1. 运算符重载

定义复数类Complex，

（1）定义私有数据成员，实部和虚部；

（2）定义带默认值的构造函数，给数据成员赋值，默认值为0；

（3）定义输出函数void display()，输出格式如“3+4i”或“-3-4i”；

（4）定义复数与复数的加、减运算符重载；复数加实数运算符重载；赋值运算符重载；

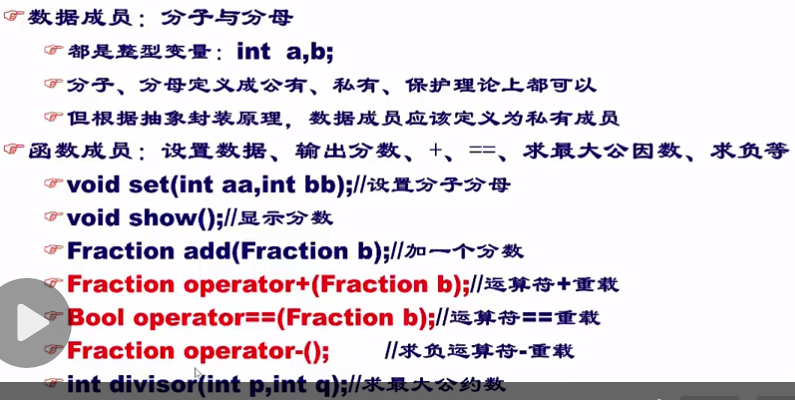
（5）定义主函数，对上述成员函数进行测试。

1. 对“>”进行重载，实现比较日期大小。

定义日期类Date，

1. 定义私有数据成员，年、月和日；
2. 定义带默认值的构造函数，给数据成员赋值，默认值为1900年1月1日；
3. 定义init函数，实现对年月日初始化；
4. 定义print\_ymd函数，实现输出年月日格式；
5. 定义print\_mdy函数，实现输出月日年格式；
6. 定义日期与日期的“>”运算符重载；
7. 定义析构函数；
8. 定义主函数，对上述成员函数进行测试。
9. 用运算符重载的知识，实现对分数的加、等于、求负数的运算。

定义分数类：



【参考代码】

第一题

using namespace std;

#include<iostream>

class Complex

{

private:

double a,b;

public:

Complex(double a=0,double b=0)

{

this->a=a;

this->b=b;

}

void display()

{

if(b>0)

{

cout<<a<<"+"<<b<<"i"<<endl;

}

else if(b<0)

{

cout<<a<<b<<"i"<<endl;

}

else if(b==1)

{

cout<<a<<"+i"<<endl;

}

else if(b==-1)

{

cout<<a<<"-i"<<endl;

}

else if(b==0)

{

cout<<a<<endl;

}

}

Complex operator+(Complex c)

{

Complex temp;

temp.a=a+c.a;

temp.b=b+c.b;

return temp;

}

Complex operator-(Complex c)

{

Complex temp;

temp.a=a-c.a;

temp.b=b-c.b;

return temp;

}

Complex operator+(double d)

{

Complex temp;

temp.a=a+d;

temp.b=b;

return temp;

}

Complex &operator = (Complex x)

{

a=x.a;

b=x.b;

return \*this;

}

};

int main()

{

Complex c1(3,4),c2(5,6),c3,c4;

cout<<"c1=";

c1.display();

cout<<"c2=";

c2.display();

cout<<"c1+c2=";

c3=c1+c2;

c3.display ();

cout<<"c1-c2=";

c3=c1-c2;

c3.display ();

c3=c1+3;

cout<<"c1+3=";

c3.display ();

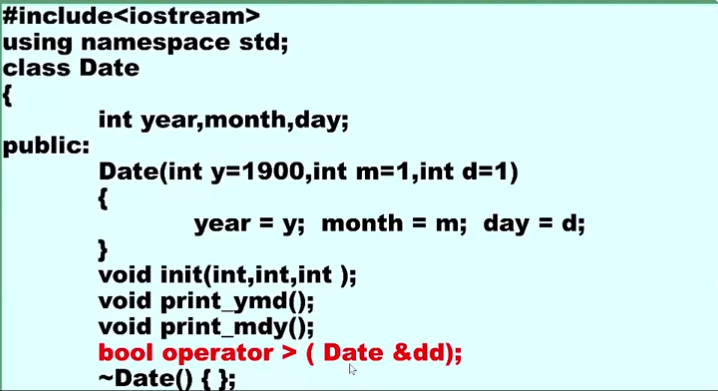
c4=c1;

cout<<"c4=c1, c4=";

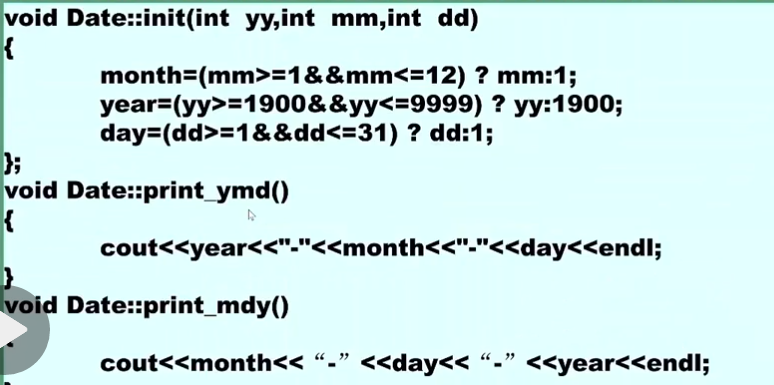
c4.display();

}

第二题:









第三题

