

中山大学移动信息工程学院本科生实验报告

(2015-2016 学年春季学期)

课程名称: Data structures and algorithms

任课教师: 张子臻、黄淦

年级	15	专业(方向)	软件工程(移动信息工程)
学号	15352408	姓名	张稼伟
电话	13531810182	Email	709075442@qq.com
开始日期	2016. 3. 28	完成日期	2016. 3. 28

1. 实验题目

1000:

实现一个复数类, 包括构造函数, 重载的加法, 减法, 乘法, 除法, 中括号和输入输出流。

1001:

实现一个字符串类, 包括构造函数, 析构函数, 重载的赋值运算符, 加法, 中括号, +=, ==, != 以及输入输出流。还有一个求字符串长度的函数。

2. 实验目的

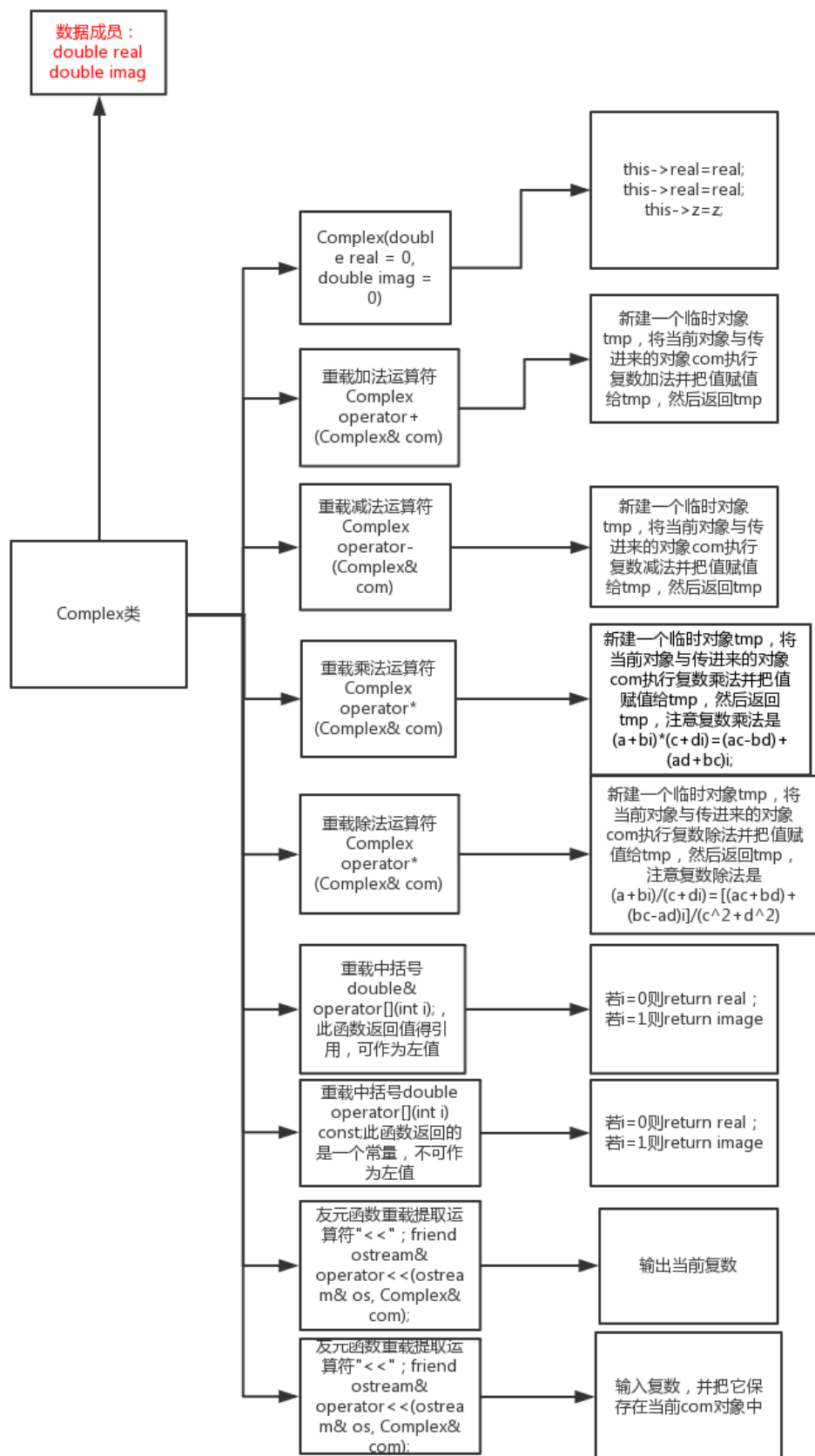
- A. 熟悉并掌握友元函数的应用。
- B. 熟悉并掌握重载运算符的写法和应用。
- C. 体会面向对象编程的特点。

