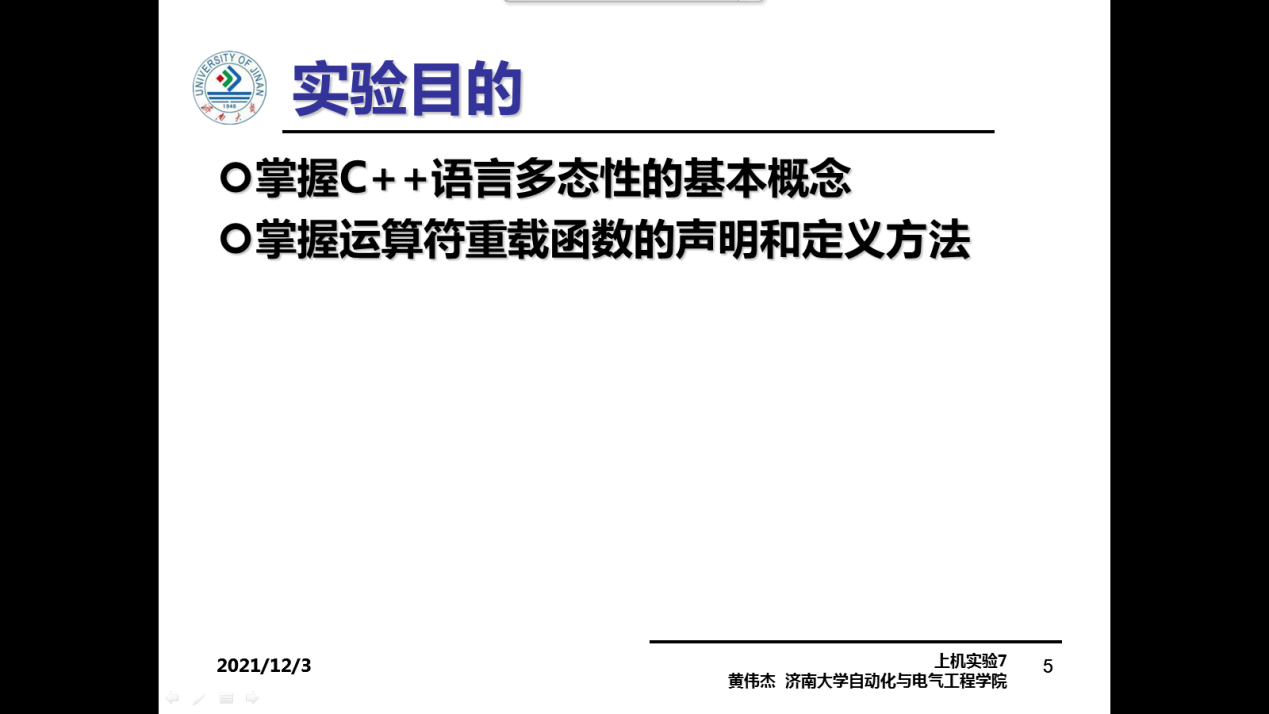