3. 程序设计

1000: 按照相关函数的功能编写代码, 其中值得注意的几点有:

1. `double& Complex:: operator[](int i)` 和 `double Complex:: operator[](int i) const` 的区别在于 `double operator[](int i) const;` 声明了函数不修改类中变量, 但这个函数只是返回一个值, 不能做左值。
而 `double & operator[](int i);` 可以做左值。
2. 复数的乘法和向量不一样, 是 $(a+bi)*(c+di)=(ac-bd)+(ad+bc)i$
3. 复数的除法是 $(a+bi)/(c+di)=[(ac+bd)+(bc-ad)i]/(c^2+d^2)$
4. 在重载提取运算符的函数中要注意输出复数的时候要分类讨论虚部为 i 或 $-i$ 时不用在 i 前输出 1。

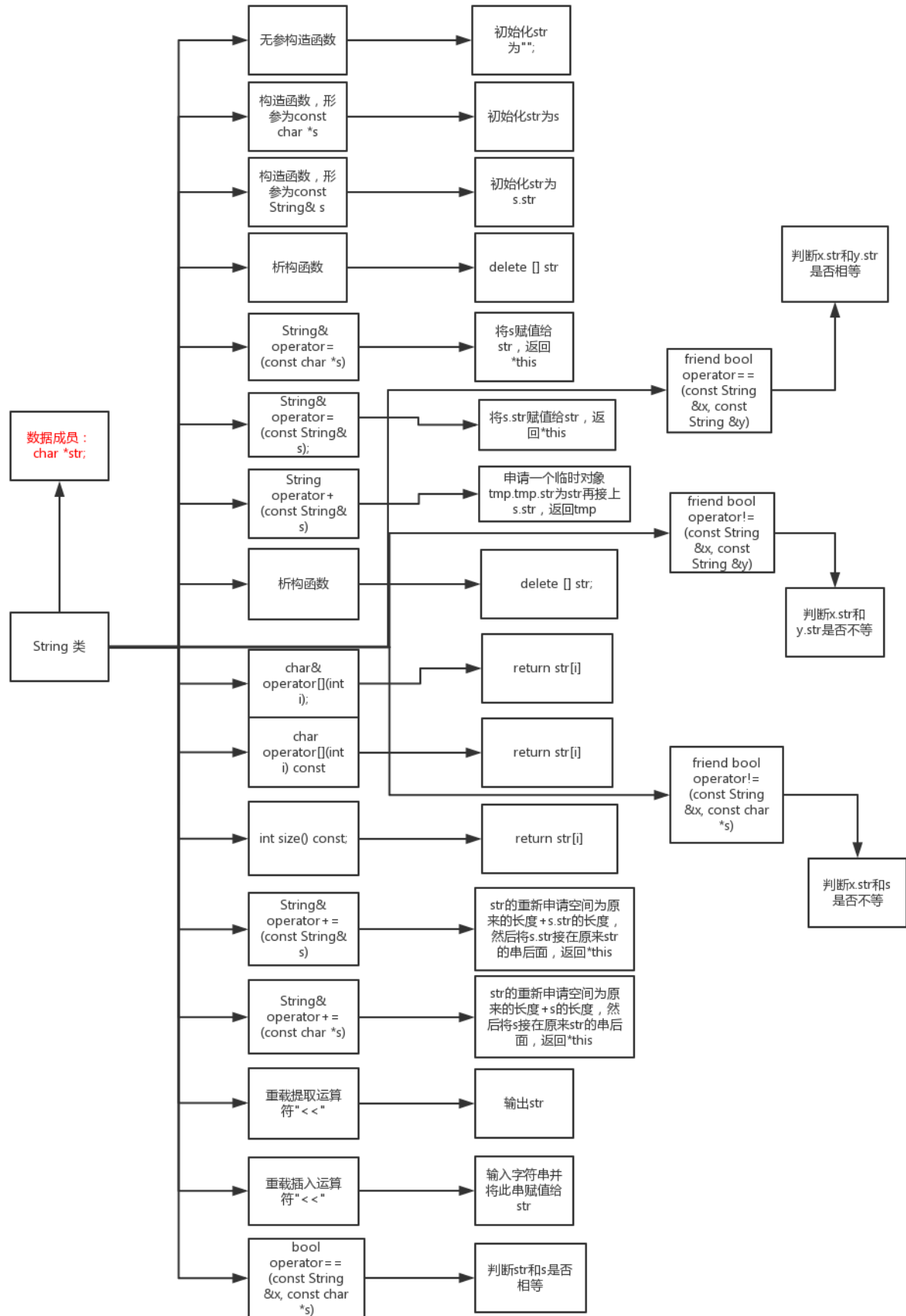
流程图如下:



1001: 按照相关函数的功能编写代码，其中值得注意的几点有：

1. 无参构造函数中，不能直接 `str=NULL`，这样后续代码中若有 `strlen(str)` 则会出错，可以 `new` 一位空间给它，并赋值成空串。
2. 重载插入符 `>>` 时，由于要获得读入串的长度我们可以先声明一个 `string` 类型的变量 `ss`，把数据读到 `ss` 中再赋值给 `str`。

流程图如下：



两题完整可测试代码详见附件 1000.cpp 和 1001.cpp

4. 程序运行与测试

1000:

```
3
1 2
1+2i
1 2
2 1
2+i
2 1
(1+2i)+(2+i)=3+3i
(1+2i)-(2+i)=-1+i
(1+2i)*(2+i)=5i
(1+2i)/(2+i)=0.8+0.6i
1 0
1
1 0
0 -1
-i
0 -1
(1)+(-i)=1-i
(1)-(-i)=1+i
(1)*(-i)=-i
(1)/(-i)=i
-1 -1
-1-i
-1 -1
1 1
1+i
1 1
(-1-i)+(1+i)=0
(-1-i)-(1+i)=-2-2i
(-1-i)*(1+i)=-2i
(-1-i)/(1+i)=-1
请按任意键继续. . .
```

```
3
45 6
45+6i
45 6
78 -89
78-89i
78 -89
(45+6i)+(78-89i)=123-83i
(45+6i)-(78-89i)=-33+95i
(45+6i)*(78-89i)=4044-3537i
(45+6i)/(78-89i)=0.212496+0.319386i
1001 213
1001+213i
1001 213
-11 -9999
-11-9999i
-11 -9999
(1001+213i)+(-11-9999i)=990-9786i
(1001+213i)-(-11-9999i)=1012+10212i
(1001+213i)*(-11-9999i)=2.11878e+06-1.00113e+07i
(1001+213i)/(-11-9999i)=-0.0214122+0.100086i
-88 123
-88+123i
-88 123
123456 78989
123456+78989i
123456 78989
(-88+123i)+(123456+78989i)=123368+79112i
(-88+123i)-(123456+78989i)=-123544-78866i
(-88+123i)*(123456+78989i)=-2.05798e+07+8.23406e+06i
(-88+123i)/(123456+78989i)=-5.34659e-05+0.00103051i
请按任意键继续. . .
```

```
2
123489 853
123489+853i
123489 853
12 -48646
12-48646i
12 -48646
(123489+853i)+(12-48646i)=123501-47793i
(123489+853i)-(12-48646i)=123477+49499i
(123489+853i)*(12-48646i)=4.29769e+07-6.00724e+09i
(123489+853i)/(12-48646i)=-0.0169086+2.53853i
-4545 -89686
-4545-89686i
-4545 -89686
-5 -8
-5-8i
-5 -8
(-4545-89686i)+(-5-8i)=-4550-89694i
(-4545-89686i)-(-5-8i)=-4540-89678i
(-4545-89686i)*(-5-8i)=-694763+484790i
(-4545-89686i)/(-5-8i)=8317+4630i
请按任意键继续. . .
```