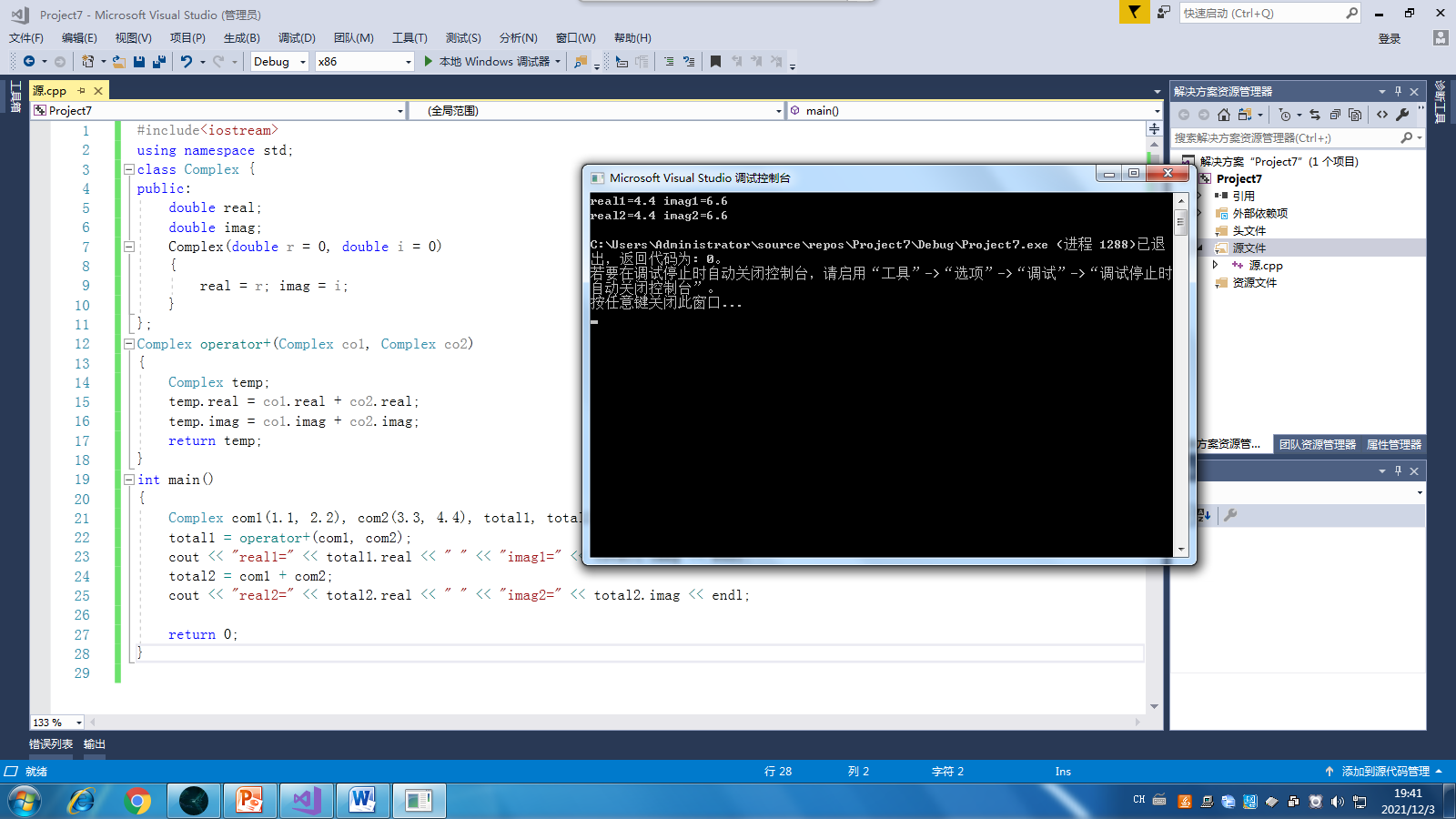
**实验七 12.03**



1. **操作**

**利用类外定义运算符重载函数**



#include<iostream>

using namespace std;

class Complex {

public:

double real;

double imag;

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

{

real = r; imag = i;

}

};

Complex operator+(Complex co1, Complex co2)

{

Complex temp;

temp.real = co1.real + co2.real;

temp.imag = co1.imag + co2.imag;

return temp;

}

int main()

{

Complex com1(1.1, 2.2), com2(3.3, 4.4), total1, total2;

total1 = operator+(com1, com2);

cout << "real1=" << total1.real << " " << "imag1=" << total1.imag << endl;

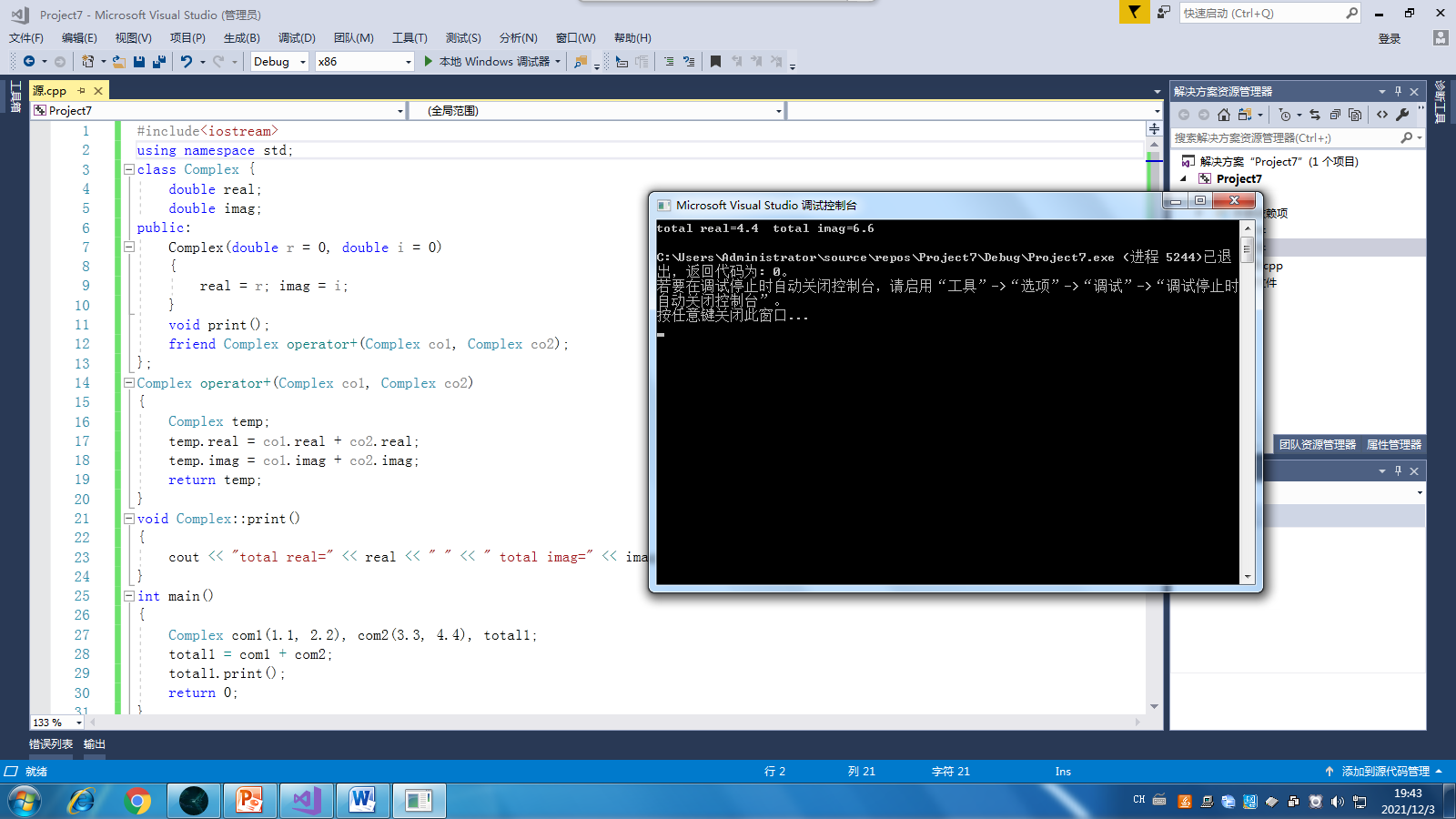
total2 = com1 + com2;

cout << "real2=" << total2.real << " " << "imag2=" << total2.imag << endl;

return 0;

}

**利用友元运算符重载函数**



#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;

temp.imag = co1.imag + co2.imag;

return temp;

}

void Complex::print()

{

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

}

int main()

{

Complex com1(1.1, 2.2), com2(3.3, 4.4), total1;