1001:

```
2
Please enter two strings
456 789
x= 456 , y = 789
in f: 45x 789 9
789
s = ""
s = "abc"
"456" + "789" = "456789"
exit: !56 456
exit: !bc abc
Please enter two strings
-8888 987
x= -8888 , y = 987
in f: -8x88 987 7
987
s = ""
s = "abc"
"-8888" + "987" = "-8888987"
exit: !8888 -8888
exit: !bc abc
请按任意键继续. . .

2
Please enter two strings
4565465 4565465
x= 4565465 , y = 4565465
in f: 45x5465 4565465 6
4565465
s = ""
s = "abc"
"4565465" + "4565465" = "45654654565465"
exit: !565465 4565465
exit: !bc abc
Please enter two strings
!!!!!! !!!!!
x= !!!!! , y = !!!!!
in f: !!x!! !!!!! !
!!!!
s = ""
s = "abc"
"!!!!!" + "!!!!!" = "!!!!!!!!!!!!!"
write failed!
exit: !!!!! !!!!!
exit: !bc abc
请按任意键继续. . .
```

```
3
Please enter two strings
adsfasdklfhak 4364654dsf
x= adsfasdklfhak , y = 4364654dsf
in f: adxfasdklfhak 4364654dsf 6
4364654dsf
s = ""
s = "abc"
"adsfasdklfhak" + "4364654dsf" = "adsfasdklfhak4364654dsf"
exit: !dsfasdklfhak adsfasdklfhak
exit: !bc abc
Please enter two strings
12309ij;losdfgasl #$$$^$%&dfasjdlfik
x= 12309ij;losdfgasl , y = #$$$^$%&dfasjdlfik
in f: 12x09ij;losdfgasl #$$$^$%&dfasjdlfik $
#$$$^$%&dfasjdlfik
s = ""
s = "abc"
"12309ij;losdfgasl" + "#$$$^$%&dfasjdlfik" = "12309ij;losdfgasl#$$$^$%&dfasjdlfi
k"
exit: !2309ij;losdfgasl 12309ij;losdfgasl
exit: !bc abc
Please enter two strings
4sd5af64asd5f4 dfd2354
x= 4sd5af64asd5f4 , y = dfd2354
in f: 4sx5af64asd5f4 dfd2354 d
dfd2354
s = ""
s = "abc"
"4sd5af64asd5f4" + "dfd2354" = "4sd5af64asd5f4dfd2354"
exit: !sd5af64asd5f4 4sd5af64asd5f4
exit: !bc abc
请按任意键继续. . .
```

5. 实验总结与心得

1. 运算符的重载可以是友元函数也可以是类的成员函数。
2. =, (), [] 和 -> 的重载必须是类的成员函数，不能为友元函数。
3. 重载插入运算符和提取运算符必须是友元函数。
4. 重载之后的运算符不能改变运算符的优先级和结合性，也不能改变运算符操作数的个数及语法结构。
5. 运算符重载是针对新类型数据的实际需要对原有运算符进行的适当的改造，重载的功能应当与原有功能相类似，避免没有目的地使用重载运算符。
6. 重载的运算符只能是用户自定义类型。
7. `T& operator[] (int i); T operator[] (int i) const;` 区分好这两种运算符重载的区别。const 对象才能调用 const 版本的 function 函数，而非 const 对象可以调用任意一种，通常非 const 对象调用不是 const 版本的 function 函数。原因是：按照函数重载的定义，函数名相同而形参表有本质不同的函数称为重载。在类中，由于隐含的 this 形参的存在，const 版本的 function 函数使得作为形参的 this 指针的类型变为指向 const 对象的指针，而非 const 版本的使得作为形参的 this 指针就是正常版本的指针。此处是发生重载的本质。重载函数在最佳匹配过程中，对于 const 对象调用的就选取 const 版本的成员函数，而普通的对象调用就选取非 const 版本的成员函数

附录、提交文件清单

实验报告一份：实验报告.pdf

代码两份：1000.cpp

1001.cpp