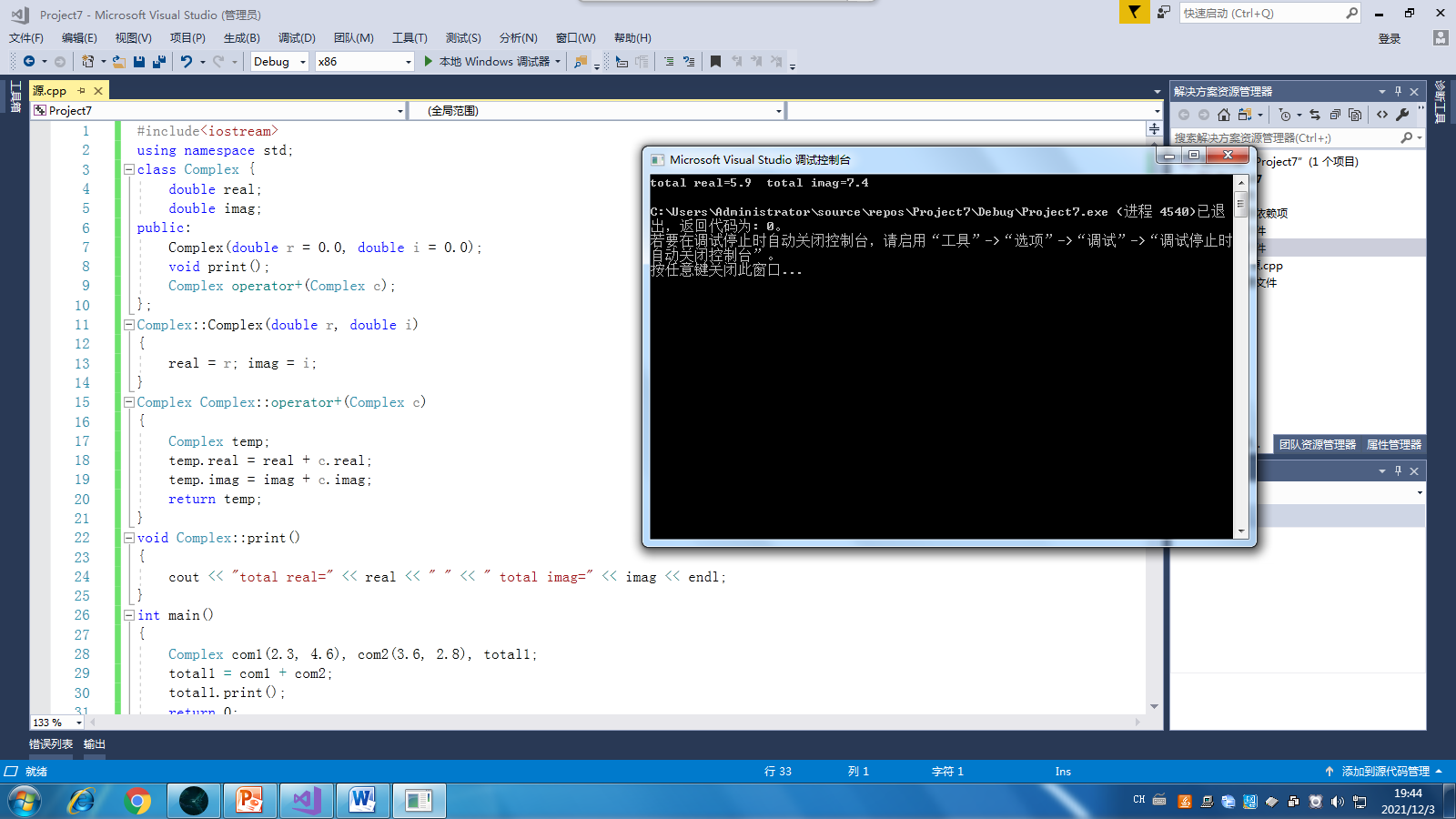
total1 = com1 + com2;

total1.print();

return 0;

}

**利用成员运算符重载函数**



#include<iostream>

using namespace std;

class Complex {

double real;

double imag;

public:

Complex(double r = 0.0, double i = 0.0);

void print();

Complex operator+(Complex c);

};

Complex::Complex(double r, double i)

{

real = r; imag = i;

}

Complex Complex::operator+(Complex c)

{

Complex temp;

temp.real = real + c.real;

temp.imag = imag + c.imag;

return temp;

}

void Complex::print()

{

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

}

int main()

{

Complex com1(2.3, 4.6), com2(3.6, 2.8), total1;

total1 = com1 + com2;

total1.print();

return 0;

}

1. **问题**

1.重载不能改变运算符的优先级和结合性。

2.重载不会改变运算符的用法，原有有几个操作数、操作数在左边还是在右边，这些都不会改变。例如~号右边只有一个操作数，+号总是出现在两个操作数之间。

3.运算符重载函数不能有默认的参数，否则就改变了运算符操作数的个数。

4.运算符可以重载为全局函数，然后声明为类的友元。

5.重载运算符()、[]、->、或者赋值运算符=时，只能将它们重载为成员函数，不能重载为全局函数。

1. **感受心得**

多态性也是C++特有的，是C语言所没有的。课本中给的定义是所谓多态性就是不同对象收到相同消息，产生不同的行为。而我认为就是用一个名字定义不同的函数，这些函数执行不同但又类似的操作，这样就可以用同一个函数名调用不同内容的函数。也就是重载。

这次上机我学会了三种不同的运算符重载函数，利用运算符重载函数就可以实现对数据的运算，C++不像C语言那样两个或者是一个变量可以直接进行运算，他需要运算符的帮助，好像操作起来更复杂了，写的代码也多了。

Copyright©2021-2099 MengyuanWang.All rights reserved.